Honda Research Institute EU
Honda Research Institute Europe GmbH
https://www.honda-ri.de/

Standardization Of Gram Matrix For Improved 3D Neural Style Transfer

Timo Friedrich, Stefan Menzel

2019

Preprint:

This is an accepted article published in IEEE Symposium Series on Computational Intelligence. The final authenticated version is available online at: https://doi.org/10.1109/SSCI44817.2019.9002780 Copyright 2019 IEEE

Standardization of Gram Matrix for Improved 3D Neural Style Transfer

Timo Friedrich and Stefan Menzel
Honda Research Institute Europe, Offenbach/Main, Germany
Email: timo.friedrich@honda-ri.de; stefan.menzel@honda-ri.de

***Abstract*—Neural Style Transfer based on convolutional neural networks has produced visually appealing results for image and video data in the recent years where e.g. the content of a photo and the style of a painting are merged to a novel piece of digital art. In practical engineering development, we utilize 3D objects as standard for optimizing digital shapes. Since these objects can be represented as binary 3D voxel representation, we propose to extend the Neural Style Transfer method to 3D geometries in analogy to 2D pixel representations. In a series of experiments, we first evaluate traditional Neural Style Transfer on 2D binary monochromatic images. We show that this method produces reasonable results on binary images lacking color information and even improve them by introducing a standardized Gram matrix based loss function for style. For an application of Neural Style Transfer on 3D voxel primitives, we trained several classifier networks demonstrating the importance of a meaningful convolutional network architecture. The standardization of the Gram matrix again strongly contributes to visually improved, less noisy results. We conclude that Neural Style Transfer extended by a standardization of the Gram matrix is a promising approach for generating novel 3D voxelized objects and expect future improvements with increasing graphics memory availability for finer object resolutions.**

Keywords*—Neural Style Transfer; voxel; binary models; style similarity.

I. Introduction

Automated processes for a rapid generation of 3D design alternatives offer a high value to the automotive development process. Exploring these alternatives is meaningful for visual studies of potential design concepts following a new style direction or for computational optimization tasks evaluating technical performance of design variations. Based on deep learning techniques, in the recent years Neural Style Transfer [1] has been proposed and successfully applied to 2D image manipulation, e.g. for generating novel digital art based on the content of a first image (photograph) and the style of a second image (painting) [2]–[4]. Further research utilized this fundamental Style Transfer (ST) idea to motion data for an automated process of character motion conversion [5]. However, only few work has been carried out in the direction of developing methods for style transfer in 3D object space. Kato et al. [6] followed a hybrid 2D/3D Neural Style Transfer utilizing 2D image style sources and applying them to 3D mesh content targets. Sykora et al. [7] took a similar approach and transferred texture samples onto 3D models in real time and achieved dynamic animations with temporal stability. Both methods alter a 2D manifold in 3D space based on a 2D style transfer. For style transfer in 3D space, Ma et al. [8] proposed an analogy-driven method which calculates shape correspondences between source objects and the objects to be modified, and afterwards optimizes the transformation for a high-quality style change of repetitive geometries, like furniture and building examples.

One important aspect for 3D Neural Style Transfer is the underlying shape representation. In our previous work [9], we have discussed the suitability of several potential 3D geometry representations, especially focusing on voxels or point clouds as starting point for 3D Neural Style Transfer by extending the 2D image manipulation framework as mentioned before. However, even though the similarity between pixels and voxels seems straight forward, in the context of Neural Style Transfer, we have to take care of some fundamental differences. The increase by one dimension results in a massive increase of memory requirements which currently limits the models to a maximum expansion of 64 voxels per dimension. Furthermore, voxel models do not hold all elements of art [10], e.g., the binary 3D voxel models (spheres and boxes) used in our experiments neglect color information and texture. Despite these shortcomings, in this paper we will show that 3D Neural Style Transfer can generate reasonable object variations by carefully choosing the classification network. The standardization of the Gram matrix leads to improved, i.e. more interpretable and less noisy, shapes, but however points to the necessity of increasing the object resolution. The paper is organized as follows. In Section II, the general framework for 2D/3D style transfer is detailed. Section III describes our experiments and results for a 2D style transfer of binary images to validate our framework and investigate the influence of lacking texture and color information in images. It also introduces our proposed standardization of the Gram matrix for improved style transfer. In Section IV, the framework is extended to 3D Neural Style transfer for voxel shape primitives and highlights our findings. Section V concludes the paper.

II. Neural Style Transfer

The target of Neural Style Transfer is the transfer of artistic style features from a style source, which might be a photo or a painting, onto a target source [1]–[4]. Then, the generated picture still depicts the recognizable content from the content source image but features characteristics like the texture and coloring of the style source image. The optimization process is based on a loss function (1) which uses the activation maps of

a pre-trained deep convolutional neural network. Usually, the VGG19 network [11] is the preferred choice for the production of the required activation maps that are the fundamental basis for Neural Style Transfer.

$$L_{total} = \alpha L_{content} + \beta L_{style} \tag{1}$$

The total loss is a weighted sum of a first term representing the perceptual recognizability of the original content and a second term representing the similarity of artistic style features between the output and the style input. The output image is iteratively adapted according to the gradient calculated by backpropagation, which leaves the network weights untouched and alters the pixel image data instead.

$$L_{content}(p, x, l) = \frac{1}{2} \sum_{(i,j)} (F_{ij}^l - P_{ij}^l)^2 \tag{2}$$

The first sub-loss for the content similarity (2) is based on the activations of a convolutional layer. The activations F_{ij}^l and P_{ij}^l of the content target $\overrightarrow{p}$ and the to be generated output image $\overrightarrow{x}$ are compared directly by applying the mean squared error. Previously, Gatys et al. [1] have shown that higher layers for l store an abstracted view of the content which is needed for neural style transfer because it leads to a less strict loss term for the content. This is necessary because the content loss has to be balanced with the style loss (3) in order to produce images which differ on the pixel level but depict the same content.

The style loss is based on multiple lower and higher layers in order to be able to encode the similarity of lower artistic features like brush stroke and higher features like image composition. Each layer based style loss is consolidated to a weighted total loss.

$$L_{style}(\overrightarrow{a}, \overrightarrow{x}) = \sum_{l=0}^{L} (\omega E_l) \tag{3}$$

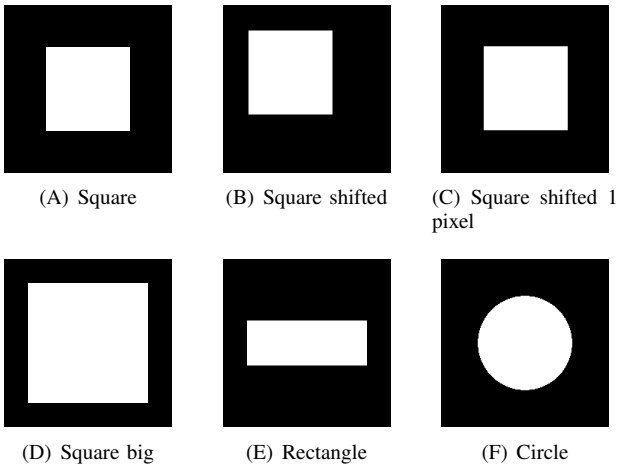
In contrast to the content loss, the style loss is not based on the convolutional activations directly, but on the Gram matrices of the activations (2). Thereby, the style loss becomes position invariant and only compares the statistics of artistic style features of a style target $\overrightarrow{a}$ and the generated image $\overrightarrow{x}$.

$$G_{ij}^l = \sum_k (F_{ik}^l F_{jk}^l) \tag{4}$$

$F_i k^l$ and $F_j k^l$ are the vectorized feature maps of layer l. The Gram matrix G_{ij}^l represents the feature correlations of a single layer. Based on the Gram matrix, the single layer loss is defined as shown in the following equation where A_{ij}^l is the Gram matrix of the activations of $\overrightarrow{x}$.

$$E_l = \frac{1}{4N_l^2 M_l^2} \sum_{(i,j)} (G_{ij}^l - A_{ij}^l)^2 \tag{5}$$

N_l is the number of filters or feature maps per layer whereas M_l is the length of the vectorized feature map.



(A) Square (B) Square shifted (C) Square shifted 1 pixel

(D) Square big (E) Rectangle (F) Circle

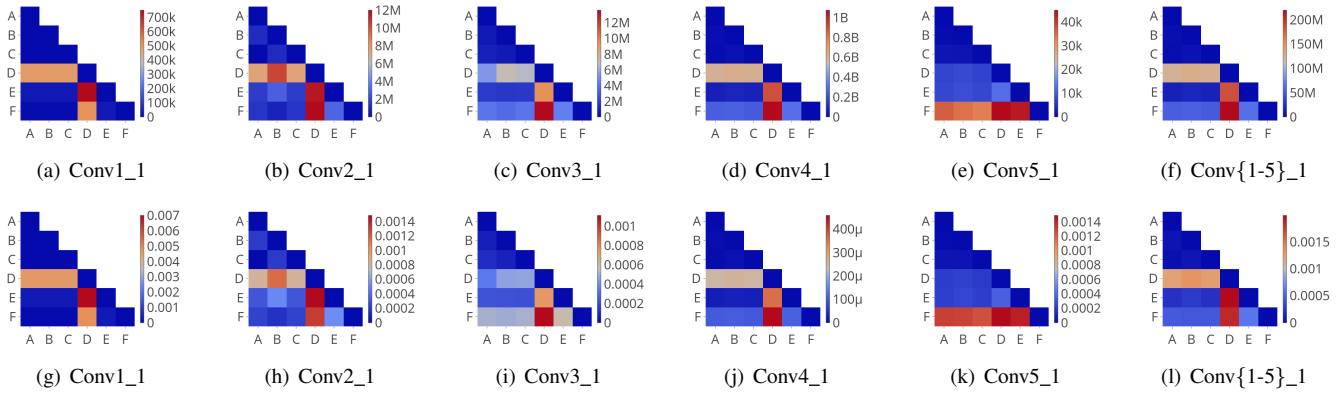
Fig. 1. 2D binary geometric primitives used for Style Transfer evaluation. The images are encoded as RGB images in order to be compatible to the VGG19 network.

An exhaustive overview of the advances in Neural Style Transfer for 2D image data has been made public by Jing [2]. Newer research on Style Transfer published by Adobe Research [3] describes the Universal Style Transfer. The auto encoder based fast forward approach produces subjectively compelling results while accepting arbitrary style targets without a lengthy training in contrast to previous fast forward approaches. Recently, first efforts have begun to collect and organize all style transfer variants under a common framework [12]. The proposed architecture shall enable the mixture of available algorithms and loss functions in a modular setup. Following the idea of transferring the 2D Neural Style Transfer concept to 3D space, in the next section we will explore the capabilities of the CNN framework for 2D binary images before moving on to 3D voxel shape primitives.

III. 2D Style Transfer with Binary Images

Regular 2D Pixel images and 3D voxel models have many properties in common but also important differences in the context of Neural Style Transfer. Apart from the additional spatial dimension, voxel models usually only encode the absence and presence of material in cells of a regular grid with zeros and ones. Hence, we handle single channel binary data instead of three channel color data of 8-bit integers. Instead of directly investigating Neural Style Transfer on 3D voxel data, we first thoroughly investigate the concept of Style Transfer on artificially created 2D binary image data in combination with the 2D Style Transfer environment which has been successfully realized and published in literature [1]. In prior experiments we had successfully reproduced style transfer results which were in line with the published results by Gatys et al.

For our tests, we have prepared a set of binary image primitives as depicted in Figure 1. We apply the preliminary tests and Neural Style Transfer according to Gatys et al. [1] on these simple shapes.



(a) Conv1_1 (b) Conv2_1 (c) Conv3_1 (d) Conv4_1 (e) Conv5_1 (f) Conv{1-5}_1

(g) Conv1_1 (h) Conv2_1 (i) Conv3_1 (j) Conv4_1 (k) Conv5_1 (l) Conv{1-5}_1

Fig. 2. Correlation plots for the 2D geometric primitives as shown in Figure 1 based on the style loss function. Maps (a-f) rely on equations (4) and (5). Maps (g-l) rely on equations (6) and (7).

A. 2D Binary Style Loss Evaluation

The primitive's images are scaled to the native VGG19 input resolution. Additionally, the binary single layer image is copied accross all three color channels of VGG19's input layer. We accept this compromise even though we realize the optimization will not alter all channels identically which will result in at least partly colored generated images. Also, we remember that VGG19 was trained on RGB image data, thus the convolutional filters are not optimized for binary images.

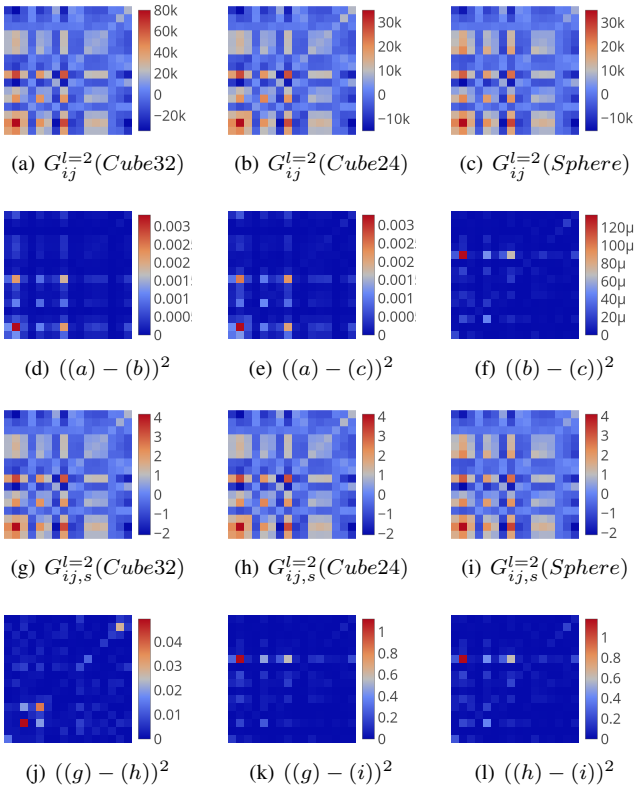
In order to check the feasibility of the style loss function (3) in combination with our 2D shape primitives, we plot the style loss of each primitive pair as a correlation matrix in Figure 2 (a)-(f). Unfortunately, the maps do not correlate to an intuitive human sorting. E.g., the big square (D) is less similar to the square (A) than the circle (F) for the convolutional layers 1-4. Obviously, the influence of the size of the filled area by the primitive seems to outweigh the influence of the shape of the circumference.

As shown with our 3D voxel primitives in Figure 3, this can be partly described by the sensitivity of the Gram matrix to the model scaling. For the sake of visualization, we have produced these figures with a small adapted 3D voxel network with less filters per convolutional layer. The Gram matrices of the small and big cube share the same pattern but have very different absolute values in contrast to the matrices of the small cube and the sphere. Hence, the style loss value is smaller for the small cube compared to the sphere instead of the small cube compared to the big cube. Because of this reason, we introduce the standardized Gram matrix approach.

$$G^l_{ij,s} = \frac{G^l_{ij} - \mu(G^l_{ij})}{\sigma(G^l_{ij})} \tag{6}$$

$$E_{l,s} = \sum_{(i,j)} (G^l_{ij,s} - A^l_{ij,s})^2 \tag{7}$$

As shown in Equation (7), we spare the scaling by the size of the corresponding activation map. As a result, the difference of the standardized Gram matrices for the previous example looks more intuitive as shown in Figure 3 (j-l).



(a) $G^{l=2}_{ij}(Cube32)$ (b) $G^{l=2}_{ij}(Cube24)$ (c) $G^{l=2}_{ij}(Sphere)$

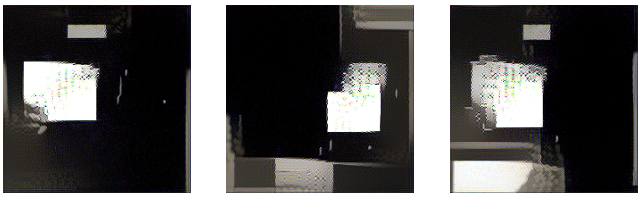
(d) $((a)-(b))^2$ (e) $((a)-(c))^2$ (f) $((b)-(c))^2$

(g) $G^{l=2}_{ij,s}(Cube32)$ (h) $G^{l=2}_{ij,s}(Cube24)$ (i) $G^{l=2}_{ij,s}(Sphere)$

(j) $((g)-(h))^2$ (k) $((g)-(i))^2$ (l) $((h)-(i))^2$

Fig. 3. 3D Gram Matrix Test for Layer 2 of ST_Enc_Pool. We used the small network for visualization for the sake of readability. (a-c) are based on Equation (4), (g-i) on Equation (6).

Figure 2 (g-l) show the regenerated correlation plots for the primitives based on the previously introduced equations (6) and (7). For the 2D approach, the standardization did not have the expected effect. The similarity except for layer 3 (Fig. 2 (i)) and 5 (Fig. 2 (k)) remains counter intuitive. Nevertheless, the absolute values per layer are now in a comparable order of magnitude for the standardized approach. In contrast to the 2D style loss test, the standardization has a significantly stronger impact for the identical test in combination with the 3D voxel



(a) Square (b) Square shifted (c) Square shifted 1px



(d) Square big (e) Rectangle (f) Circle

Fig. 4. Representative style reconstruction results for each 2D geometric primitive with the original style loss by Gatys.



(a) Square (b) Square shifted (c) Square shifted 1px



(d) Square big (e) Rectangle (f) Circle

Fig. 5. Representative style reconstruction results for each 2D geometric primitive with the adapted standardized Gram matrix based style loss.

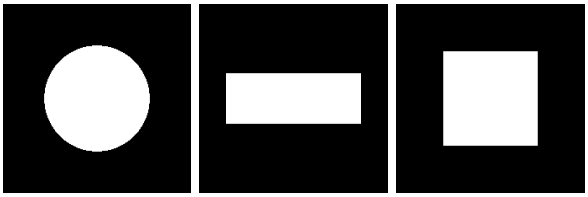
primitives as shown later in Figure 9.

B. 2D Binary Style Reconstruction

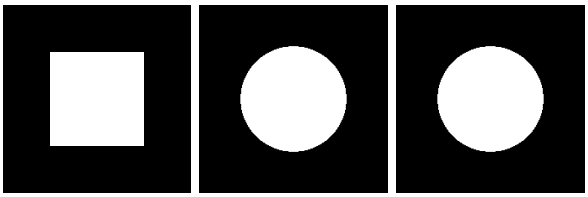
According to the style reconstruction test introduced by Gatys et al. [1], we reproduce the reconstructions with our geometric primitives. The results are depicted in Figure 4. We use a uniform random noise initial image for $\vec{x}$, the Adam optimizer and VGG19's layers Conv{1-5}_1, each weighted equally. The generated images depict artistic style features of the primitives as 90° angles and straight lines for the square like targets, and round edges for the circle target.

Additionally, Figure 5 shows the results for the standardized Gram Matrix approach. All hyper parameters had been fixed between Figure 4 and 5. In comparison, the second picture set shows less noise and clearer style features.

Based on these outcomes, we state that the artistic style loss function by Gatys et al. can be applied on binary image data even though several elements of art [10] are missing.



(a) Content Targets



(b) Style Targets



(c) Results: Original ST

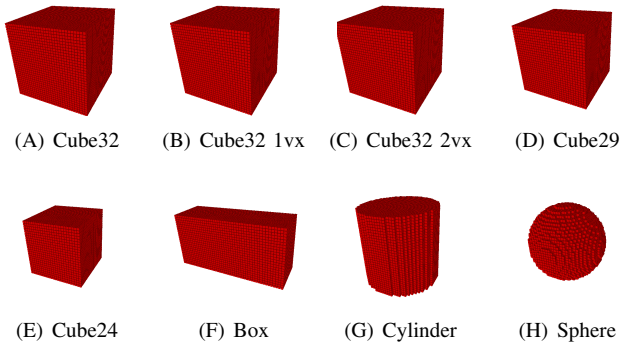


(d) Results: Standardized Gram Matrix ST

Fig. 6. Neural Style Transfer results for the 2D geometric primitives. Based on VGG19's layers Conv{1-4}_1.

C. 2D Binary Style Transfer

Based on the original published algorithm as well as the algorithm extended with the standardization of the Gram matrix, we perform Neural Style Transfer with a subset of our 2D geometric primitives, see Figure 6. Both versions produce reasonable results with the exact same hyper parameter setup, except for the weights α and β that have to be manually tuned. The generated images based on Gatys et al.'s original equation show noticeably more noise artifacts than the results based on our extended algorithm as depicted in Figure 6 (c) and (d). Also, the images of the last row differ in terms of the scaling of the cubic features, meaning the image (c) shows more and smaller 'steps'. This is most likely due to the changed relative weighting between the convolutional layers used for the style loss computation and can be easily adapted to personal preference. It is worth to mention that our geometric primitive based Style Transfer has a strong visual similarity to Ma's [8] 2D examples. Since Neural Style Transfer does not rely on explicit analogy detection, we do not need a source model. Hence, we can use arbitrary image pairs.



(A) Cube32 (B) Cube32 1vx (C) Cube32 2vx (D) Cube29

(E) Cube24 (F) Box (G) Cylinder (H) Sphere

Fig. 7. 3D geometric primitives used for voxel based Neural Style Transfer evaluation. Models A-C are identical but shifted by 0,1 or 2 voxels in one dimension. The associated numbers encode the edge length of the models.

IV. 3D Voxel Style Transfer

Extending the Neural Style Transfer from 2D image data based Neural Style Transfer to 3D voxel based Style Transfer creates several challenges. Until now, only a handful of voxel based classification networks have been published [13]–[16]. Thus, there are no 'standard' networks like the VGG19 available for voxel data. Furthermore, the sparsely available networks are usually trained on voxel models with a resolution of 32^3 voxels. We have decided to use a minimum resolution of 64^3 voxels for our Style Transfer experiments since we believe this is the minimum resolution with which style is recognizable besides the block-like style already inherent in coarse voxel models. Thus, our memory requirement is considerable higher during training and inference compared to the available network architectures. We have voxelized the aligned ModelNet40[1] dataset. Obvious broken voxel models, like empty models due to 2D source meshes, have been removed from the dataset. During Training, we use mirroring and jitter as data augmentation similar to Maturana [14].

A. Classification Network

Our initial setup for this work was based on the classification networks ST_VoxNet64 and ST_Enc in Table I. The networks are closely related to the VoxNet [14] architecture and the encoder part of the variational Autoencoder by Borck et al. [16]. Only minor modifications were necessary for our implementations, e.g. the input layer has to handle the doubled resolution per axis. Additionally, we added extra layers in order to downsample to the same size right before the fully connected layers.

In order to stay close to the 2D setup as described in the previous sections where VGG19 is utilized, we decided to also use networks with a simple feed forward setup without any parallel paths like the ones used in the Voxception-ResNet by Brock et al. [16]. All networks in this work were trained using the Adam optimizer, with batch normalization and data augmentation. We compare the performance of our networks to known networks from the ModelNet40 challenge [13] and a KNN classifier as a baseline, see Table I. During the experiments like style reconstruction, it has become clear that the classifier network architecture is essential for the quality of the generated voxel models. We have observed the following issues.

[1]http://modelnet.cs.princeton.edu/

[2]We have chosen the one-view setup of VRN in this context since our models are one-view models too. Brock [16] achieves an accuracy of 95.5% with his best ensemble model.

TABLE I
Voxel Classificator Accuracy

Classifier	ModelNet40 Accuracy	Comment
Distance KNN	72%	64^3, 5 neighbors
VoxNet [14]	83%	32^3
VRN [16] [2]	89%	32^3
ST_Enc	84%	64^3, aligned
ST_Enc_Pool	84%	64^3, aligned, no conv. striding
ST_Enc_Pool_big	86%	64^3, aligned, no conv. striding
ST_VoxNet64	85%	64^3, aligned

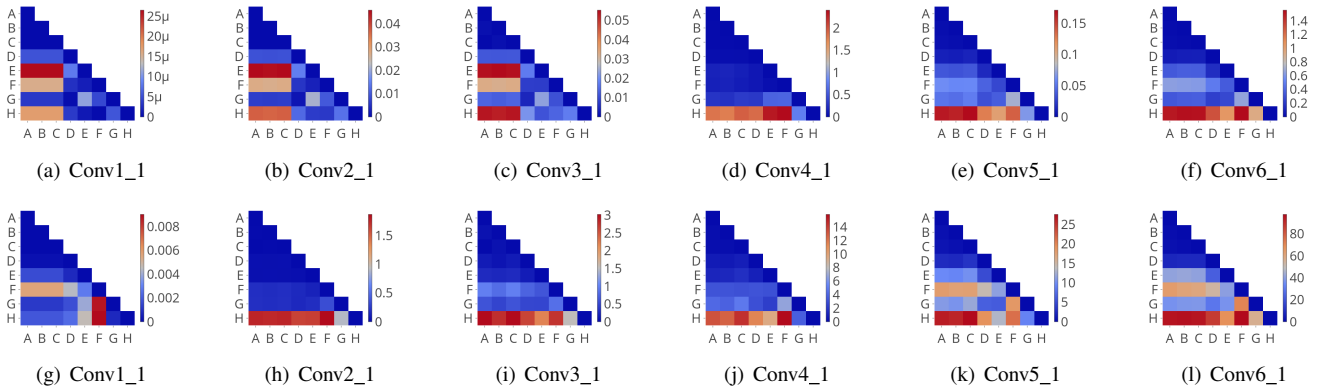


(a) (b)

Fig. 8. Style loss correlation plots showing the incompatibility of convolutional striding >1 with Neural Style Transfer. (a) Conv_1 layer with $stride = (1, 1, 1)$ and (b) Conv_2 layer with $stride = (2, 2, 2)$ of the network ST_Enc.

First, the classification network should provide enough convolutional layers with reasonable big filter banks. We have observed multiple times that small and less activation maps in combination with Neural Style Transfer lead to very noisy generated models. Also in case of style reconstruction, the generated voxel geometries do not show distinct style features of the target voxel model.

Second, downsampling by convolution with a stride bigger than one should not be used. We show in Figure 8 the style similarity correlations maps of two consecutive layers from our ST_Enc network. The first layer downsamples each axis of the data roughly by a factor of two with a stride of two for each dimension. In order to proof our statement, we look at the voxel primitives A, B and C, which are all cubes with an edge length of 32 voxels. B and C are shifted by one and two voxels in one dimension. For the first layer, see Figure 8 (a), these three models show a to be expected loss value of zero. In contrast, Figure 8 (b) shows a non-zero value for the cube shifted by one voxel compared to the other two cubes. Due to the downsampling by the convolution in between the layers, the second layer receives a very different input in case of the shifted cube by one voxel. The reason for this is that the filter hits the faces and edges of the cube at different positions



(a) Conv1_1 (b) Conv2_1 (c) Conv3_1 (d) Conv4_1 (e) Conv5_1 (f) Conv6_1

(g) Conv1_1 (h) Conv2_1 (i) Conv3_1 (j) Conv4_1 (k) Conv5_1 (l) Conv6_1

Fig. 9. Correlation plots for the 3D geometric primitives as shown in Figure 7 based on the style loss function. Maps (a-f) rely on equations (4) and (5). Maps (g-l) rely on equations (6) and (7).

TABLE II
ST_ENC_POOL_BIG LAYOUT

Layer	Size	Configuration
FC_Out	(40)	SoftMax
FC_2	(64)	Elu, BN
FC_1	(128)	Elu, BN
Conv_6	(7,7,7,128)	Elu, BN
Conv_5	(9,9,9,128)	Elu, BN
Conv_4	(11,11,11,64)	Elu, BN
Conv_3	(13,13,13,64)	Elu, BN, maxPool
Conv_2	(29,29,29,32)	Elu, BN
Conv_1	(31,31,31,32)	Elu, BN, maxPool
Input	(64,64,64,1)	Voxel Model

because the striding is larger than one. Hence, the output of the convolution is very different. In order to prevent this issue, we set all convolutional striding to one and add maxpool layers for downsampling. Based on our observations, we build the ST_Enc_Pool_big network with the architecture shown in Table II. If not stated otherwise, we use this network for all voxel based results in this work. We achieved a classification accuracy of around 86% on the ModelNet40 test set. Since it was not the goal to generate a new state of the art voxel classifier but a potentially suited architecture for the Neural Style Transfer context, we are satisfied with this reasonable accuracy at this point in time.

B. 3D Style Loss Evaluation

As discussed in Section III-A, we also generate the correlation maps for our voxel primitives in the Figure 9. The maps (a)-(f) again show the results for the standard neural style loss function while maps (g)-(l) are based on the proposed equations (6) and (7) including Gram matrix standardization. In contrast to the 2D setup, we notice a much stronger effect of the standardized Gram matrix approach. In general, the correlation maps in Figure 9 (g)-(l) show a more intuitive order except for the first map. But compared to the absolute scale of the other five maps, the whole map for layer Conv1_1 can

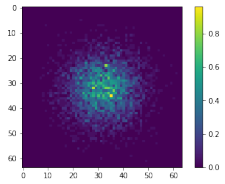


Fig. 10. Planar central slice of the initial voxel model for $\vec{x}$ used for style reconstruction optimizations.

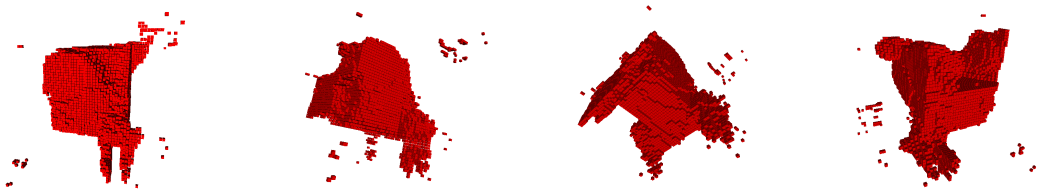
be interpreted as basically zero. Especially, the style loss for the standard cube (A) and the small cube (E) have improved. Nevertheless, some correlation results remain disputable. E.g. the lengthy box (F) is considered to be less similar to the standard cube (A) than the cylinder (G). Even though the geometric features of the box and cube are identical, it requires further research to decide when the size and spatial expansion of an object needs to be considered.

C. 3D Style Reconstruction

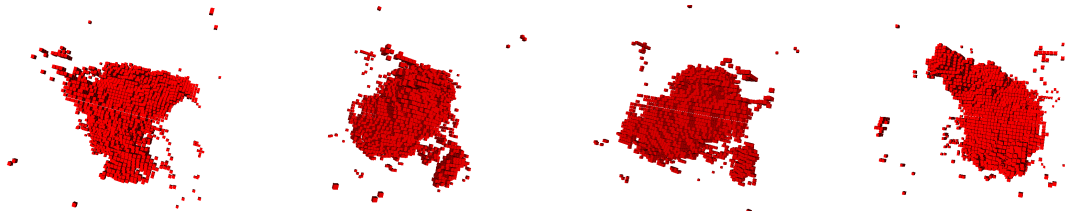
According to Section III-B, we perform the reconstruction tests on two of our voxel primitives shown in Figure 7(A) and 7(H). From our experimental experiences the best results for the reconstruction tests are achieved based on an initial voxel model $\vec{x}$ set to a Gauss like "cloud", see Figure 10, positioned at the center of the whole voxel space. In order to produce the reconstructions shown in Figure 11, we have chosen layers Conv_{1-4} of our ST_Enc_Pool_big network. Similar to our reconstruction results for the 2D primitives, we again observe less noisy results with our standardized Gram matrix approach. Still, both approaches produce reasonable voxel models for the given task. The style features of the cube (A) are better recognizable in general compared to the sphere (H) based counterparts. Flat surfaces and 90° angles are clearly visible whereas the roundish structures of the sphere are less clear but still present.

D. 3D Style Transfer

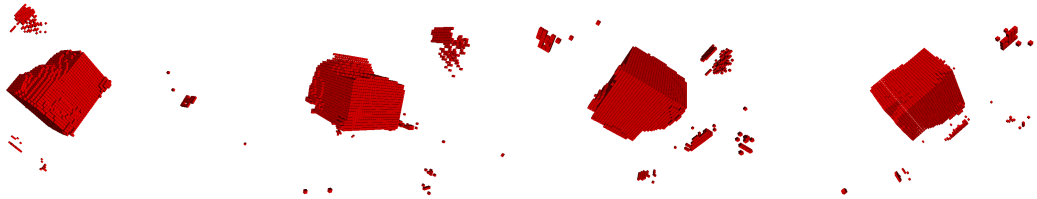
One explanation for this difference in clarity of low level features in case of the sphere compared to the cube might



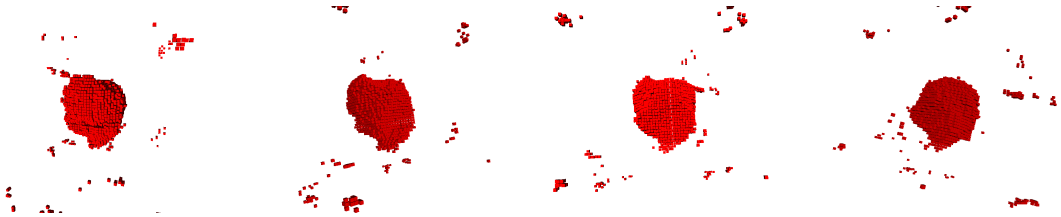
(a) Style Reconstruction: Cube, original Neural Style Transfer



(b) Style Reconstruction: Sphere, original Neural Style Transfer



(c) Style Reconstruction: Cube, standardized Gram matrix approach

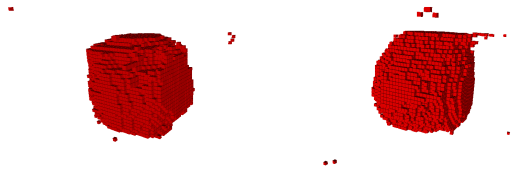


(d) Style Reconstruction: Sphere, standardized Gram matrix approach

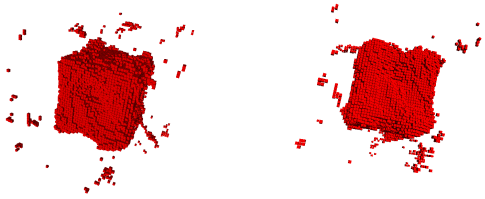
Fig. 11. Different view angles on the voxel style reconstruction results. Models (a-b) are based on equations (4) and (5) and models (c-d) on equations (6) and (7).



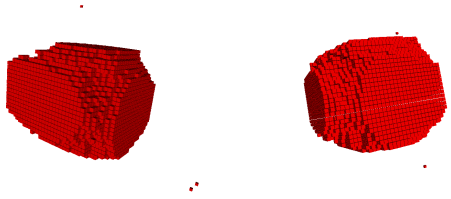
(a) Content: Cube, Style: Sphere, original Neural Style Transfer



(b) Content: Sphere, Style: Cube, original Neural Style Transfer



(c) Content: Cube, Style: Sphere, standardized Gram matrix approach



(d) Content: Sphere, Style: Cube, standardized Gram matrix approach

Fig. 12. Voxel based Neural Style Transfer results for Cube and Sphere pairs. Models (a-b) are based on equations (4) and (5), and models (c-d) are based on equations (6) and (7).

be the usage of low layers from the classification network in combination with the different surface properties of

the models. The cube shows basically three different surface features, namely corners, edges and flat area, on which the network filters are convoluted. Hence, the reconstructions are quite limited in surface variability in order to produce matching Gram matrices. In contrast, the continuous surface change of the sphere results in more surface features from which the style reconstruction optimization can choose. This increased freedom might also lead to the increased amount of noise or irregularity visible in the sphere reconstructions in Figure 11.

As we have shown in our previous paper [9], the content reconstruction works as expected similar to Gatys et al. [1].

We perform Neural Style Transfer with the cube, Figure 7 (A), as content target and the sphere, Figure 7 (H), as style target, as well as vice versa. In general, we can state that the algorithm produces reasonable shapes similar to our results with the 2D binary images, Figure 6. The cubes get rounded edges and corners, and the spheres show features typical for cubes like flat surfaces. The standardized Gram matrix approach shows more distinct features as it was also the case for the binary images. We again notice the increased amount of rough surfaces and noise with the sphere as style target compared to the style reconstruction results. Hence, we assume the causes also apply in this context. Currently, the voxel results are not as crisp compared to he 2D binary image results shown in Figure 6. Due to our experience regarding the sensitivity of Neural Style Transfer to the underlying classification network, we assume further improvements can be achieved with proper tuning of the network regarding layer numbers and sizes. Additionally, we rely on a rather small ModelNet40 dataset compared to the over one million images used for VGG19's training which also influences the quality of the learned convolutional filters and thus directly influences the later Neural Style Transfer. Despite these limitations, the results presented in this paper indicate that Neural Style Transfer can be realized for binary 3D voxel data with potentially arbitrary complex shapes.

V. Conclusion

In this paper we investigated the potential application of Neural Style Transfer for 3D voxel data. Because of the 2D pixel and 3D voxel analogy, we have shown in a 2D setup with monochromatic images and simple geometric primitives that the style loss function for ST can be applied on data lacking color, texture and smooth image gradients. Furthermore, we have introduced a new approach proposing a standardization of the Gram matrix in the style loss function which has yielded better results for the style reconstruction and style transfer tasks. Confirmed by our 2D tests, we have applied the same processes on 3D voxel data. As extension to most research on voxel classification, we have increased our model resolution to 64 voxels per dimension and trained classification networks competitive to the state of the art. We have found that the underlying network architecture is crucial for a successful Style Transfer setup and give some best practices for layer types and their setup, number of layers and size of filter banks. Again, the standardized Gram matrix based style loss yields subjectively better results compared to the original style loss definition. The models show better visual features and suffer from less noise.

Due to the small voxel resolution, we are currently limited to geometric primitives with very simple features. In order to apply Neural Style Transfer on more sophisticated voxel models, an increase of the resolution is inevitable which will be a focus of the follow up research.

Acknowledgment

The authors thank Heiko Wersing of Honda Research Institute Europe for the fruitful discussions and inspirations throughout this project.

References

[1] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image Style Transfer Using Convolutional Neural Networks," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, JUN 2016, pp. 2414–2423.

[2] Y. Jing, Y. Yang, Z. Feng, J. Ye, and M. Song, "Neural style transfer: A review," *CoRR*, vol. abs/1705.04058, 2017.

[3] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M. Yang, "Universal style transfer via feature transforms," *CoRR*, vol. abs/1705.08086, 2017.

[4] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9906 LNCS, 2016, pp. 694–711.

[5] D. Holden, I. Habibie, I. Kusajima, and T. Komura, "Fast Neural Style Transfer for Motion Data," *IEEE Computer Graphics and Applications*, vol. 37, no. 4, pp. 42–49, 2017.

[6] H. Kato, Y. Ushiku, and T. Harada, "Neural 3D Mesh Renderer," *openaccess.thecvf.com*.

[7] D. Sýkora, O. Jamriška, J. Lu, and E. Shechtman, "StyleBlit: Fast Example-Based Stylization with Local Guidance," vol. 38, no. 2, 2018.

[8] C. Ma, H. Huang, A. Sheffer, E. Kalogerakis, and R. Wang, "Analogy-driven 3d style transfer," in *Computer Graphics Forum*, vol. 33, no. 2. Wiley Online Library, 2014, pp. 175–184.

[9] T. Friedrich, N. Aulig, and S. Menzel, *EngOpt 2018 Proceedings of the 6th International Conference on Engineering Optimization*. Springer International Publishing, 2018.

[10] "Understanding Formal Analysis." [Online]. Available: http://www.getty.edu/education/teachers/building_lessons/formal_analysis.htm

[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[12] F. Uhde and S. Mostaghim, "Towards a general framework for artistic style transfer," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10783 LNCS, 2018, pp. 177–193.

[13] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D ShapeNets: A deep representation for volumetric shapes," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June-2015, pp. 1912–1920, 2015.

[14] D. Maturana and S. Scherer, "VoxNet: A 3D Convolutional Neural Network for real-time object recognition," in *IEEE International Conference on Intelligent Robots and Systems*, vol. 2015-Decem, 2015, pp. 922–928.

[15] C. R. Qi, H. Su, M. Niebner, A. Dai, M. Yan, and L. J. Guibas, "Volumetric and multi-view CNNs for object classification on 3D data," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-December, pp. 5648–5656, 2016.

[16] A. Brock, T. Lim, J. M. Ritchie, and N. Weston, "Generative and discriminative voxel modeling with convolutional neural networks," *CoRR*, vol. abs/1608.04236, 2016.