TextureAda: Deep 3D Texture Transfer for Ideation in Product Design Conceptualization

Rgee Wharlo Gallega[1(✉)], Arnulfo Azcarraga[2], and Yasuyuki Sumi[1]

[1] Future University Hakodate, Hakodate 041-8655, Hokkaido, Japan
r-gallega@sumilab.org

[2] De La Salle University, 922 Manila, Metro Manila, Philippines

Abstract. In the product design life-cycle, the conceptual design stage is an important and time-consuming phase where designers ideate in the form of sketches and 3D renderings of a product. Specifically, with 3D renderings, the choice of material texture and color is an important aspect that is often critiqued by designers because it impacts the product's visual aesthetic and the impression it evokes in the customer when first viewing the product; thus, making material selection and the conceptual design stage, overall, challenging. In this study, we turn to deep texture synthesis for generating material textures and propose a novel method, TextureAda. TextureAda creates high-fidelity textures by performing adaptive instance normalization between multiple layers of a texture generator and a pre-trained image encoder. Our experiments show that our method beats previous methods in texture synthesis visually and quantitatively. Lastly, we show how TextureAda can be applied for ideation in product design conceptualization by material texturing 3D models of furniture.

Keywords: Texture synthesis · Texture transfer · Product design

1 Introduction

Product design deals with developing physical items like furniture and electronic appliances to address specific customer needs and improve quality of life. In order to conceptualize, create, and sell a product, product designers undergo the following stages [26]: research, brief specification, conceptual design, design development, detailed design, and production, as in Fig. 1. Notably, in the conceptual design stage, designers ideate product designs in the form of 3D renderings where they are critiqued based on many aspects, especially in material choice. The material's texture and color are driving factors of the product's personality which comprises its aesthetics, associations with certain concepts, and perceptions evoked by the customer [1]. Choosing the right colors and material textures is critical in product design, and exploring alternatives can be time-consuming for designers. As a result, design critique sessions can iterate for days

H. Degen and S. Ntoa (Eds.): HCII 2023, LNAI 14050, pp. 493–505, 2023.
https://doi.org/10.1007/978-3-031-35891-3_30

or even weeks until the product's 3D rendering reaches a desirable look, making design conceptualization a challenging process.

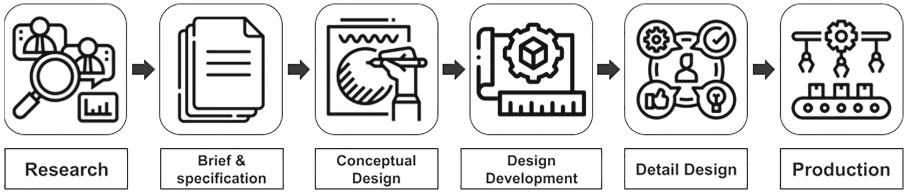

Fig. 1. The product design life-cycle and its stages from Rodgers & Milton [26].

In generative artificial intelligence, texture synthesis has been a longstanding task of creating image textures from real-life exemplars using deep generative models. Previous works train decoders that generate a single texture [12,13,28,35,38] or a wide variety of textures [2,22,23,31,32] in a single feed-forward pass. Other studies take a step further in applying generated textures onto 3D objects by inputting exemplar object images [14,27,40,43] or textual descriptions [16,17,25].

In this study, we propose to leverage deep texture synthesis and traditional 3D rendering for material texturing parts of 3D models during product design conceptualization. Building on previous deep texture synthesis methods, we introduce TextureAda, a generative model that performs adaptive instance normalization [15] from multiple layers of a pre-trained VGG-19 [36] feature extractor onto multiple layers of a modified Texture Net [38] to synthesize high-fidelity textures. Results from our experiment in texture synthesis, show that TextureAda beats previous methods visually and quantitatively based on Single Frechet Inception Distance (SIFID) [11,35]. Furthermore, we show its application in the material texturing of 3D furniture for product design conceptualization. Overall, the main contributions of this study are (1) TextureAda, a novel deep generative model for texture synthesis and material texturing of 3D models, and (2) results of a quantitative study where TextureAda outperforms previous methods in texture synthesis based on SIFID.

2 Generative AI for Texture Synthesis and Transfer

2.1 Texture Synthesis

Texture synthesis is the task of inferring image textures or patterns based on real-world examples. Pioneering studies proposed non-parametric methods in texture generation by predicting neighboring pixels [9,41] or texture patches [8,21] from a given sample of exemplar pixels. With the advent of deep learning, many studies have used neural networks to synthesize textures. Based on neural style transfer, the early work of Gatys et al. [10] optimizes a noise image on exemplar texture image features from a pre-trained VGG-19 [36] that are represented as Gram matrices. Subsequent works train decoders on texture images for rapid generation [12,13,28,35,38]. Rather than optimizing an image, Ulyanov et al. [38] propose Texture Net, a convolutional generator trained on an exemplar texture in order to generate texture variants in a single feed-forward pass. Similarly, Shaham et al. [35] utilize generative adversarial networks (GANs) to create images including textures by learning on a single example. Recently, the works of Houdard et al. [12,13] use optimal transport from local exemplar texture patches for detailed texture synthesis, and demonstrate applying this technique on both optimizing a noise image and training a generative model. On the other hand, Mordvintsev et al. [28] rely on neural cellular automata for high-fidelity texture generation.

Other studies have proposed deep generative models to synthesize multiple types of textures [2,22,23,31,32]. For instance, PSGAN [2] is an extension of GAN that can generate different types of image textures, and interpolate between samples. Lin et al. [23] uses a similar approach by building on top of StyleGAN-2 and integrating a texton broadcasting module for a more accurate and broader synthesis of textures. Lately, several works have introduced text-to-image models [31–33] that are trained on large image datasets [34], allowing near-universal texture synthesis by simply inputting text. With textures synthesis having advanced over the past several years, this study investigates how these methods can be integrated into material texture synthesis for product design conceptualization.

2.2 3D Texture Transfer

There has also been a line of research that deals with synthesizing and applying textures to 3D shapes by inputting images [14,27,40], 3D shapes [43], or text [6,16,17,25]. Initial studies [3,40] develop pipelines to extract texture patches from input images of objects and map them onto untextured 3D models. Using neural networks, other studies propose differentiable renderers [5,19,24] that have demonstrated 3D texture transfer from a single image of an object while also reshaping the 3D model accordingly. Similarly, 3DStyleNet [43] trains two neural networks to transfer textures from a source 3D model to a target model, while also reshaping the target. With a focus on furniture, Hu et al. [14] devise a pipeline of neural networks that are trained to semantically transfer material textures from an image onto corresponding parts of a 3D model. Studies that focus on interior scenes enable transferring materials from an interior image to a 3D room [42] and between 3D interior scenes [29].

With the introduction of CLIP [30], there has also been an emergence of methods for texturing 3D objects using textual descriptions. ClipMatrix [16] textures an SMPL 3D model according to a text prompt by using differentiable rendering and minimizing a loss between the embeddings of the 3D rendering image and the input text. Succeeding works like Text2Mesh [25] and TANGO [6] also utilize differentiable rendering to texture 3D meshes while also adjusting their topology to match the input description. The work of Jin et al. [17]

performs semantic style transfer onto 3D indoor scenes by also using natural language. Despite these methods being able to perform language-guided texture transfer, most of them use differentiable rendering which is time-consuming and unsuitable for quick ideation during product design conceptualization.

3 TextureAda

We adopt a novel texture synthesis method called TextureAda, which uses a modified Texture Net [38] generator and a pre-trained VGG-19 [36] encoder. TextureAda also involves another technique: adaptive instance normalization (AdaIN). Adaptive instance normalization was proposed by Huang et al. [15] for style transfer in order to transfer styles from an arbitrary number of image sources in real-time. This is done by normalizing the intermediate features of an image generator according to the features of a style image that is encoded using the VGG-19. While the study of Huang et al. [15] performs AdaIN at a single layer of their generator, in this study, AdaIN is performed several times at multiple layers of Texture Net in order to create higher-fidelity image textures.

3.1 Network Architecture

TextureAda uses a Texture Net generator and VGG-19 encoder for texture synthesis, where their architectures are shown in Fig. 2.

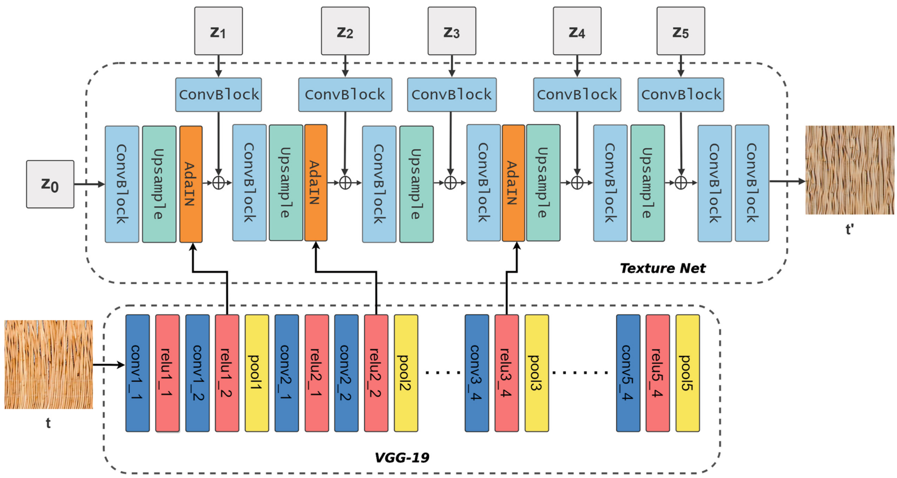

Fig. 2. Overview of TextureAda. TextureAda uses the Texture Net architecture and performs adaptive instance normalization from multiple layers of a pre-trained VGG-19.

The inputs to TextureAda are a tensor Z, which contains noise images z_0 to z_n, and a reference texture image t. t is inputted into the VGG-19 to encode its features, and the noise images of Z are sequentially inputted into Texture Net. We use 6 noise images as the generator's input and set the largest noise image of Z to 256 square pixels. AdaIN is then performed between the intermediate features of Texture Net and the extracted features from VGG-19. AdaIN is calculated using the following equation,

$$x' = \sigma(y)\left(\frac{x-\mu(x)}{\sigma(x)}\right) + \mu(y) \tag{1}$$

where x represents the intermediate features in TextureAda at a certain layer, y represents the features of texture t extracted using VGG-19 at a certain layer, and x' represents the outputted intermediate features after the operation. μ and σ calculate the mean and standard deviation, respectively. Specifically, we use the features from VGG-19 layers 'relu1_2', 'relu2_2', and 'relu3_4'. The output to TextureAda is a generated image texture t' that aims to resemble reference texture t.

In order to perform multiple AdaIN operations, the following modifications are made to the Texture Net architecture:

- All instance normalization layers are removed.
- AdaIN is performed after each upsampling block or convolutional block in Texture Net.
- For all convolutional blocks, the number of hidden features is increased from multiples of 8 to multiples of 64.

3.2 Training Details and Implementation

TextureAda is trained for 500 epochs using the Adam optimizer [20] at a fixed learning rate of $1e-4$. The covariance matrix loss [39] is used as the training loss function. In calculating the loss, we utilize the pre-trained VGG-19 to extract the features of both the real and generated image textures at layers 'relu1_2', 'relu2_2', 'relu3_4', and 'relu4_4'. The system was implemented using the PyTorch deep learning library.

4 Experiments

We evaluate TextureAda based on its performance in texture synthesis with previous methods and show how it can be applied in the material texturing of 3D models of products during design conceptualization. For texture synthesis, TextureAda was compared with the following benchmarks: Texture Nets [38], the feed-forward style transfer network [18], and the vanilla AdaIN network [15].

All texture synthesis methods were tested on two datasets: a dataset of textures of furniture and a subset of the Describable Textures Dataset [7]. The furniture textures dataset (FTD) is comprised of 35 material texture images that were segmented from images of furniture scraped from the internet. The entire Describable Textures Dataset (DTD) [7] comprises 5640 in-the-wild texture images across 47 categories. However, for this experiment, we only used categories that contain material textures such as the "braided" and "woven" categories, and removed duplicates and images that were not material textures (e.g., faces). The DTD subset that was used in this experiment contains 319 texture images across 17 categories. All images were resized to 256×256 pixels. The other benchmark methods were trained using their respective configurations.

To quantitatively compare the quality of the generated textures, the Single Image Frechet Inception Distance (SIFID) metric [11,35] is used. SIFID measures the similarity between the features of a generated image texture and real image texture that are encoded using the Inception Network [37].

Lastly, we propose applying texture synthesis methods such as TextureAda in texturing 3D models for product design conceptualization. For the scenario of choosing materials for furniture design, we transferred TextureAda's generated textures onto 3D models of chairs and tables from ShapeNet [4]. All chair and table models were manually segmented by their parts. The Blender API was used in applying the textures and rendering the 3D models.

5 Results and Discussion

For evaluating TextureAda with other benchmarks on texture synthesis, we show visual comparisons of their generated textures and also their average SIFID scores. For the furniture textures dataset, a sample of the texture images is in Fig. 3. Visually, the textures from our proposed method, TextureAda, resemble much more closely to the ground truth textures in comparison to the other methods and are of higher fidelity. For instance, in row 1 of Fig. 3, the textures of the vanilla Texture Net and AdaIN network contain discolorations while the texture of our method does not. Additionally, in row 4, the textures of both the Texture Net and AdaIN network exhibit blurs in some regions, while our method's texture does not. The textures created by the Feedforward Style Transfer network do not show any visual artifacts, our method creates much more detailed textures. It is also worth mentioning that TextureAda does not exactly copy the ground truth textures, yet is able to capture their patterns and styles.

For the DTD subset, we present a visual comparison of generated textures in Figs. 4 and 5. All methods can perform texture synthesis on non-stochastic textures. Interestingly in Fig. 5, for textures that repeat by larger image patches such as in rows 2 and 3, and the non-stochastic texture in row 4, our method is shown to learn local patterns and repeat them, whereas the other methods are not able to. However, some textures from TextureAda exhibit artifacts such as color jitters on rows 1, 2, and 4, and also lines on row 3; thus, there can be room for improvement in TextureAda.

We also present the SIFID scores for texture synthesis on both the furniture textures dataset and Describable Textures Dataset in Tables 1 and 2, respectively. Our method outperforms all previous methods based on the average SIFID of all generated textures from both datasets.

Lastly, to show the application of our method in material texturing for product design conceptualization, we semantically apply the generated textures of

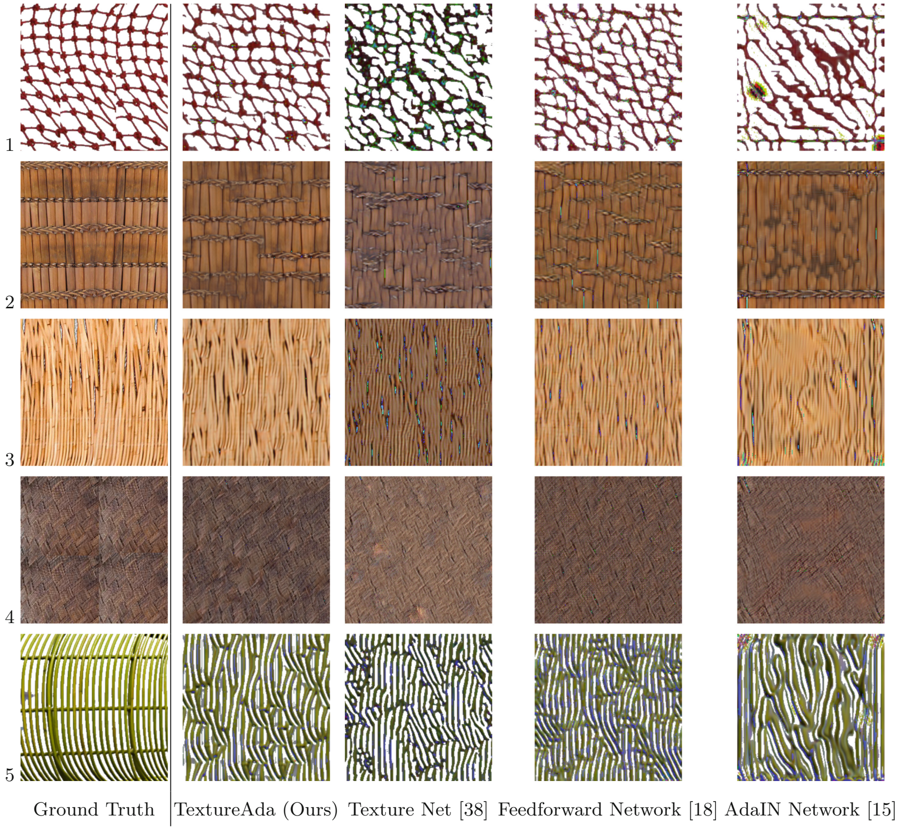

Fig. 3. Visual comparisons of generated textures from the furniture images dataset between our method and previous deep texture synthesis methods.

Table 1. Texture synthesis performance comparisons on the furniture images dataset based on average SIFID. The bolded value indicates the method with the lowest score which indicates a higher similarity to the ground truth texture.

Deep Texture Synthesis Method	Average SIFID
TextureAda (Ours)	**0.00009**
Texture Net [38]	0.16270
Feedforward Network [18]	0.00014
AdaIN Network [15]	0.20840

TextureAda from the furniture images dataset onto chairs and tables from the ShapeNet dataset. Figures 6 and 7 show combinations of similar and drastically different textures mapped onto parts of different types of chairs, respectively.

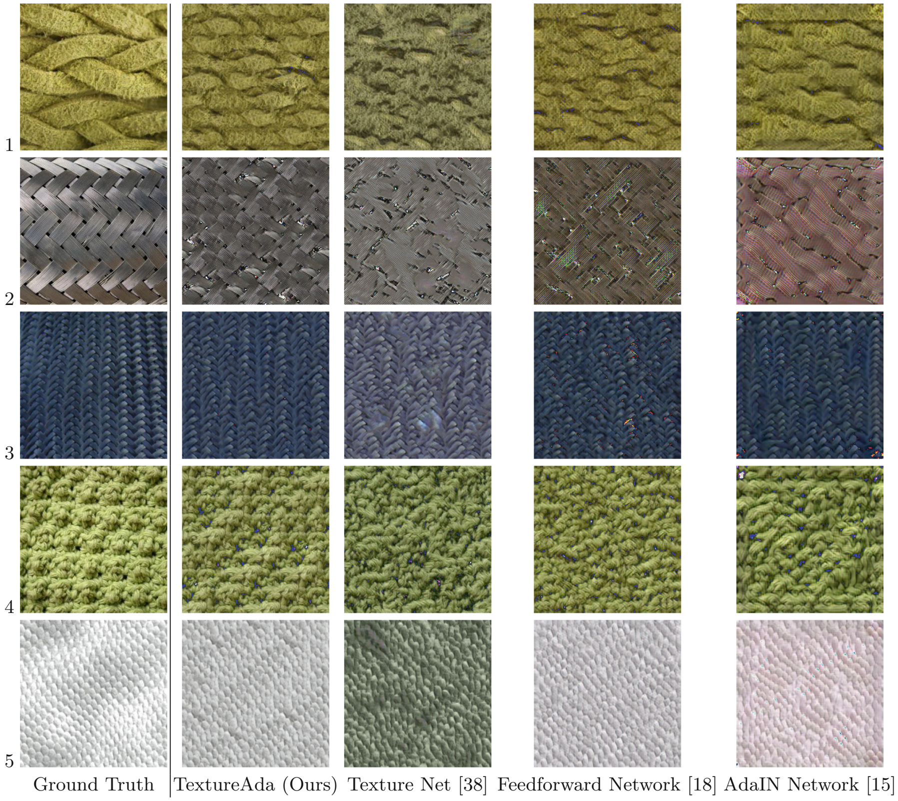

Fig. 4. Visual comparisons of generated textures from the subset of the Describable Textures Dataset between our method and previous deep texture synthesis methods. The textures shown are from the "braided" and "bumpy" categories.

Table 2. Texture synthesis performance comparisons on the subset of the Describable Textures Dataset [7] based on average SIFID. The bolded value indicates the method with the lowest score which indicates a higher similarity to the ground truth texture.

Deep Texture Synthesis Method	Average SIFID
TextureAda (Ours)	**0.000079**
Texture Net [38]	0.000105
Feedforward Network [18]	0.000145
AdaIN Network [15]	0.000192

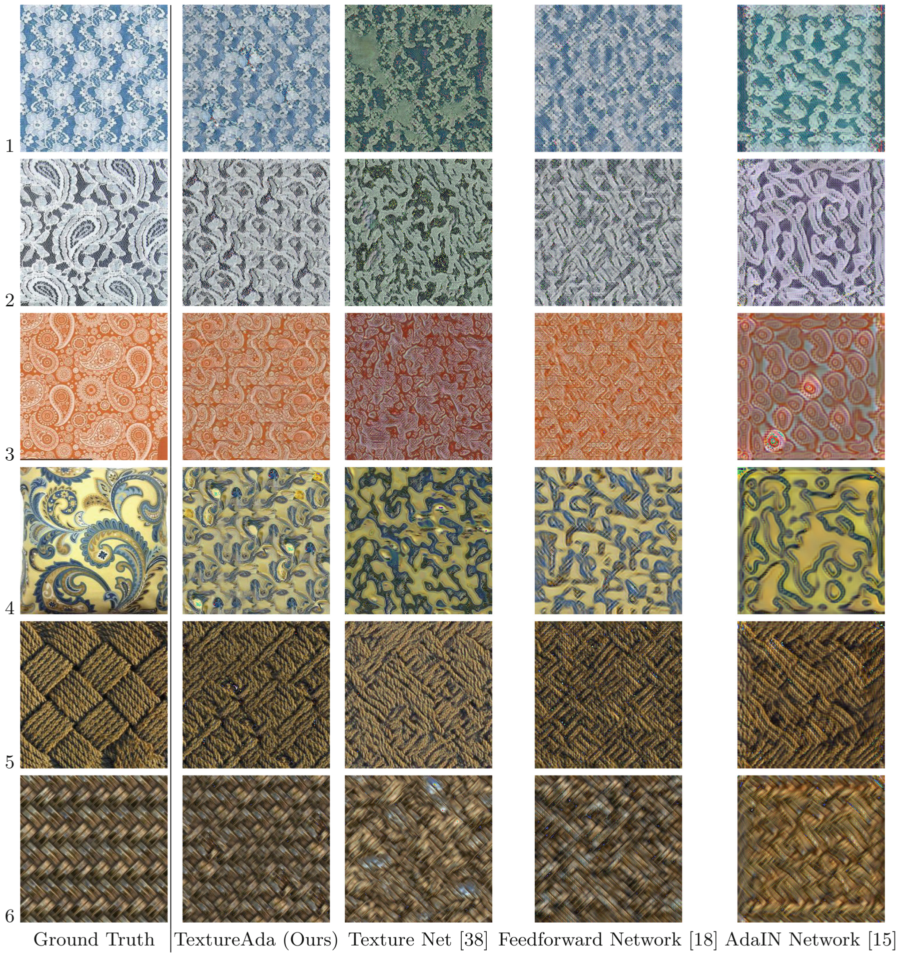

Fig. 5. Visual comparisons of generated textures from the subset of the Describable Textures Dataset between our method and previous deep texture synthesis methods. The textures shown are from the “lacelike”, “paisley”, and “woven” categories.

Fig. 6. 3D part-based texture transfer using similar-looking textures created by TextureAda.

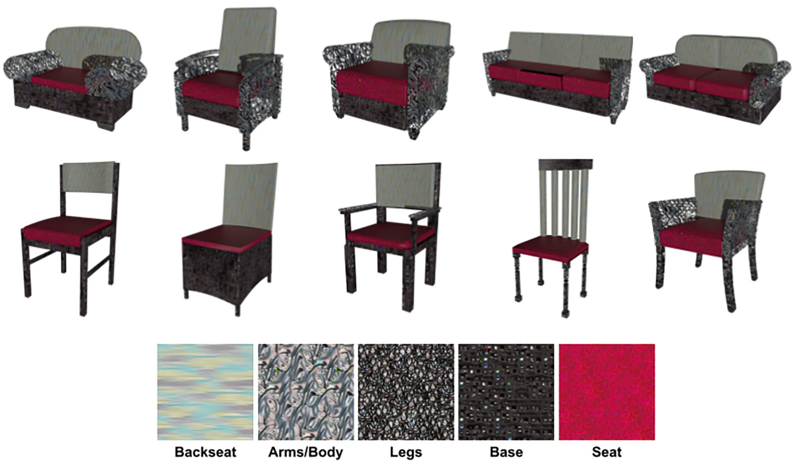

Fig. 7. 3D part-based texture transfer using different textures created by TextureAda.

6 Conclusion and Future Work

TextureAda is a novel texture synthesis method that is based on Texture Net [38] and performs adaptive instance normalization (AdaIN) [15] between multiple layers of Texture Net and a pre-trained VGG-19 [36] with the goal of creating higher-fidelity textures. Texture synthesis experiments on two datasets show that TextureAda beats previous methods visually and based on SIFID. Most importantly, we show how it can be potentially adopted for ideating in product design conceptualization by applying generated material textures to 3D models of furniture.

For future work, in order for TextureAda to be usable by designers, we plan to incorporate it into an application that allows designers to further post-process the generated textures such as changing their color and pattern sizes and also removing undesirable textures. We believe that utilizing deep texture synthesis methods like TextureAda would ease the task of quickly ideating and choosing material textures for 3D models in product design conceptualization.

References

1. Ashby, M., Johnson, K.: The art of materials selection. Mater. Today **6**(12), 24–35 (2003). https://doi.org/10.1016/S1369-7021(03)01223-9
2. Bergmann, U., Jetchev, N., Vollgraf, R.: Learning texture manifolds with the periodic spatial GAN. In: 34th International Conference on Machine Learning, ICML 2017, vol. 1, pp. 722–730 (2017)
3. Bi, S., Kalantari, N.K., Ramamoorthi, R.: Patch-based optimization for image-based texture mapping. ACM Trans. Graph. **36**(4) (2017). https://doi.org/10.1145/3072959.3073610
4. Chang, A.X., et al.: ShapeNet: an information-rich 3D model repository. arXiv preprint arXiv:1512.03012 (2015)
5. Chen, W., et al.: Learning to predict 3D objects with an interpolation-based differentiable renderer, pp. 1–12 (2019). https://nv-tlabs.github.io/DIB-R/
6. Chen, Y., Chen, R., Lei, J., Zhang, Y., Jia, K.: TANGO: text-driven photorealistic and robust 3D stylization via lighting decomposition. In: NeurIPS, pp. 1–13 (2022). http://arxiv.org/abs/2210.11277
7. Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., Vedaldi, A.: Describing textures in the wild. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 3606–3613 (2014). https://doi.org/10.1109/CVPR.2014.461
8. Efros, A.A., Freeman, W.T.: Image quilting for texture synthesis and transfer. In: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 2001 (August), pp. 341–346 (2001). https://doi.org/10.1145/383259.383296
9. Efros, A.A., Leung, T.K.: Texture synthesis by non-parametric sampling. In: Proceedings of the IEEE International Conference on Computer Vision **2**(September), 1033–1038 (1999). https://doi.org/10.1109/iccv.1999.790383
10. Gatys, L., Ecker, A.S., Bethge, M.: Texture synthesis using convolutional neural networks. Advances in Neural Information Processing Systems, vol. 28 (2015)

11. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs trained by a two time-scale update rule converge to a local nash equilibrium. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
12. Houdard, A., Leclaire, A., Papadakis, N., Rabin, J.: Wasserstein generative models for patch-based texture synthesis, pp. 269–280 (2021)
13. Houdard, A., Leclaire, A., Papadakis, N., Rabin, J.: A generative model for texture synthesis based on optimal transport between feature distributions. J. Math. Imaging Vis. (2022). https://doi.org/10.1007/s10851-022-01108-9
14. Hu, R., Su, X., Chen, X., Van Kaick, O., Huang, H.: Photo-to-shape material transfer for diverse structures. ACM Trans. Graph. **41**(4) (2022). https://doi.org/10.1145/3528223.3530088
15. Huang, X., Belongie, S.J.: Arbitrary style transfer in real-time with adaptive instance normalization. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 1510–1519 (2017)
16. Jetchev, N.: ClipMatrix: Text-controlled creation of 3D textured meshes (2021). http://arxiv.org/abs/2109.12922
17. Jin, B., Tian, B., Zhao, H., Zhou, G.: Language-guided semantic style transfer of 3D indoor scenes, pp. 11–17 (2022). https://doi.org/10.1145/3552482.3556555
18. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution, pp. 694–711 (2016)
19. Kato, H., Ushiku, Y., Harada, T.: Neural 3D mesh renderer, pp. 3907–3916 (2018). https://doi.org/10.1109/CVPR.2018.00411
20. Kingma, D.P., Ba, J.L.: Adam: a method for stochastic optimization. In: 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, pp. 1–15 (2015)
21. Kwatra, V., Schödl, A., Essa, I., Turk, G., Bobick, A.: Graphcut textures: image and video synthesis using graph cuts. ACM Trans. Graph. **22**(3), 277–286 (2003). https://doi.org/10.1145/882262.882264
22. Li, Y., Fang, C., Yang, J., Wang, Z., Lu, X., Yang, M.H.: Diversified texture synthesis with feed-forward networks (2017)
23. Lin, J., Sharma, G., Pappas, T.N.: Towards universal texture synthesis by combining Texton broadcasting with noise injection in StyleGAN-2 (2022). http://arxiv.org/abs/2203.04221
24. Liu, S., Chen, W., Li, T., Li, H.: Soft rasterizer: a differentiable renderer for image-based 3D reasoning 2019-Octob, 7707–7716 (2019). https://doi.org/10.1109/ICCV.2019.00780
25. Michel, O., Bar-On, R., Liu, R., Benaim, S., Hanocka, R.: Text2Mesh: text-driven neural stylization for meshes, 13492–13502 (2022). https://arxiv.org/abs/2112.03221
26. Milton, A., Rodgers, P.: Product design. Laurence King Publishing (2011)
27. Mir, A., Alldieck, T., Pons-Moll, G.: Learning to transfer texture from clothing images to 3D humans. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7021–7032 (2020). https://doi.org/10.1109/CVPR42600.2020.00705
28. Mordvintsev, A., Niklasson, E., Randazzo, E.: Texture generation with neural cellular automata (2021). http://arxiv.org/abs/2105.07299
29. Perroni-Scharf, M., Sunkavalli, K., Eisenmann, J., Hold-Geoffroy, Y.: Material swapping for 3D scenes using a learnt material similarity measure. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pp. 2033–2042 (2022). https://doi.org/10.1109/CVPRW56347.2022.00221

30. Radford, A., et al.: Learning transferable visual models from natural language supervision (2021). http://arxiv.org/abs/2103.00020
31. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. ArXiv abs/2204.06125 (2022)
32. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models, 10674–10685 (2022). https://doi.org/10.1109/cvpr52688.2022.01042
33. Saharia, C., et al.: Photorealistic text-to-image diffusion models with deep language understanding (2022). http://arxiv.org/abs/2205.11487
34. Schuhmann, C., et al.: LAION-400M: open dataset of CLIP-Filtered 400 million image-text Pairs, 1–5 (2021). http://arxiv.org/abs/2111.02114
35. Shaham, T.R., Dekel, T., Michaeli, T.: SinGAN: learning a generative model from a single natural image 2019-Octob, 4569–4579 (2019). https://doi.org/10.1109/ICCV.2019.00467
36. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR abs/1409.1556 (2015)
37. Szegedy, C., et al.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)
38. Ulyanov, D., Lebedev, V., Vedaldi, A., Lempitsky, V.S.: Texture networks: feed-forward synthesis of textures and stylized images (2016)
39. Virtusio, J.J., Tan, D.S., Cheng, W.H., Tanveer, M., Hua, K.L.: Enabling artistic control over pattern density and stroke strength. In: IEEE Transactions on Multimedia (2020)
40. Wang, T.Y., Su, H., Huang, Q., Huang, J., Guibas, L., Mitra, N.J.: Unsupervised texture transfer from images to model collections. ACM Trans. Graph. **35**(6) (2016). https://doi.org/10.1145/2980179.2982404
41. Wei, L.Y., Levoy, M.: Fast texture synthesis using tree-structured vector quantization. In: Proceedings of the ACM SIGGRAPH Conference on Computer Graphics, pp. 479–488 (2000). https://doi.org/10.1145/344779.345009
42. Yeh, Y.Y., et al.: PhotoScene: photorealistic material and lighting transfer for indoor scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 18541–18550 (2022). https://doi.org/10.1109/cvpr52688.2022.01801
43. Yin, K., Gao, J., Shugrina, M., Khamis, S., Fidler, S.: 3DStyleNet: creating 3D shapes with geometric and texture style variations. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 12436–12445 (2021). https://doi.org/10.1109/ICCV48922.2021.01223