3DStyleMerge: Part-Compatible 3D Style Transfer

Abhinav Upadhyay, Alpana Dubey, Suma Mani Kuriakose

Accenture Labs
Bangalore, India
k.a.abhinav@accenture.com, alpana.a.dubey@accenture.com, suma.mani.kuriakose@accenture.com

Abstract

In this work, we propose a 3D style transfer framework, 3DStyleMerge, that transfers style elements from style to content 3D objects. We apply a combination of learning-based and geometric-based approaches to perform style transfer. Our approach ensures that the functionality of the content 3D object is preserved, by allowing only compatible operations. To evaluate the proposed 3D style transfer framework, we conduct a user study with 3D designers. Our evaluation results demonstrate that our approach effectively generates new designs and the generated designs aid in designers' creativity.

Introduction

As we move towards product customization, additive manufacturing, and extended reality environments, the demand for 3D models is ever increasing. The global 3D mapping and modeling market size was valued at USD 13.49 billion in 2020 and is expected to grow at a CAGR of 20.9% over the period 2021-2026 (Intelligence 2021). The increasing need for VFX technologies in movies, 3D applications in games, and the huge shift towards Virtual Reality are major growth factors for the 3D mapping and modeling market. This increasing demand has given rise to multiple 3D design softwares. These softwares help in modeling, analyzing, and translating 3D models. However, there is still a lack of tools that aid in designers' creativity. In this paper, we present an approach for 3D style transfer to bridge this gap.

One of the ways to generate stylized design variants is by leveraging the concept of Neural style transfer (NST) (Jing et al. 2019). A considerable amount of research has been conducted in the field of Neural style transfer for 2D images and videos (Jing et al. 2019). 2D Neural style transfer has seen enormous success in the fields of fashion, gaming, interior designing, virtual reality, etc. In addition, applying style from a 2D image to a 3D object has seen some progress (Berkiten et al. 2017). However, the process of identifying the style of a 3D object and applying it to another 3D object is still an active research area. 3D style transfer can be achieved through transfer of shape, texture or 2D patterns. We propose to advance the existing landscape of style transfer by merging compatible elements from two 3D objects to create stylized 3D objects. In the field of image and video processing, Convolutional neural networks (CNN) have been successfully applied to generate novel stylized images. However, the usage of these methods for 3D objects is not straightforward due to the unstructured representations of 3D models such as point clouds, meshes, etc. Hence, efficient geometric representations are required to use style transfer concepts for three-dimensional shapes. In recent years, we witnessed an increased focus on proposing Deep neural network (DNN) models for 3D object classification (Hanocka et al. 2019), part segmentation (Hanocka et al. 2019), reconstruction, etc. However, there has been very little attention on synthesizing 3D models.

In this work, we propose a 3D style transfer framework, 3DStyleMerge, that transfers style elements from style to content 3D objects. We apply a combination of learning-based and geometric-based approaches to achieve this. We first segment the 3D objects into their components, then compute the compatibility between the components, and finally perform style transfer. Our approach ensures that the functionality of the content 3D object is preserved, by allowing only compatible operations. Next, we demonstrate the results of our style transfer framework. To evaluate our approach, we conduct a user study with 3D designers. Our evaluation results show that our approach effectively generates stylized designs and the generated designs aid in designers' creativity.

The remainder of this paper is structured as follows: Section 2 discusses the related work. In Section 3, we describe our approach for style transfer on 3D models. We describe the dataset and discuss results in Section 4. Finally, Section 5 concludes with future work.

Related Work

Prior works on style transfer mainly focus on 2D-to-2D style transfer on images and videos (Jing et al. 2019). We find few 2D-to-3D style transfer approaches which aim at applying specific textures or colors from 2D samples to 3D objects (Berkiten et al. 2017). Hu et al. (Hu et al. 2017) present a method for defining elements that characterize a given style, where the elements are co-located across the shapes of the style. Stornaiuolo et al. (Stornaiuolo et al. 2020) present a 3D-to-3D topology transfer paradigm based on transformations in 3D space. They develop a 3D conditional Generative Adversarial Network, Vox2Vox, that performs volumet-

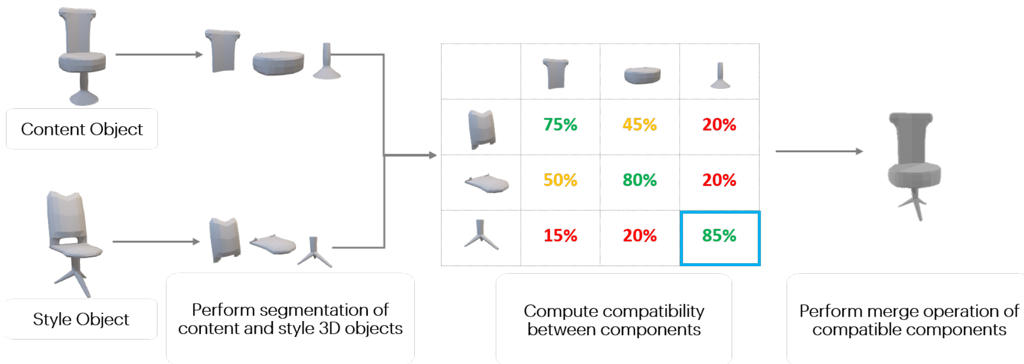


Figure 1: 3DStyleMerge approach

ric transformations to modify the internal structure of any 3D object while maintaining its overall shape. Guan et al. (Guan et al. 2020) develop a 3D shape modeling tool based on functionality-aware model evolution. The tool generates new shapes via part recombination, possibly across object categories. They introduce the concept of functionality partial matching, which analyzes the functional plausibility of new shapes not as a whole, but in parts, with respect to learned functionality models. Liu et al. (Liu and Jacobson 2019) present a 3D stylization algorithm that can deform an input 3D shape into the style of a cube while maintaining the content of the original shape. Ma et al. (Ma et al. 2014) apply the analogy approach to 3D shapes. Rather than simply transferring the style of an exemplar to a base shape, they take as input a source shape similar to the exemplar and a target shape and synthesize a new shape that follows the structure of the target but possesses the style of the exemplar. Their approach requires a correspondence between the source and base shapes and an analogy relationship between the source and target. Lun et al. (Lun et al. 2016) propose a geometric-based approach for synthesizing shapes by transferring style between 3D objects. Given an exemplar and a target shape, the approach generates an output that preserves the target shape's structure and functionality, while minimizing the style distance between the output and the exemplar. They perform element-level geometric operations - substitution, addition, removal, and deformation to achieve style transfer. However, these works require a specific formulation of analogy between the different parts of the analyzed objects that limits their application to few collections of 3D models. Our work is closely related to (Lun et al. 2016) which applies geometric-based approaches to transfer style elements from style 3D object to content 3D object. Unlike Lun et al. (Lun et al. 2016), we apply a combination of learning-based and geometric-based approaches to perform style transfer.

3DStyleMerge: 3D Shape Transfer

In this section, we introduce 3DStyleMerge - our proposed 3D Style merge framework, which transfers style elements from style to content 3D objects. Our approach (as shown in Figure 1) consists of three steps:

1. 3D Part Segmentation: Part Segmentation of 3D content and style objects
2. 3D Shape Compatibility: Evaluation of compatibility between segmented parts
3. Style Merge: Merging components from style to content 3D object

We discuss each of them in the subsequent sub-sections.

3D Part Segmentation

For part segmentation, we leverage the state-of-the-art 3D segmentation approach, MeshCNN, to segment the components of the 3D objects. We apply MeshCNN, a Convolutional neural network designed specifically for triangular meshes, which operates directly on the mesh edges. The network learns to classify the edges into different segments. The network combines specialized convolution and pooling layers that operate on the mesh edges, by leveraging their intrinsic geodesic connections. Mesh Convolutions are applied on every edge and the four edges of its incident triangles, and mesh pooling is applied via an edge collapse operation that retains surface topology, thereby generating new mesh connectivity for the subsequent convolutions. Due to space limitations, we present only the necessary details of MeshCNN. For a detailed understanding, please refer to the paper (Hanocka et al. 2019).

3D Shape Compatibility

We evaluate compatibility between pairs of segments to assess the impact of the merge operation on the functionality of the content object. Shapes are deemed to be more stylistically similar if they share more geometrically similar elements, i.e., common shape characteristics. The compatibility operation is specifically designed to maintain the content object's structure by preserving element connectivity and functionality during merge operations. Earlier approaches (Lun et al. 2016) have focused on compatibility computation by defining several metrics and descriptors to capture features. We use a data-driven approach to measure the compatibility between 3D components. We use Siamese neural network to achieve this. Siamese networks consist of two identical subnetworks with shared weights, i.e., they consist of the same network copied with a Contrastive loss function. Figure 2

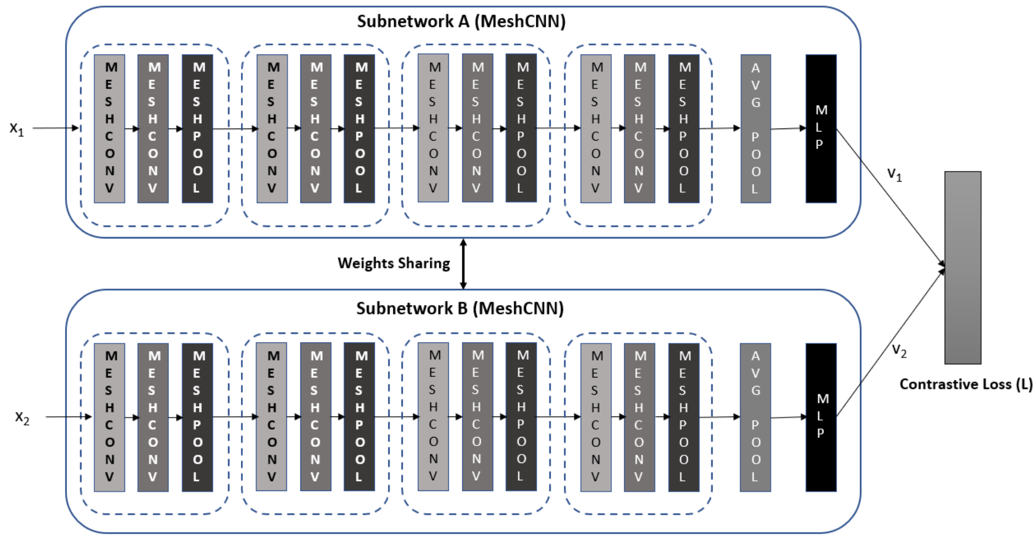


Figure 2: Siamese neural network for part compatibility with MeshCNN (Hanocka et al. 2019) as identical subnetwork

shows the architecture of our proposed Siamese network. We use MeshCNN as our subnetwork to extract the feature representation (embedding) for 3D components. MeshCNN network consists of a series of Mesh Convolution and Mesh Pooling blocks, followed by an Average Pooling and a Linear layer. The components of MeshCNN are discussed in Section 3.1. The training set for a Siamese network consists of triplets (x_1, x_2, y), where x_1 and x_2 are 3D components and $y \in 0, 1$ indicates whether x_1 and x_2 are compatible (y = 1) or not compatible (y = 0). The output of the Linear layer are 100-dimensional embedding representations v_1 and v_2 for input 3D components x_1 and x_2 respectively. The network loss can be formalized as the Contrastive loss function which measures the ability of the function to place compatible parts nearby and keep incompatible parts further apart. The objective of the Contrastive loss function is to minimize the positive pairwise (compatible parts) distance and maximize the negative pairwise (incompatible parts) distance. Mathematically, Contrastive loss L is

$$L = y * ||v_1 - v_2||^2 + (1-y) * max(0, m - ||v_1 - v_2||)^2 \quad (1)$$

where v_1 is the vector representation (embedding) for input x_1, v_2 is the vector representation (embedding) for input x_2, and y is the label (0/1).

Style Merge

After calculating the compatibility of segments in the content and style objects, the next step is to substitute a compatible style segment in the content object. We achieve this by adopting a geometric based approach using Blender Python API (Blender 2021). The below 4 steps are involved to achieve this merge operation:

1. Convert content and style segments to a uniform scale
2. Align the center of geometry of the content segment with its substitute compatible style segment
3. Substitute the content segment with the style segment
4. Fill any gaps that may appear after substitution

Experiments and Results

In this section, we discuss the dataset and results.

Dataset

We experiment with the well-known Shape COSEG dataset for part segmentation, which is a collection of CAD models belonging to 10 object categories. The ground truth labels are only available for two categories - chairs and vases, which consist of 700 objects.

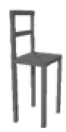
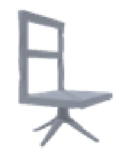
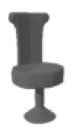
There is no open-source dataset available for computing compatibility. We curate our own dataset by manually tagging pairs of 3D components as compatible (y=1) or incompatible (y=0) with the help of 3D designers and subject matter experts. We collect datasets from various open-source databases such as GrabCAD (GrabCAD 2021), 3D Warehouse, etc. The dataset consists of a total of 500 pairs of data points, of which 250 pairs are compatible and the other 250 pairs are incompatible.

Evaluation and Results

Our approach consists of three major components - 3D Part Segmentation, 3D Shape Compatibility, and 3D Style Merge. In this section, we discuss the evaluation of these components.

3D Part Segmentation: We evaluate the performance of the part segmentation network on the COSEG dataset. We split the dataset into train (85%) and test (15%) sets for each category. We observe an accuracy of 99% for chairs and 97% for vases on the test set (same as reported in (Hanocka et al. 2019)).

Table 1: 3D Style Merge results for a single style source

Content 3D Object	Style 3D Object	Stylized 3D Object
		
		
		
		
		

3D Shape compatibility: We evaluate the performance of our 3D Shape compatibility model. We split the dataset into train (80%) and test (20%) sets. We observe an accuracy of 92.4% on the training set and 88.2% on the test set. The results show that our approach is effectively able to find compatible pairs.

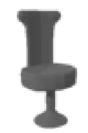
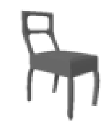
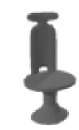
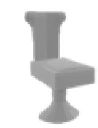
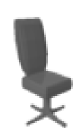
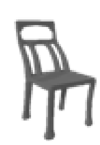
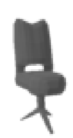
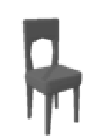
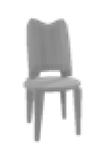
3D Style Merge: We conduct a user study with ten designers to qualitatively evaluate the generated style merged 3D objects. The designers have experience working on various 2D and 3D design tools such as Autodesk Maya, Zbrush, Photoshop, Arnold, Substance Painter, Autodesk Fusion 360, Rhinoceros 3D, AutoCAD, Blender, Autodesk 3DS Max, Keyshot and Marmoset. We present 10 randomly sampled style merged designs (along with their content and style 3D models) to each designer and ask them to provide their feedback along two aspects:

1. Do you see a style transfer from style to content 3D object (Yes/No)?
2. Rate the clarity of output (on a Likert scale of 1-5)

We apply majority voting for categorical parameters and averaging for numerical parameters to aggregate the designers' responses. We observe that all the designers see a style transfer from the style to content 3D objects. The designers' rating for clarity of outputs has a median value of 4.3. The designers are aware of tools that can achieve style transfer for 2D images. However, they are not aware of any such tools for automated 3D style transfer. We measure the inter-rater agreement amongst designers on whether they see a style transfer using Fleiss' Kappa (McHugh 2012). We observe the κ coefficient to be 0.98 which indicates a level of strong agreement among the designers. The results demonstrate that our approach is effectively able to generate stylized designs by merging the elements from a style to a content 3D object.

We perform a quantitative analysis to evaluate the quality of the stylized 3D object. There are certain metrics proposed to evaluate the stylized output of 2D style transfer (Jing et al. 2019). However, there is no single metric established to evaluate the output of 3D style transfer. In recent years, there has been an attempt to propose metrics to evaluate the quality of 3D meshes using reference (or ground truth) samples

Table 2: 3D Style Merge results for multiple style sources

Content 3D Object	Style 3D Object - 1	Style 3D Object - 2	Stylized 3D Object
			
			
			
			
			

(Abouelaziz, El Hassouni, and Cherifi 2016). These metrics mainly leverage the spatial deformation between the reference and distorted samples. However, these metrics cannot be directly applied to our case, as they capture the distortion in the generated sample with respect to the reference object. We propose a metric, Quality score, to evaluate the quality of style transfer. We apply a supervised approach to classify the 3D mesh model as “Noisy” or “Clean”. We leverage the MeshCNN classification network (as shown in Subnetwork of Figure 2), except the last layer which we change from Softmax to Sigmoid activation function to achieve this. We curate a dataset of noisy and clean meshes from various open-source databases - LIRIS/EPFL General-Purpose Database (Lavoué et al. 2006) and GrabCAD (GrabCAD 2021). The dataset contains 600 samples (300 samples for noisy meshes and 300 samples for clean meshes), belonging to different categories such as vases, chair, boxes, etc. Out of the 600 samples, 100 samples were already labelled and 3D designers manually labelled the remaining 500 samples as “Noisy” or “Clean”. We randomly split the dataset into train (80%) and test (20%) sets. We train the model for 100 epochs. We observe a training accuracy of 86.8% and a test accuracy of 83.2%. The trained model is used to predict whether the stylized 3D model (generated by our style merge approach) is “Noisy” or “Clean”. We define Quality score as the predicted probability of the stylized 3D model being “Clean”. Our approach doesn’t require any reference object to assess the quality of a 3D mesh. We observe the mean Quality score as 0.82 (with the maximum score being 0.87 and the minimum score being 0.74) for stylized 3D designs generated by our approach. The evaluated Quality score implies that the proposed framework has successfully transferred style elements to the content with good quality. Our work is closely related to (Lun et al. 2016), which applies geometric based approach for style transfer. They evaluated their results through qualitative analysis by conducting user studies. They did not propose any metrics to evaluate the quality of their outputs. As there is no baseline metrics, we are unable to compare our results with the existing work.

Conclusion and Future Work

In this work, we propose a 3D style transfer framework, 3DStyleMerge, that transfers style elements from style to content 3D objects. We apply a combination of learning-based and geometric-based approaches to perform style transfer. Our approach ensures that the functionality of the content 3D object is preserved by allowing only compatible operations. We demonstrate the results of our style transfer framework. We conduct a user study with 3D designers to evaluate our stylized 3D designs. Our evaluation results demonstrate that our approach effectively generates stylized designs and the generated designs aid in designers' creativity. As future work, we plan to improve the quality of the generated stylized 3D models.

References

Abouelaziz, I.; El Hassouni, M.; and Cherifi, H. 2016. No-reference 3d mesh quality assessment based on dihedral angles model and support vector regression. In *International Conference on Image and Signal Processing*, 369–377. Springer.

Berkiten, S.; Halber, M.; Solomon, J.; Ma, C.; Li, H.; and Rusinkiewicz, S. 2017. Learning detail transfer based on geometric features. In *Computer Graphics Forum*, volume 36, 361–373. Wiley Online Library.

Blender. 2021. Blender Python API. https://docs.blender.org/api/current/index.html. [Online; accessed 10-Jun-2021].

GrabCAD. 2021. GrabCAD. https://grabcad.com/. [Online; accessed 10-Jun-2021].

Guan, Y.; Liu, H.; Liu, K.; Yin, K.; Hu, R.; van Kaick, O.; Zhang, Y.; Yumer, E.; Carr, N.; Mech, R.; et al. 2020. 3D Shape Generation via Functionality-Aware Model Evolution. *arXiv preprint arXiv:2005.04464*.

Hanocka, R.; Hertz, A.; Fish, N.; Giryes, R.; Fleishman, S.; and Cohen-Or, D. 2019. Meshcnn: a network with an edge. *ACM Transactions on Graphics (TOG)*, 38(4): 1–12.

Hu, R.; Li, W.; Kaick, O. V.; Huang, H.; Averkiou, M.; Cohen-Or, D.; and Zhang, H. 2017. Co-locating style-defining elements on 3d shapes. *ACM Transactions on Graphics (TOG)*, 36(3): 1–15.

Intelligence, M. 2021. 3D MAPPING AND 3D MODELLING MARKET. https://www.mordorintelligence.com/industry-reports/3d-mapping-and-3d-modelling. [Online; accessed 10-Mar-2021].

Jing, Y.; Yang, Y.; Feng, Z.; Ye, J.; Yu, Y.; and Song, M. 2019. Neural style transfer: A review. *IEEE transactions on visualization and computer graphics*.

Lavoué, G.; Gelasca, E. D.; Dupont, F.; Baskurt, A.; and Ebrahimi, T. 2006. Perceptually driven 3D distance metrics with application to watermarking. In *Applications of Digital Image Processing XXIX*, volume 6312, 63120L. International Society for Optics and Photonics.

Liu, H.-T. D.; and Jacobson, A. 2019. Cubic stylization. *arXiv preprint arXiv:1910.02926*.

Lun, Z.; Kalogerakis, E.; Wang, R.; and Sheffer, A. 2016. Functionality preserving shape style transfer. *ACM Transactions on Graphics (TOG)*, 35(6): 1–14.

Ma, C.; Huang, H.; Sheffer, A.; Kalogerakis, E.; and Wang, R. 2014. Analogy-driven 3D style transfer. In *Computer Graphics Forum*, volume 33, 175–184. Wiley Online Library.

McHugh, M. L. 2012. Interrater reliability: the kappa statistic. *Biochemia medica*, 22(3): 276–282.

Stornaiuolo, L.; Dehmamy, N.; Barabási, A.-L.; and Martino, M. 2020. 3D Topology Transformation with Generative Adversarial Networks. *arXiv preprint arXiv:2007.03532*.