3D Style Transfer: Shape-to-Shape Integrated with Texture Style Variation

Zhouheng He(zhouheng@ualberta.ca)
Zihan Su(zsu5@ualberta.ca)

1 Abstract

How artistic would it be if people could apply their favorite art piece style to objects they use daily? Imagine a picture of a camping site among the mountains, with bright stars in the sky. Existing style transfer formulations have performed an artistic transfer from Vincent van Gogh's The Starry Night style to this picture. The original night sky will be stylized into magical spirals with brightening colors, which can be done using convolutional neural networks. However, the realm of 3D objects instead of 2D images has not yet been fully discovered. In this project, we will focus on training through the neural network to transfer the style of a piece of art onto a 3D model. The goal of our research is to present an improved formulation that could both edit the shape and the texture of the 3D model so that it looks in the style of a given style image from multiple viewpoints. Considering the advantages of compactness and geometric properties, we will use 3D meshes for the style transfer. To be more specific, for a 3D teapot that is modeled in the mesh format, by keeping adjusting the losses regarding the relations between the vertices, pixel colors, and the style image, the neural network could eventually generate a teapot with the stylized shape and texture (Figure 1).

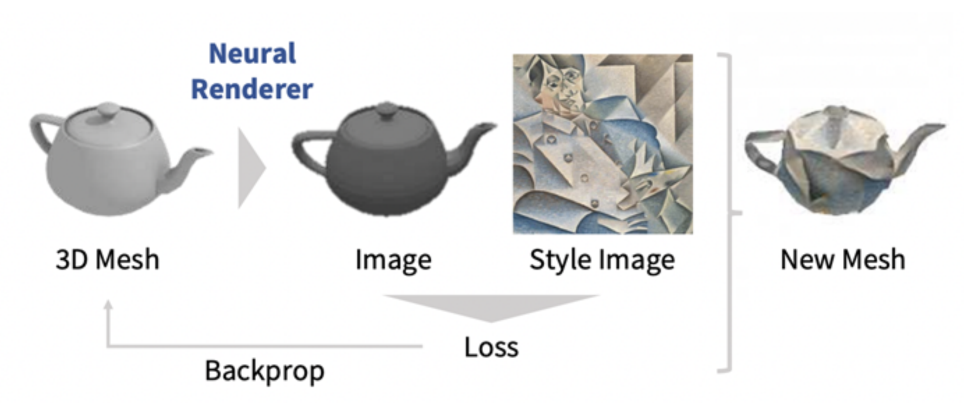

Figure 1: Demonstration of the goal of altering both the shape and texture of a 3D object. [2]

2 Introduction

Commonly, style transfer in 2D images is used in the art domain (Figure 2), which grants normal images artistic meaning. The demand for style transfer in three dimension objects is trending nowadays; while editing 3D content is arguably a more arduous and time-consuming task, which makes automatic tools especially attractive, an equally successful formulation of style transfer for the 3D domain has not been proposed. 3D style transfer, in contrary to 2D image style transfer, focuses more on the innovation domain, which includes not only building indoor meshes but also reshaping objects and adding additional textures to them.

Figure 2: The same picture being transferred into different artistic styles. [16]

The current existing work uses point clouds or meshes to generate the 3D objects' surface and hence alter the color or shape of the objects but does not extend to both of these attributes of the objects. Classical methods consider shape transformation from one object to another with the guidance of shape correspondence and the texture transformation map from one shape to another by minimizing distortion. Deep learning methods are only able to address either geometry or texture stylization. For example, [1] focuses on geometric transfer, which synthesizes stylized textures instead of directly importing from the input images. [2] on the other hand, demonstrates how to apply the style of art painting to the surface of 3D objects as texture. None of the above methods is able to perceive and transfer the geometric and texture style jointly from one 3D object to another.

In this project, we aim to produce a novel and effective method which can jointly transfer both the geometric and textural attributes of 3D objects. Unlike the other existing methods, which only focus on one aspect of the attributes, we would like to outperform them since 3D objects' shape, and texture that can be altered simultaneously have more potential in both the art and science domains.

3 Brief Summary of Existing Work

The neural style transfer on 2D images has achieved significant success, but an equally developed algorithm for the 3D style transformation has yet to be discovered [3]. The previous work related to 3D style transfer focuses mainly on shape and color stylization. In this section, we introduce the neural style transfer first, as it is the base on where 3D stylization is built, followed by the existing studies that will transfer the style of paintings to the shapes and texture.

3.1 Neural style transfer

Neural style transfer is an optimization technique that takes a content image and a style image as input and then generates an image that is similar to the content image but painted in the specified style [6]. For 2D images, style transfer is achieved by minimizing content loss and style loss simultaneously.

3.2 3D Representation in neural networks

Common 3D representations used in such areas include voxels, point clouds, and meshes. Voxel is an extension of 2D pixels to 3D, as a volumetric data structure organized along a 3-dimensional grid, regular 3D convolutional neural networks can directly be trained on voxel representations of 3D shapes[5]. However, with a low memory efficiency, it is difficult to process high-revolution voxels[6]. Point clouds have relatively high scalability, but the lack of surfaces increases the difficulty of texture transfer[6]. Prior studies show that meshes, which consist of vertices and surfaces, have more advantages over other 3D formats[6].

3.3 3D Representation in neural networks

3D reconstruction is a traditional topic but still a difficult goal in computer vision. In 2018, the 3D mesh-based reconstruction that was proposed by Ushiku et al. was evidenced to overperform the existing voxel-based approach [2]. They also discovered that single-image 3D reconstruction could be realized without 3D supervision. Perspective transformer nets (PTN) were introduced to learn 3D structures using silhouette images from multiple viewpoints [2].

For the 2D-to-3D stylization part, with only 2D supervision, Ushiku et al. proposed a gradient-based mesh editing to edit the vertices and textures of a 3D mesh to match it onto a specified art style [2]. They applied the method to the initial untextured 3D models. This application demonstrates the potential of integrating mesh renderers into neural networks. Moreover, 3DSTYLENET, the model proposed by Yin et al., is composed of three components: Geometric style transfer network, pre-trained texture style transfer network, and joint geometry and texture optimization [3].

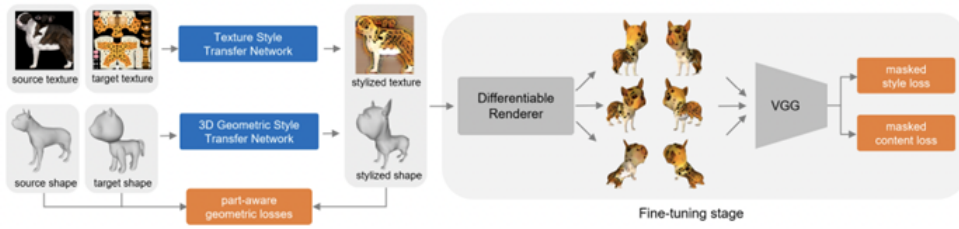

Figure 3: An overview of 3DSTYLENET. [3]

4 Plan To Do

For CMPUT 414, we will only focus on transferring a style of painting, including both shape and color, onto a 3D mesh. The potential tasks could be divided into two parts: The estimation of 3D structures from images and the artistic style transfer applied to the generated 3D models.

5 How to Implement

Our project will focus on the deep learning approach. Typically there are several aspects that we can focus on to improve the performance of our solution, which includes the architecture of the deep learning network and the representation of models.

For this project, we will mainly use python as a developing language, and the database in which the AI will be trained will be imported from earlier existing projects. The benefit of directly importing pre-existing data is to save time and effort in gathering data. Also, large-scale existing data that has already been used in research also grants better quality of these data.

Regarding related libraries, We plan to primarily use PyTorch and PyTorch3d for the construction of neural network architecture toward a 3d shape. For the data structure, we plan to apply the style transfer to 3D mesh as it is easy to be applied texture upon.

Our code will start with implementing some pre-existing work found in the paper with code and improve it based on it. Right now, we are aiming at work done by Hiroharu Kato[2].

6 Timeline

From now to Oct 22, 2022 - Project Literature Review

Each day before the deadline, each of us will read through an article. We will take notes on it and summarize it, following the instructions provided by the professor. Three days before the deadline, we will start organizing what we have found so far to finalize the submission.

Oct 22, 2022 - Nov 5, 2022 - 3D Reconstruction

Working on the estimation of the 3D model from a single 2D image

Nov 5, 2022 - Nov 20, 2022 - 2D-to-2D style transfer

Working on applying stylization to the initial state of the 3D models.

Nov 20, 2022 - Dec 1, 2022 - Creativity and improvement

Working on the improvement of the implementation.

Dec 1, 2022 - Dec 5, 2022 - Correctness

Checking the correctness of the codes. Detecting bugs if they exist.

Dec 1, 2022 - Dec 3, 2022 - Presentation

Working on making PowerPoint and rehearsal.

Dec 5, 2022 - Dec 14, 2022 - Final Report

The first draft should be done before Dec 10, 2022.

7 Short Description of 5 Labs

First Lab: In the first lab, we will have the code from the reference article run successfully. Based on the code, we will discuss our preliminary ideas with Guangfang to see if we can start coding in this direction.

Second Lab: We are expected to start the implementation of the 3D reconstruction before this lab. We will save this lab from discussing questions we met during the implementation.

Third Lab: We plan to use this lab as a Q&A session for the 3D geometric style transfer. We will ask Guanfang about issues we met during the implementation as well as discuss with classmates.

Forth Lab: We will discuss 3D texture stylization with Guanfang and classmates who have related research topics with us.

Fifth Lab: We would like to save this lab as a Q&A session to ask Guanfang and other classmates about practical problems we met during the implementation. We would also like to demonstrate what we have achieved to people who are interested.

8 Literature Review

8.1 Review of the 2D style transfer

As the foundation of the 3D style transfer, 2D style transfer is the area that we consult first for our research. It focuses on the projection of artistic style onto existing 2D images. Traditional art style transfer techniques like non-photorealistic rendering (NPR) are designed only for a particular artistic style, which limits the practical application in real life. However, Gaty et al. started the era of neural style transfer by introducing the deep neural network to the style transfer in 2015 [15]. Thereafter the style transfer became a popular topic, and countless research is conducted to improve the performance of the neural style transfer.

The main advantage of neural style transfer is the ability to perform satisfying generalizations toward different art styles. It well solved the issue that has existed in the old approaches for a long time, that is, a program can only define one specific framework for a style in the realm of computer graphics, namely image analogies. Besides, the traditional approaches only use low-level image features and often fail to capture image structures effectively [7].

Through multiple experiment trials, convolutional neural networks (CNNs) are capable of extracting the content information from images and style information from artworks. The algorithm for this process is to iteratively optimize the desired feature that will be transferred to the image, which contains both the original image's information as well as the art source's artistic style.

There are two current categories of NST algorithms. The first one is Image-Optimisation-Based Online Neural Methods that iteratively optimize images. This method first models and extracts the style and content from the source images and then recombines them as the target representation. After that, it iteratively produces results that match closer to the target. The second one is Model-Optimisation-Based Offline Neural Methods, which optimizes an offline model and only produces a stylized image with a single forward pass. The advantage of this approach is its high efficiency, and its feed-forward network is optimized over a large set of images for one or more style images.

8.2 Review of the 3D reconstruction

When doing 3D style transfer, the first step is to generate corresponding 3D objects given a content image, like faces, chairs, buildings, and cars. There are usually two approaches to achieve this step. The first one is to import meshes into our database directly. The shortage of this approach is obvious: It would be rather hard to obtain a large amount of dataset of meshes that can be used for our project. Although there have already existed some datasets built by other researchers, and we can certainly adapt their data and continue our research based on that, there is an alternative that is proposed by a researcher: to generate 3D models using CNNs.

The idea behind generating source images using CNN, presented by Alexey et al.[8], is to use an "up-convolutional" network. It assumes that high-level information is given and uses supervised training to generate high-quality images. Compared to the previous approaches, which mainly use unsupervised generative models, this approach from Alexey et al. has the advantage of extremely high-quality images which reach 128 * 128 pixels, while the prior can only generate 48 * 48 pixels images. The latter also has the ability to have full control over the images rather than random sampling.

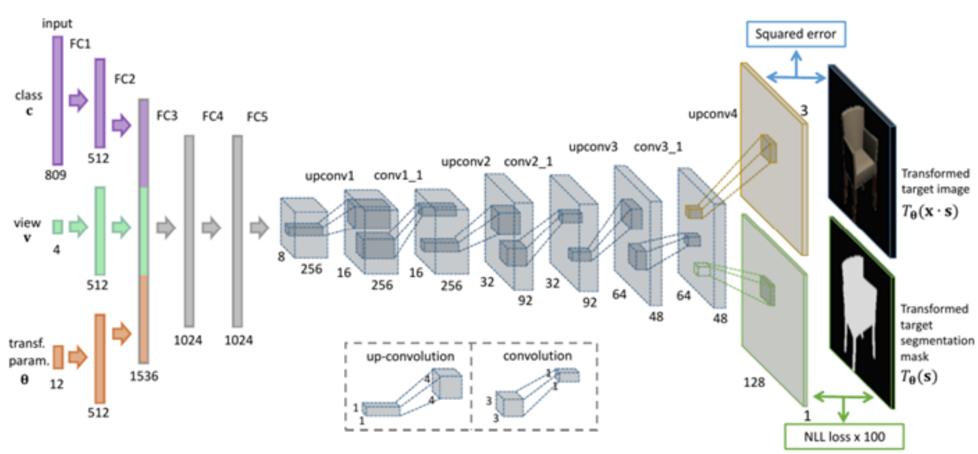

Figure 4: Architecture of a 1-stream deep network ("1s-S-deep") that generates 128 * 128 pixel images. [8]

The network trained specifically for this generative work learns not only to generate the training samples but also a generic implicit representation, which allows the transformation between different object views or instances, and all the intermediate images being useful and meaningful. In addition, the program shows the ability to generate brand-new object styles when trained in a stochastic regime for transferring object styles.

Another factor that affects the final effect of the 3D reconstruction is the choice of the 3D representation, as demonstrated by Kato et al. stated in 2018. The forms in 3D understanding could be categorized into two types: rasterized and geometric. Rasterized formats include voxels and multi-view RGB(D) images, while the geometric formats include point clouds and polygon meshes [2]. Rasterized forms are widely used since they can be directly processed by CNNs; geometric forms require some modifications to be integrated into neural networks [2]. However, the meshes are often an advanced choice when constructing 3D models and applying for style transfer on that. This is because it is memory efficient and has suitable geometric attributes: vertices and surfaces [2].

Kato et al. presented a single-image 3D mesh reconstruction without 3D supervision, which exceeded the voxel-based approach. And by introducing a smoothness loss to the algorithm, the program could generate 3D mesh objects with smooth surfaces, which is the foundation of the following gradient-based mesh editing part [2].

8.3 Review of the example-based painting on 3D mesh sculpture

Once the 3D reconstruction is completed, the main focus of the research is encountered. That is, to paint on a 3D model. We discovered that recent research prefers 3D mesh models over other 3D formats. Among the common representations in 3D understanding, high-resolution voxels are hard to process for their low scalability and inefficient memory, while the lack of surface of the point clouds causes difficulty in lighting and texture stylization [2]. Scalable meshes with groups of vertices and surfaces now seem to be a good match for 3D style transfer. The main benefits of using meshes could be summarised into two categories: (1) Remarkably reduce the model size as the number of points that need to be manipulated when processing significantly decreases; (2) Efficiently simplify the geometric transformation with its property of vertices. Based on this knowledge, we decided to propose a 3D style transfer algorithm specifically for mesh-based sculptures [2].

While an enormous effort has been made to paint and sculpt the 3D models so that both texture and geometry could be stylized while preserving the identity content, a few studies demonstrated distinguished contributions to improving the effectiveness of generated results.

The example-based illumination-dependent stylization algorithm proposed by Fišer et al. in 2016 raised the thinking of the importance of lighting when painting on a 3D model. They pointed out that one of the major reasons that the current state-of-art techniques failed to generate results that were consistent with the real art piece was the neglect of the lighting effects [12]. Although lighting is a prime technique that artists use to depict 3D scenes, previous methods tend to pay more attention to the colors and normals. Therefore, when they run into regions with similar colors, it is likely that they cannot distinguish the difference between them [12]. In such cases, adding illumination conditions to the 3D stylization algorithm is necessary.

The results of the illumination-dependent 3D style transfer were desirable. By comparing the different combinations of the existing methods, StyLit, the algorithm of colors with illumination conditions proposed by Fišer et al., was the only one that precisely captured all of the lighting and shading effects in the stylized 3D object. However, this approach requires similar lighting environments between the source and target meshes, which greatly limits the performance of the application [12].

The existing studies on 3D style transfer focus not only on the effectiveness of the results but also on the runtime of executing the program. In 2019, Sýkora et al. presented another algorithm, StyleBlit, that could reduce the total running time to a huge extent by introducing a conception of local guidance. Local guidance is a descriptive guiding channel containing vast spatial variations, which consists of normal values, texture coordinates, or a displacement field [13]. This feature allows the program to approximate the corresponding location in the source exemplar given a random pixel in the target 3D model without expensive computation [13]. The authors compared the results generated from the study with the results from the study conducted by Fišer et al.. They found that the two methods produced visually similar output, and surprisingly no observable seams were observed in the ones generated by the newly proposed approach. Moreover, with the running time being 0.05 seconds by StyleBlit and 56 seconds by StyLit, their approach was more than 1000 times faster. Even with such a good performance on 3D stylization, there are still some limitations that exist in StyleBlit to be solved in future research. One of the major issues that StyleBlit has is that it performs well only for the transformation from stochastic styles. With regards to the textures that have more obvious patterns on, StyleBlit will fail to simulate the correct style on the 3D models [13]. Another limitation is that during the chunk alignment process, when the guidance channels used for the style transfer do not include local guidance or other guidances dominate the influence, misalignment between the source and the target could be visible [13].

8.4 Review of the texture and geometry stylization on 3D mesh sculpture

The studies above focused mainly on texture stylization, yet our goal is to apply a 3D style transfer to both the texture and geometry of a model. A large number of recent studies have built neural networks to achieve this. Among them, the neural 3D mesh renderer (NMR) proposed by Kato et al. in 2018 is a big step forward to a better style transfer on general 3D objects.

Kato et al. integrated the mesh rendering into neural networks with a new approximate gradient and then performed editing operations on mesh vertices and surfaces based on NMR, so that the final result could have both shape and texture fitted in a given art style. For a more detailed implementation, the program repeatedly adjusts the vertices and pixel colors through backpropagation and gradient descent so that the sum of the content loss, style loss, and regularizer is minimized. The major issue of this approach is that NMR cannot understand semantically meaning, causing problems such as the inability to capture facial parts [14].

In 2021, specifically for the face-portrait style transfer, a study introduced landmark translation to identify the coarse geometry shape of a given artistic portrait. Han et al.

utilized landmarks as a guide to deform the 3D face model. The entire system presented in the study can be separated into two stages. The first stage is responsible for the geometric style transfer, which is where landmark translation benefits. The second stage is to transfer the texture style from the source image to the 3D mesh model that is generated in the first stage [14]. A differentiable renderer is developed in this stage to optimize the texture in multiple views.

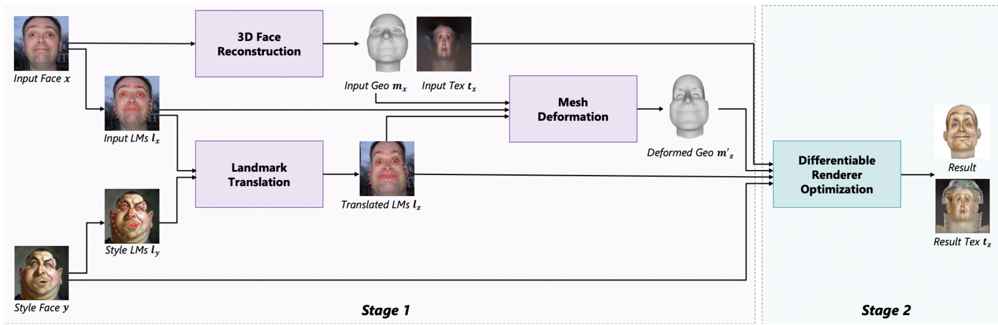

Figure 5: An overview of the system developed by Han et al.. [14]

The researchers used the loss function similar to the one that Kato et al. used, only to replace the loss for regularizer by adding weights to the content loss. It could be observed from the results that the new landmark translation captured the face portraits relatively precisely and at high speed. The average processing time for the landmark translation per example only took 0.2 seconds, while the multi-view optimization was rather slow (255 seconds per example). Therefore, future efforts could be devoted to the replacement with a faster 3D texture stylization.

As stated before, our research aims to develop a system that can efficiently stylize a 3D mesh model texturally and geometrically. Based on the studies we have reviewed so far, we lean toward using landmarks to implement our geometry style transfer for its effectiveness and efficiency. We will extend the use of the landmarks to more scenes so that the landmark translation could facilitate the deformation of the basic 3D model from multiple art styles, not only from face portraits.

For texture stylization, there is not a firm preference on how painting on a 3D model should be conducted. But we will attempt the gradient-descent mesh editing in NMR and the fast local-guidance algorithm proposed by Sýkora et al.. We will also add lighting conditions if there is a quantum leap in the quality of the stylized 3D meshes. The results of the two attempts are expected to give us a straightforward demonstration of the performance of the combination of the landmark translation and other texture stylizations so that we could know where the efforts need to be put in order to accelerate the program while the final results could be similar to a given style without losing any identity traits.

8.5 Review of algorithms on implementing neural networks for training

When facing multiple choices of importing 3D models/sculptures, efficiency, performance, and memory occupancy is our scale for deciding which approach to take. After importing

3D models/sculptures, it is essential to establish effective algorithms for the neural network and start training. One popular way is to adopt the NeRF idea.

NeRF is a continuous 5D function whose input is a 3D location and 2D viewing directions and outputs color and volume density. NeRF is approximated by a multi-layer perceptron and is trained using a loss function, along with some randomly sampled camera rays. But the disadvantage of NeRF approach is obvious: It requires a large amount of memory and time to train since this approach requires that the memory be shared between three memory-intensive components. As a result, this limits the resolution of stylized results and the stylization method.

Thu et al. proposed a new training method inspired by coordinate descent [11]: They decouple the loss function of transfer and NeRF and minimize them one at a time. Only the extractor, the target style image, and rendered images are featured when computing loss for transfer; only the volumetric renderer and target images are computed for loss of NeRF.

This process alternating training regime allows the computer hardware to focus training on one of the parameters and allows us to perform stylization on the whole image and achieve a more globally consistent result. When training NeRF, it provides higher resolution results and also allows the method to be applied to dynamic scenes.

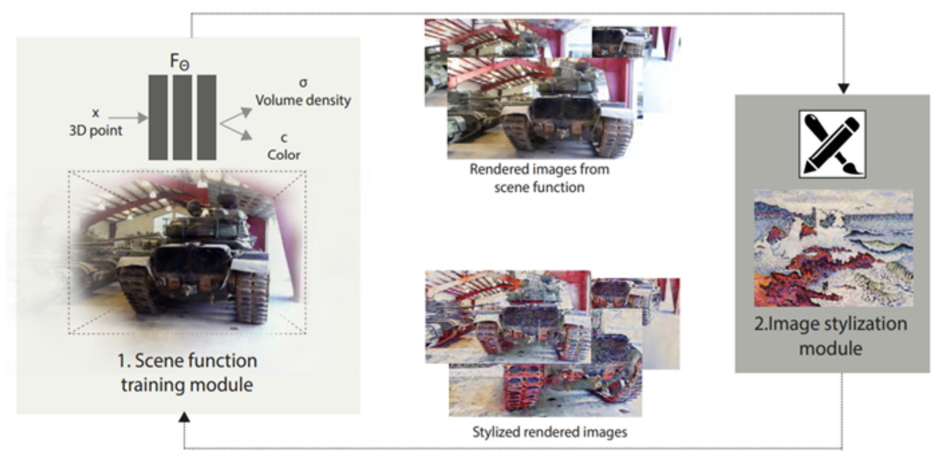

Figure 6: An alternating training approach to stylize implicit scene representations. [11]

8.6 Review of the difficulty of using 3D meshes

Previous contents have manifested the advantages of using 3D meshes to perform 2D to 3D style transfer and introduced a few stylization approaches based on that. However, there are also challenges when using meshes as input. Because the data structure of each mesh is a completed graph with a large size, it is challenging to integrate them properly into neural networks. Several approaches have been proposed to process meshes beforehand so they can be used in neural networks. In the experiment conducted by Kato et al. in 2018, instead of generating a mesh from scratch, they deformed a predefined mesh to generate a new mesh. This way, the number of parameters in a mesh is significantly reduced, even less than the size of s typical voxel.

Another major issue formed because of the different distances and angles of each viewpoint. These properties create view-dependent stretch and size artifacts, and as a result, the simple approach of performing 2D style transfer does not consistent with 3D data like surface normals and depth.

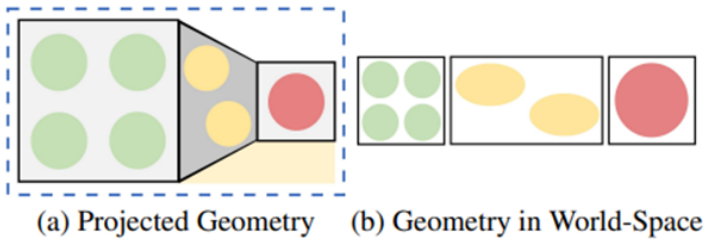

Figure 7: Stylizing mesh faces in 2D (screen-space) is dependent on the angle and size of the projected geometry. [10]

Lukas et al. tried a few approaches to solve the mentioned problems [10]: They utilized depth to render image patches at increasingly larger screen-space resolutions by splitting the loss calculation, and as a result, patterns were now in a view-independent way. By bilinearly interpolating each pixel from four neighboring texels, they defined their texture using a Laplacian Pyramid, which could help to avoid magnification and minification artifacts in the texture. They also applied the concept that each area in screen space is inversely proportional to its depth and incorporated depth awareness by optimizing at multiple screen-space resolutions. That is, larger patterns will appear in the front space of an image rather than the back and finally leads to an equally large style in world space.

To solve the stretching effect caused by different viewing angles, Lukas et al. combined coarse and fine style losses and optimized fine details only for areas that could be seen from good angles. And due to the fact that coarse patterns are not easily affected by the stretch artifacts, they need to use all pixels and a high-resolution style image to optimize coarse stylization patterns. This creates a combination of coarse and fine patterns without stretching artifacts.

This method also has its limitations, each texture image is optimized separately due to the per-scene neural style transfer algorithm, and an already-existing implicit texture representation that could enable training generative models can be adopted to simplify their work. Incomplete mesh reconstruction will also lead to holes in rendered poses, and this can be solved by importing mesh completion techniques firsthand. An insufficient number of poses can also lead to unobserved surfaces during optimization, and this can be fixed by using inpainting techniques.

8.7 Review of stylization on other 3D representations

Another common representation in 3D understanding is point clouds, which is also a tested method to handle the style transfer in 3D.

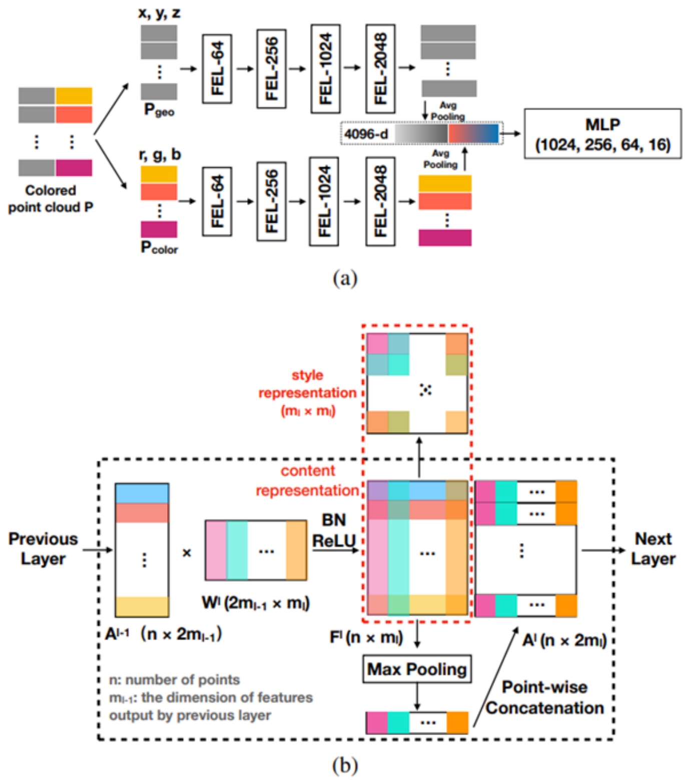

Figure 8: Network architecture for feature extraction. [9]

Inspired by the idea of 2D style transfer to modify the source image as a whole, Xu et al. implemented a method that could edit a whole point cloud by stylizing it from a target [9]. They developed a PointNet- based stylization network named PSNet to extract the representation for the geometry property and the color property of a point cloud, respectively. After that, the authors used Gram-matrix to encode the input point cloud's intermediate outputs and then match it with the style representation of the geometry and color property of another style cloud. PSNet is also able to stylize the color property of a point cloud from an image, which is treated as pixels, as the gram-based style representation is invariant to the number of input points. In all, the PSNet is able to stylize the geometry and color of a point cloud from another point cloud, even if these two point clouds are different.

By modifying the PointNet adopted, two MLPs are applied to geometry and color properties, respectively. Following the idea of neural style transfer, they seek a stylized point cloud that minimizes the loss function between the content representation of the stylized point cloud and the content point cloud. When transferring the style of an image to the point cloud, the image is treated as a set of pixel values, and the stylized point cloud is found following the same loss function equation. With the adoption of the style transfer of point clouds, large-scale indoor scenes or terrain point clouds can be used for virtual reality applications in the future.

9 Reference

[1] Segu, Mattia, et al. "3dsnet: Unsupervised shape-to-shape 3d style transfer." arXiv preprint arXiv:2011.13388 (2020).
[2] Kato, Hiroharu, Yoshitaka Ushiku, and Tatsuya Harada. "Neural 3d mesh renderer." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.
[3] Yin, Kangxue, et al. "3dstylenet: Creating 3d shapes with geometric and texture style variations." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021.
[4] "Neural Style Transfer: Tensorflow Core." TensorFlow, https://www.tensorflow.org/tutorials/generative/style_transfer.
[5] "[3D Machine Learning] - 3D Data Representations." Antoine Toisoul, www.antoinetlc.com/blog-summary/3d-data-representations.
[6] Wikipedia contributors. "Neural Style Transfer." Wikipedia, 28 June 2022, en.wikipedia.org/wiki/Neural_style_transfer.
[7] Y. Jing, Y. Yang, Z. Feng, J. Ye, Y. Yu and M. Song, "Neural Style Transfer: A Review," in IEEE Transactions on Visualization and Computer Graphics, vol. 26, no. 11, pp. 3365-3385, 1 Nov. 2020, doi: 10.1109/TVCG.2019.2921336.
[8] Dosovitskiy, Alexey, et al. "Learning to generate chairs, tables and cars with convolutional networks." IEEE transactions on pattern analysis and machine intelligence 39.4 (2016): 692-705.
[9] Cao, Xu, et al. "Psnet: A style transfer network for point cloud stylization on geometry and color." Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2020.
[10] Höllein, Lukas, Justin Johnson, and Matthias Nießner. "StyleMesh: Style Transfer for Indoor 3D Scene Reconstructions." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.
[11] Nguyen-Phuoc, Thu, Feng Liu, and Lei Xiao. "SNeRF: stylized neural implicit representations for 3D scenes." arXiv preprint arXiv:2207.02363 (2022).
[12] Fišer, Jakub, et al. "Stylit: illumination-guided example-based stylization of 3d renderings." ACM Transactions on Graphics (TOG) 35.4 (2016): 1-11.
[13] Sýkora, Daniel, et al. "StyleBlit: Fast Example-Based Stylization with Local Guidance." Computer Graphics Forum. Vol. 38. No. 2. 2019.
[14] Han, Fangzhou, et al. "Exemplar-based 3d portrait stylization." IEEE Transactions on Visualization and Computer Graphics (2021).
[15] Gatys, Leon, Alexander S. Ecker, and Matthias Bethge. "Texture synthesis using convolutional neural networks." Advances in neural information processing systems 28 (2015)
[16] Gatys, Leon A., Alexander S. Ecker, and Matthias Bethge. "A neural algorithm of artistic style." arXiv preprint arXiv:1508.06576 (2015).