IADIS International Journal on Computer Science and Information Systems
Vol. 18, No. 2, pp. 161-177
ISSN: 1646-3692

# SP3F-GAN: GENERATING SEAMLESS TEXTURE MAPS FOR FASHION

Honghong He[1,2], Zhengwentai Sun[1], Jintu Fan[1,2] and Pik Yin Mok[1,2]
[1]*School of Fashion and Textiles, The Hong Kong Polytechnic University, Hong Kong*
[2]*Labortary for Artificial Intelligence in Design, Hong Kong Science Park, Hong Kong*

**ABSTRACT**

Creating seamless textures is crucial to attain realistic 3D virtual objects and environments. This is because when stochastic textures are tiled in a straightforward manner, they produce cluttered visuals with noticeable seams that lack authenticity. This paper proposes GAN-based method for automatic seamless texture synthesis by designing three main components in generator block: (i) the texture style encoder, (ii) residual tiling blocks and (iii) the BRDFs tileable decoder in the generator of an adversarial expansion network, resulting in a continuous texture output at the seam intersection area. In addition, considering the different properties of virtual environment rendering materials, the spatially varying BRDFs (albedo, normal, roughness and displacement map) are all designed in the proposed model to handle multi-layer texture representation. Furthermore, the proposed method for generating seamless multi-layer texture maps that incorporates different loss functions, allowing us to control both the geometric shape and visual fidelity of the synthesized textures. Qualitative and quantitative experiments on the describable textures dataset (DTD) show that the generated texture maps are not only seamlessly tiled but also exhibit superior visual quality in preserving details compared to previous deep texture synthesis methods.

**KEYWORDS**

Generative Adversarial Network (GAN), Seamless texture synthesis, BRDFs, Image generation, Fashion

## 1. INTRODUCTION

Seamless textures are an essential component of many digital art and 3D applications, as they allow artists and designers to create realistic or stylized surfaces with ease. These textures are designed to be applied multiple times without any noticeable repetition or breaks in the pattern, resulting in a continuous and visually appealing surface. However, the process of creating seamless textures presents a formidable challenge, demanding meticulous planning and attention to detail in order to achieve flawless alignment of texture patterns' edge when they are tiled. Due to this complexity, the creation of such textures is usually done by skilled artists using specialized software to capture (Guo et al., 2020; Li et al., 2018) and reconstruct the details of real-world materials (e.g., leather, human skin), or they create their own patterns from scratch through various sources of inspiration and creative processes (Lu et al., 2017). Recent advances in convolutional neural networks (CNNs) and generative adversarial networks (GANs) have been applied to texture synthesis problems showing unprecedented levels of realism and quality (Gatys et al., 2015; Hertz et al., 2020; Liu et al., 2020; Mardani et al., 2020). However, the output of these methods is not always seamless when it is tiled. This means that while these methods can generate visually appealing textures, they may not be suitable for use in certain applications where seamless tiling is necessary. While some recent techniques have attempted to address the problem of tileable texture synthesis (Bergmann et al., 2017; Frühstück et al., 2019), they either require a certain degree of regularity or produce textures that lack visual fidelity. Furthermore, many existing methods for synthesizing textures focus on generating a single RGB image. Accurately simulating the appearance of real-world materials in virtual environments requires a spatially varying BRDF (Montes & Ureña, 2012) to represent complex materials that exhibit different properties at different locations on the surface. The BRDFs typically consists of multiple images, such as albedo, normal, roughness and displacement map etc., which are used collectively to accurately represent the properties of the material. Therefore, a single RGB image is usually insufficient to capture the complexity of the material's appearance. Figure 1 illustrates the differentiation between various BRDFs of original textures and seamless textures.

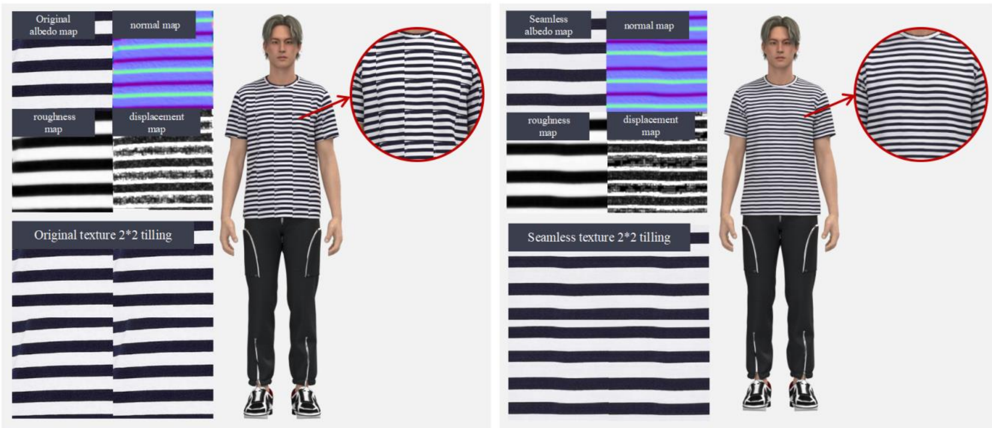


Figure 1. Simple tiling the original texture leads to visible disruptions where the seams intersect. The proposed SP3F-GAN generates seamless texture maps from any arbitrary content input

In this paper, we take these 4 types of texture maps into account (albedo, normal, roughness, and displacement map) and propose a **GAN-based method** for synthesizing **Seamless Patterns for 3D Fashion** (**SP3F-GAN**) from any arbitrary content input. We train a generative network using an adversarial extension model, which differs from the model (Zhou et al., 2018) in that SP3F-GAN involves three main components in generator block: (i) the texture style encoder $E_s$, (ii) residual tiling blocks $E_t$ and (iii) the BRDFs tilable decoder $D_t$. First the texture style representation is obtained from the $E_s$, which is then operated as a $2 \times 2$ tiled style representation in the latent space. The tiled feature representation is then decoded to produce a seamless output image via $D_t$. The discriminator is trained to classify whether a $2k \times 2k$ texture block is real (a crop from the input exemplar) or fake (synthesized by generator). In addition, in order to allow patterns to be reasonably rendered in 3D scenes, we designed four decoders to cope with the generation of different texture maps. The proposed method is shown to outperform state-of-the-art solutions for seamless texture synthesis. Diverging from prior works on deep texture synthesis, the SP3F-GAN exhibits the capacity to generate high-quality texture patterns that that faithfully preserve the geometric representation of the original input. Moreover, it facilitates the seamless texture maps generation of infinite patterns that are well-suited for mapping to on a wide range of surfaces. The outcomes of the SP3F-GAN model find practical utility in real-world applications such as digital fabric creation, 3D fashion design, and related fields, as shown in Figure 2. Notably, the data of the 3D avatars in this work is from (*CLO3D*, 2023), by making alterations to facial shapes and applying makeup according to the desired preferences. Additionally, all the 3D garments showcased have been designed and created by our own. The key contributions of this work are summarized as follows:

i. An automatic deep generative architecture for synthesizing seamless multi-layer texture maps is proposed that combines different loss functions to control the synthesis of textures geometric shape and visual fidelity.
ii. This is the first of its kind of thorough investigation for extending seamless patterns for 3D fashion applications, providing texture data support for future research in the 3D garments reconstruction domain.
iii. Extensive qualitative and quantitative experiments have confirmed the exceptional capability of the proposed SP3F-GAN in generating seamless texture maps.

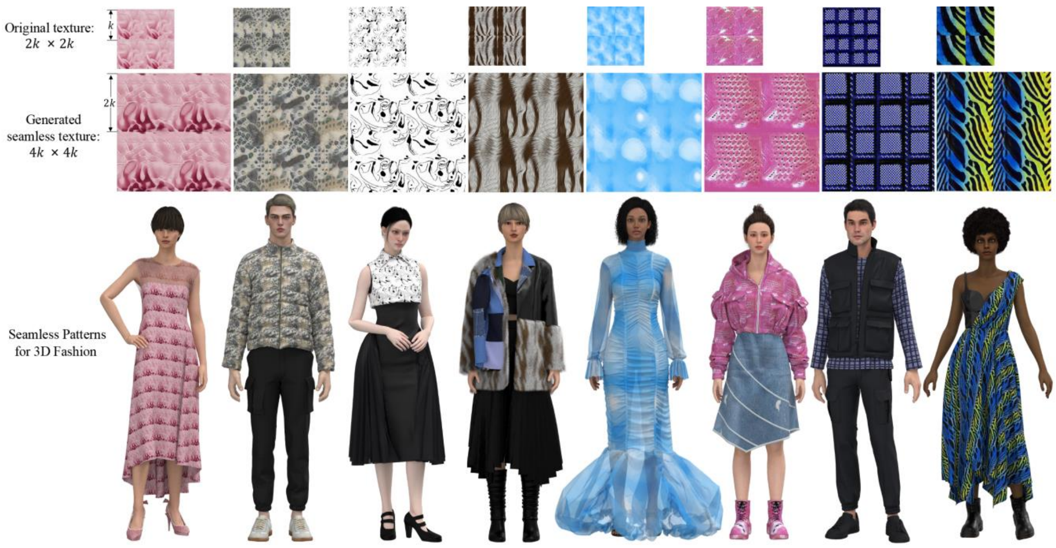


Figure 2. Tiling the original texture creates a discontinuity at the intersection of the seams, as shown in the top row. Our method can generate seamless textures, which are able to apply to fabric pattern swatches, 3D garments and etc

## 2. RELATED WORK

**Texture synthesis.** Traditional texture synthesis methods aim to generate texture that is similar to a given input sample texture. These methods can be broadly categorized into non-parametric texture synthesis and parametric texture synthesis techniques. Earlier non-parametric texture synthesis algorithms included image quilting, by stitching together small patches from an input texture to generate a larger texture image (Efros & Freeman, 2001; Efros & Leung, 1999); Graph cuts, by minimizing the energy function between generated and input images (Kwatra et al., 2003); Genetic algorithms, by searching for optimal correspondences that match well with the input images (Dong et al., 2007), and PatchMatch algorithms, by using nearest-neighbor search to find patches in the input texture that are similar to patches in the generated texture (Barnes et al., 2009; Cohen et al., 2003; Kaspar et al., 2015; Levin et al., 2004; Liang et al., 2001). Although these non-parametric texture synthesis methods achieve remarkable outcomes in producing high-quality textures with diverse features, they necessitate more manual input, such as adjusting the number and size of patches. The family of parametric texture synthesis algorithms, requiring less manual operations, belong to a category of methods that rely on statistical description of the input texture, which generate a new texture by imposing a set of statistical constraints (Galerne et al., 2010; Portilla & Simoncelli, 2000; Rabin et al., 2012). These algorithms, prioritize faithfully reproducing input or sample textures, which lack control over the output. And while these methods can successfully generate a variety of textures, they may struggle with highly complex or abstract textures that lack clear statistical regularities. Textures are less suited to methods based on local patch-based sampling or statistical modeling, which may result in outputs that lack spatial coherence, appear distorted, or fail to capture the desired complexity (e.g., hair, fur, and natural landscapes) (Akl et al., 2018).

Most recently, deep network models have gained popularity due to their superior capability in generating diverse and realistic content across various domains (Karras et al., 2017; Karras et al., 2019; Karras et al., 2020) (Lin et al., 2023). Their effectiveness and versatility have contributed to their widespread adoption and recognition as a powerful approach for various generation tasks. This approach learns the texture feature distribution from input images based on networks such as CNN (Krizhevsky et al., 2012), VAE (Kingma & Welling, 2013), UNet (Ronneberger et al., 2015), GAN (Goodfellow et al., 2014), etc., which can then be used to generate new images matching learned texture representations. For example, Li and Wand (2016) proposed a deep feature extension based on patch-based texture synthesis, including combining a generative Markov random field model with a discriminatively trained deep convolutional neural network (dCNN) to control texture layout, thereby improving texture visual credibility. The approach developed by (Jetchev et al., 2016) is based on a texture synthesis model that generative adversarial networks (GANs) have learned, which extends the input noise distribution space from a single vector to a complete space tensor, enabling the creation of an architecture with multiple properties. These methods have demonstrated their effectiveness in the field of texture synthesis. Building upon this concept, the proposed method also employs a GAN model inspired by (Zhou et al., 2018) whereas we take the entire dataset as input and extends this model to perform multi-texture map synthesis, allowing the generation of multiple tileable textures simultaneously.

**Seamless texture synthesis.** While texture synthesis has been a popular research topic for a long time, the specific aspect of synthesizing tileable textures has garnered less attention until recent years. In the early stages, one approach for obtaining texture images involved scanning physical textures and applying certain image transformations (Shapiro, 1995; Sterken et al., 1992) to achieve a visually consistent appearance. It relies on manual operations, prove to be arduous, and inadequate in effectively tackling common texture challenges such as non-uniform illumination, irregularities, and the diverse nature of textures. Early seamless texture synthesis methods mainly involved copying and blending patches from input textures to create larger seamless textures. Various algorithms, such as texture quilting (Efros & Freeman, 2001) and Wang tiles (Cohen et al., 2003), have been developed to ensure smooth transitions and continuity between the patches. However, these methods may not adequately capture specific features or statistical properties of the target domain, resulting in synthesized textures that do not match the desired style or context in optimization schemes that forcibly eliminate seams.

Researchers continue to address these limitations and explore new techniques that harness the power of deep neural networks for seamless texture synthesis. Bergmann et al. (2017) propose the Periodic Spatial GAN, which can create textures from a set of random numbers that represent the expected general image, local variation, and periodic repetition of learning in generated images. GANosaic (Jetchev et al., 2017) extends such method to generate textures by optimizing the latent noise space to produce textures that match the overall content of a given guidance image. While these methods produce smoother transitions between texture patches, they still suffer from issues relating to texture scalability and variability. Rodriguez-Pardo and Garces (2022) solves the scalability problem by completing textures into tiles through tile search and feature repetition. Relatedly, Lin et al. (2023) introduced a multi-scale texture broadcast module based on SyleGAN-2 (Karras et al., 2020). This module, combined with noise injection, effectively introduces suitable sensing bias into the synthesis process, facilitating the seamless textures synthesis. These methods achieve seamless texture generation by explicitly incorporating periodic patterns into the generator. However, they train on small datasets or datasets with simple texture structures (lack of real photos or complex texture structures) so that the generated textures may not exhibit the same level of realism or complexity as desired. Our proposed method deviates from the aforementioned approaches. Instead, we introduce a novel approach that offers a different perspective on seamless texture synthesis. Inspired by (Zhou et al., 2018) in non-stationary texture synthesis task, in which we modified its network structure and extended it to BRDFs for generating tileable texture maps.

## 3. METHOD

Our ultimate goal is to generate seamless texture maps, which can then be used for downstream tasks, such as digital fabric creation, 3D fashions, etc. These seamless texture maps are perceptually similar to smaller input texture samples being tiled. However, by default, the input texture samples are simply and naively tiled with significant seams. Our main idea is to eliminate the seams by automatically learning from the network to generate tileable texture maps from arbitrary texture input examples. The GAN (Goodfellow et al., 2014) is used in our method to learn the implicit representation of textures in two neural modules, generator $G$ and discriminator $D$. Given a $k \times k$ source block $S_A$ cropped from the input sample, the generator is trained to generate a new $2k \times 2k$ output $\hat{T}_A$. Among them, the other maps of the input sample are normal map, roughness map and displacement map, respectively denoted as $S_N$, $S_R$ and $S_D$, which corresponding to the generated new texture maps represented as $\hat{T}_N$, $\hat{T}_R$ and $\hat{T}_D$. Discriminator $D$ is trained to distinguish between these four $2k \times 2k$ texture maps are real (cropped from input samples) or fake (synthesized by $G$). Synthetic seamless texture maps can be applied to 3D garments for visualization and presentation. The diagram in Figure 3 depicts our proposed method of SP3F-GAN.

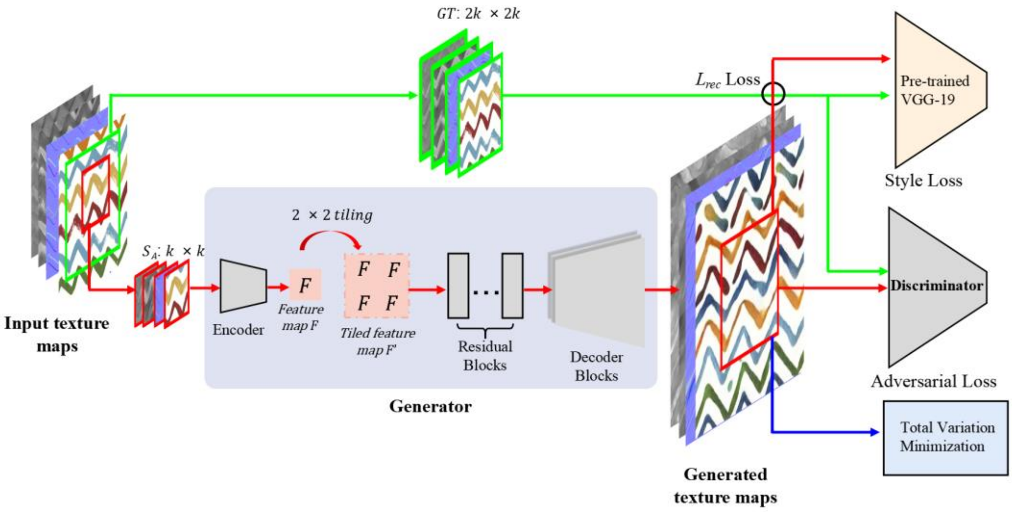


Figure 3. Illustration of the proposed method for seamless texture maps generation

**Network architecture.** The network architecture of SP3F-GAN is adopted from (Rodriguez-Pardo & Garces, 2022; Zhou et al., 2018), including a residual connected encoder-decoder, convolution generator $G$, and convolution discriminator $D$, the overall network framework is shown in Figure 4. However, their methods mainly focus on 2D textures. To achieve 3D seamless textures, we extend their methods with two major aspects. First, instead of synthesizing a single RGB image, we extend Generator's outputs to 4 texture maps (albedo, normal, roughness and displacement map). Second, to keep the continuity in the seam region, we applied Total Variation Loss as a regularization term.

**Generator.** The proposed generator consists of three main components: (i) the texture style encoder $E_s$, (ii) residual tiling blocks $E_t$ and (iii) the BRDFs tileable decoder $D_t$. First, the texture style encoder considers the input texture as an albedo map $S_A$. This is because that the input picture mainly represents the basic texture style information such as color, geometric printing. We implement this encoder with a Convolutional Neural Network (CNN) structure that down samples the $S_A$ four times. When the spatial size of $S_A$ is set to $k$, the shape of the obtained style representation $F$ could be $k/4$. To synthesis seamless textures, we approach to tile the style representation twice both horizontally and vertically:

$$F' = \begin{bmatrix} F & F \\ F & F \end{bmatrix} \quad (3\text{-}1)$$

As a result, the shape of tiled style representation $F'$ could be $k/2$ which is the half of the input albedo map $S_A$. This explicitly tiling linearly transform the textures which simulates the real scenario. However, this process may lead to artifacts during the margins. To allow the tiling seamlessly, we then apply the residual tiling blocks $E_t$ to fuse the tiled representation $F'$ into $\mathcal{F}$. During the residual blocks, it could eliminate discontinuous artifacts in the tiling margin.

Finally, a tileable decoder $D_t$ is designed to synthesis tileable albedo map, normal map, roughness and displacement map. However, synthesizing multiple texture maps with a single generative model presents additional challenges for the case of a single texture map because different texture maps produce different visual expressions. Inspired by (Rodriguez-Pardo & Garces, 2022), we extend the decoder to four decoding texture maps for each of the four different data types. Specifically, instead of implementing a single decoder that outputs multi-channel texture maps, we adopt the $D_t = \{D_t^A, D_t^N, D_t^R, D_t^D\}$ as a set of decoders with distinct learnable parameters. Each superscript of $D_t$ denotes the corresponding texture map. The network framework of the style encoder $E_s$, residual tiling blocks $E_t$ and the tileable decoder $D_t$ is shown as the style encoder $E_s$, residual tiling blocks $E_t$ and tileable decoder $D_t$ in Figure 4. The overall generation process of seamless texture maps can be expressed as:

$$\hat{T}_A, \hat{T}_N, \hat{T}_R, \hat{T}_D = \mathcal{G}(S_A) = D_t(F') = D_t(E_t(E_s(S_A))) \tag{3-2}$$

**Discriminator.** The synthesis of seamless textures takes a texture style image as reference and output tileable textures. This form of synthesis could be considered as image translation. Therefore, we adopt the PatchGAN discriminator structure from image translation tasks (Isola et al., 2017; Zhou et al., 2018; Zhu et al., 2017). Unlike the traditional implementation that only discriminate a global real/fake, the PatchGAN discriminator considers the input image as several patches. It will make discrimination on each patch so that improve the synthesis quality on local regions. However, their design only considers a discrimination on regular RGB image which is quite different from our synthesized $\hat{T}_A, \hat{T}_N, \hat{T}_R, \hat{T}_D$. Our implementation of PatchGAN discriminator takes the concatenation of synthesized images and input image.

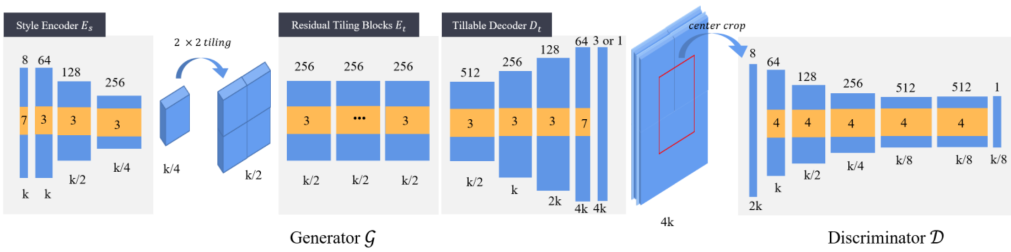


Figure 4. The illustration of architecture of the proposed generator and discriminator. The top of each blue region indicates the number of feature channels, while the spatial resolution of feature maps is indicated underneath. Furthermore, the kernel sizes are specified within the central yellow regions of the diagram

**Loss Function.** We train the network according to the standard GAN framework (Goodfellow et al., 2014), and iteratively optimize generator $\mathcal{G}$ and discriminator $\mathcal{D}$. We denote the real input $x_{\text{real}}$ and fake input $x_{\text{fake}}$ as follows:

$$x_{\text{real}} = S_A \oplus T_A \oplus T_N \oplus T_R \oplus T_D$$

$$x_{\text{fake}} = S_A \oplus \hat{T}_A \oplus \hat{T}_N \oplus \hat{T}_R \oplus \hat{T}_D \tag{3-3}$$

where $\oplus$ denotes channel-wise concatenation. The loss function for optimizing $\mathcal{D}$ is defined as:

$$\mathcal{L}_{\mathcal{D}} = -\mathbb{E}[\log \mathcal{D}(x_{\text{real}})] - \mathbb{E}[\log (1 - \mathcal{D}(x_{\text{fake}})] \tag{3-4}$$

The adversarial loss for optimizing $\mathcal{G}$ is:

$$\mathcal{L}_{adv} = -\mathbb{E}[\log \mathcal{D}(\mathcal{G}(S_A))] \tag{3-5}$$

To stabilize the training process and fasten the training time, we used three additional loss terms: reconstruction loss $\mathcal{L}_{\text{rec}}$, style loss $\mathcal{L}_{style}$ (Gatys et al., 2015), and total variation loss $\mathcal{L}_{tv}$ (Larsen Greiner et al., 2022) for optimizing $\mathcal{G}$. The $\mathcal{L}_{\text{rec}}$ is defined as:

$$\mathcal{L}_{\text{rec}} = \mathcal{L}_1^a + \mathcal{L}_1^n + \mathcal{L}_1^r + \mathcal{L}_1^d \tag{3-6}$$

where the superscript $a$, $n$, $r$, and $d$ denotes the L-1 distance between the generated and ground truth albedo, normal, roughness and displacement map respectively. To improve the perceptual quality of the synthesized textures, we then apply style loss on $\hat{T}_A$ and $\hat{T}_N$ separately:

$$\mathcal{L}_{\text{style}} = \mathcal{L}_{\text{style}}\left(\hat{T}_A, T_A\right) + \mathcal{L}_{\text{style}}\left(\hat{T}_N, T_N\right) \tag{3-7}$$

where the definition of a single style loss term is as follows:

$$\mathcal{L}_{\text{style}}\left(\hat{T}_A, T_A\right) = \frac{1}{4n^2M^2}\sum_{i,j}\left(G_{ij}^{\hat{T}_A} - G_{ij}^{T_A}\right)^2 \tag{3-8}$$

where $n$ is the channel number and $D$ is pixel number of a feature map, $G_{ij}^{\hat{T}_A l}$ denotes the gram matrix of $\hat{T}_A$ as $l$th feature map. Since the feature map are obtained by a pretrained VGG-19, it could be only applied on texture maps with three channels. Therefore, we only apply the style loss on albedo and normal maps. To apply constraints on the continuity of the synthesized textures, we then applied total variation loss on the synthesized textures:

$$\mathcal{L}_{tv} = \mathcal{L}_{tv}\left(\hat{T}_A\right) + \mathcal{L}_{tv}\left(\hat{T}_N\right) + \mathcal{L}_{tv}\left(\hat{T}_R\right) + \mathcal{L}_{tv}\left(\hat{T}_D\right) \tag{3-9}$$

For each total variation loss term, it calculates the mean average of gradients of an image as follows:

$$\mathcal{L}_{tv}(\mathrm{x}) = \sum_{i,j}\left(\left(x_{i,j+1} - x_{ij}\right)^2 + \left(x_{i+1,j} - x_{ij}\right)^2\right)^{\frac{1}{2}} \tag{3-10}$$

where x stands for the synthesized image, $x_{ij}$ is a pixel from image x at location $(i, j)$.
The overall loss for optimizing the generator $\mathcal{G}$ is:

$$\mathcal{L}_{\mathcal{G}} = \lambda_{adv}\mathcal{L}_{adv} + \lambda_{rec}\mathcal{L}_{rec} + \lambda_{style}\mathcal{L}_{style} + \lambda_{tv}\mathcal{L}_{tv} \tag{3-11}$$

where $\lambda_{adv}$, $\lambda_{adv}$, $\lambda_{adv}$, and $\lambda_{adv}$ are the weight of each loss term.

## 4. EXPERIMENT

In this section, we evaluate our 3 main contributions. First, we examine the impact of the design choices of Generator $\mathcal{G}$ on multilayer texture maps (Section 4.1), and then we discuss the impact of the loss function (Section 4.2). Finally, we compare our SP3F-GAN with the SOTA methods on seamless texture synthesis (Section 4.3).

## 4.1 Consistency of Texture Maps

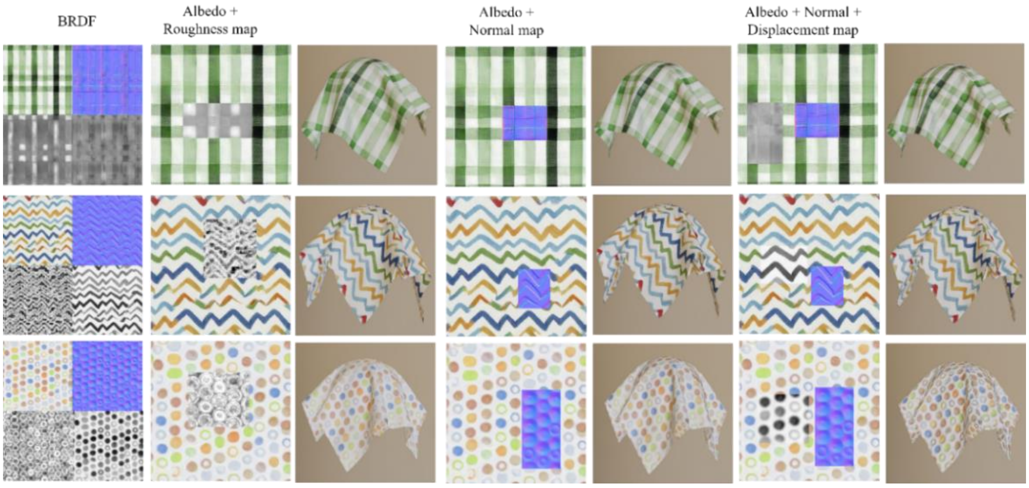


Figure 5. Illustration of consistency of synthesized texture maps

Our generator $\mathcal{G}$ can generate high-quality seamless texture maps but cannot guarantee that the pattern of these texture maps is consistently at a specific location. Gradients may not be shared in texture maps because $L_{style}$ and $L_{tv}$ are calculated independently for each map. In this section, we demonstrate the usefulness of our generator through a practical application. For the four generated seamless texture maps, we randomly select two or three maps (e.g., albedo + roughness map, albedo + normal map, and albedo + normal + displacement map) to overlay on the fabric swatches for 3D garments and observe whether the pixel-level consistency between the maps is preserved. Figure 5 shows the consistency of synthesized texture maps.

## 4.2 Ablation Study of Loss Function

The generator $\mathcal{G}$ is trained using a composite loss function composed of adversarial, total variation, perceptual, and pixel-level components, as detailed in the Loss Functions section of the proposed method. Each of these components has a distinct effect on the result. We undertake ablation study to independently explore the effects of distinct components (or their combination) on the generation of seamless texture maps, which can help us better understand their respective contributions to the synthesis process. First, we find that $\mathcal{L}_{adv}$ is critical for preserving high-level semantic consistency in synthetic outputs. It guarantees that the texture generated matches the desired appearance and general structure. Then is our newly created total variation component $\mathcal{L}_{tv}$ , which plays an important role in generating seamless texture. Although minimize $\mathcal{L}_{tv}$ alone, an 'extremely regular' texture pattern without semantic information is obtained (as shown in Figure 6 $\mathcal{L}_{tv}$). Mainly because $\mathcal{L}_{tv}$ calculates the change or difference in pixel intensity between contiguous pixels in the synthesized texture, it penalizes high variations (sudden changes or discontinuities in the texture) and encourages smooth transitions by reducing this loss. In addition, inspired by , we added $\mathcal{L}_{style}$ during training procedure. $\mathcal{L}_{style}$ effectively represents texture in synthetic output by adding finer details, increasing the realism and authenticity of the generated texture maps. The final component is $\mathcal{L}_{rec}$ that is the combination of the $\mathcal{L}_1$ norm of the BRDFs. The absence of $\mathcal{L}_{style}$ and $\mathcal{L}_{adv}$ in the loss function results in significant color distortion in the generated texture. Without the $\mathcal{L}_{rec}$ loss function, it is difficult to constrain the generated pattern on the texture shape (e.g., the specific spacing of the green and white stripes in Figure 6). Similarly, the omission of $\mathcal{L}_{tv}$ leads to a less smooth and visually coherent texture in the generated output. Consequently, the combination of these components yields the most convincing and satisfactory results, in this experiment we set $\lambda_{adv}$ = 1.0, $\mathcal{L}_{tv}$ = 0.01, $\lambda_{style}$ =1.0 and $\lambda_{rec}$ = 100. Figure 6 illustrates the study's findings.

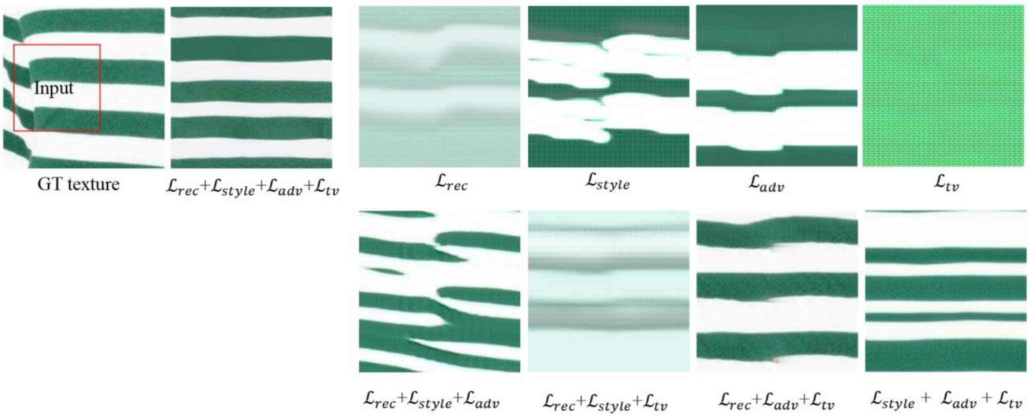


Figure 6. Illustration of an ablation study of the impact of loss functions on synthesized texture quality (this study was mainly conducted on albedo maps). The results demonstrate that the combination of all loss function yields the most optimal and superior textures

## 4.3 Comparison Experiment

### 4.3.1 Qualitative Comparison

We compare with the SOTA seamless texture synthesis method proposed Efros and Freeman (2001) 's image quilting for texture synthesis and transfer, PSGAN by Bergmann et al. (2017) and Self-organising by (Niklasson et al., 2021), using samples from the describable textures dataset (DTD) (Cimpoi et al., 2014) in Figure 7 (frontal-parallel pattern examples) and Figure 8 (irregular and non-stationary pattern examples). Efros and Freeman (2001) 's method is a well-known and widely used seamless texture synthesis algorithm that leverages local patches from a source texture and intelligently stitches them together.

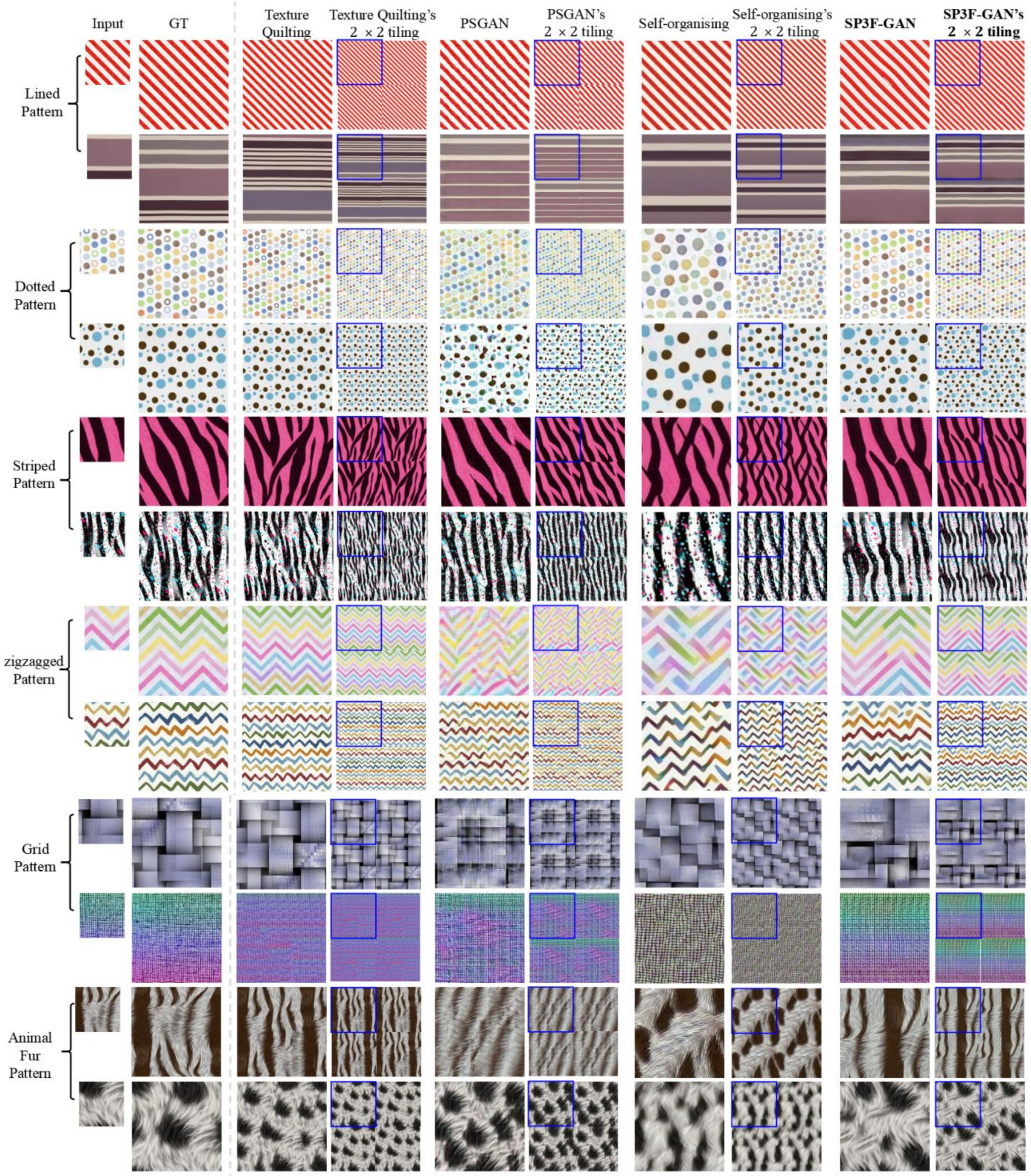


Figure 7. Qualitative comparison with SOTA method. On the left column, we show the input textures cropped from the ground truth. From left to right, the synthesized results of the texture quilting (Efros & Freeman, 2001), PSGAN (Bergmann et al., 2017), the self-organising textures (Niklasson et al., 2021),and the proposed SP3F-GAN. Outputs are tiled a similar number of times (at least twice in each dimension) for better visualization. Our method generally captures better the overall structure of the frontal-parallel pattern examples

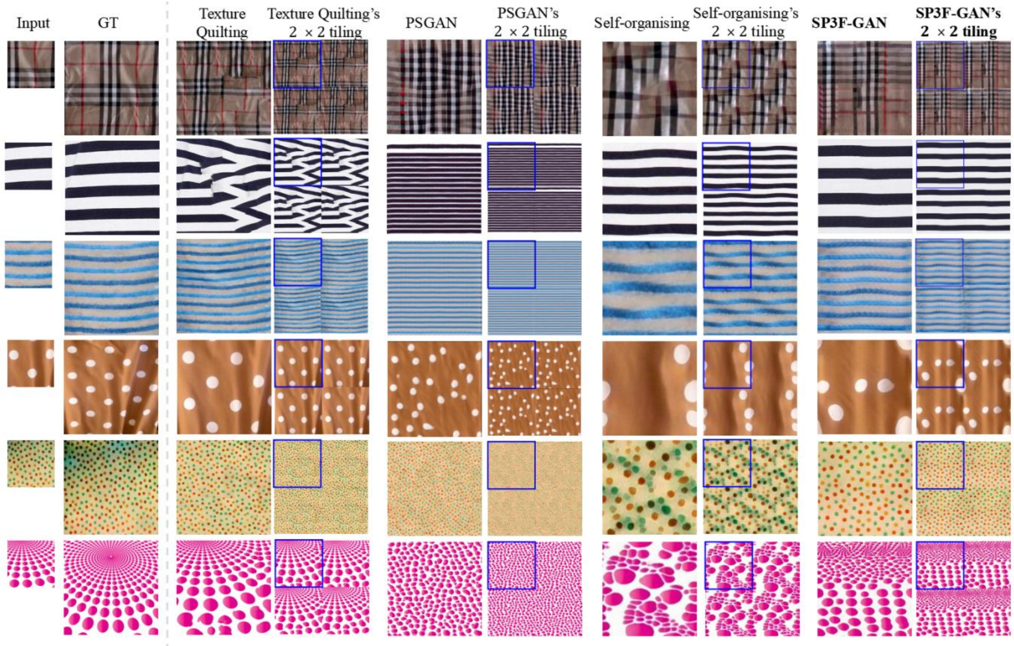


Figure 8. Qualitative comparison with SOTA method. The proposed SP3F-GAN captures better the structure of the irregular and non-stationary pattern examples, while providing seamless and semantically coherent borders, for enhancing tileability

In Figure 7, it is evident that this algorithm excels in reconstructing frontal-parallel pattern images with accuracy, particularly suited for regular textures (e.g., lined and dotted pattern). However, it falls short in generating seamless textures compared to the proposed SP3F-GAN. That is to say, Efros and Freeman's algorithm in generating seamless textures largely depends on the nature of the input pattern. The input pattern is originally a tileable texture, and the generated textures by Efros and Freeman (2001)'s method are seamless. Otherwise, poor results will be obtained. This problem is especially noticeable for stochastic textures (e.g., striped and animal fur pattern) and gradient color textures (e.g., grid and smeared pattern). In addition, in the case of Figure 8, for highly irregular textures (fabric patterns with wrinkles) and non-stationary pattern textures (polka-dot pattern), the method proposed by Efros and Freeman (2001) fails to generate satisfactory textures. This failure can be attributed to the inability to find a 'perfect patch' that seamlessly integrates into the overall texture synthesis process. The method also struggles with handling perspective distortion at the local level, let alone achieving global seamlessness in the synthesized textures. The Periodic Space GAN (PSGAN) proposed by Bergmann et al. (2017) PSGAN performs superiorly in finding the regularity and periodicity of the texture's internal patterns. For example, in the line and dot pattern examples in Figure 7 and Figure 8, it works even for non-stationary pattern textures (rosy polka-dot pattern). Textures generated by PSGAN exhibit a deficiency in capturing fine-grained details or intricate patterns found in real-world textures. This limitation becomes apparent when observing the presence of zigzagged and grid patterns in Figure 7 and Figure 8 (clear artificial artifacts). Although PSGAN can accurately discover the periodic pattern representation (patterns can be seamless locally), it fails when the required changes are global (meaning that the effect is not good for 2×2 tiling).

Textures can be applied to fabric pattern samples or 3D garments where seamless tiling is a minimum expectation. Another comparable method is the self-organising textures proposed by (Niklasson et al., 2021) using neural cellular automata. Although they generated textures are global tileable, this method cannot well preserve the local geometric structure of the texture pattern, which limits the application of the generated seamless texture on its downstream tasks. In addition, some textures with complex patterns, such as the generated gradient grid pattern texture, have obvious color differences with ground truth (GT) textures. Comparing the above methods, the proposed SP3F-GAN can not only generate tileable patterns (global level) but also preserve the high-level semantic structure (local level), thereby yielding clear and realistic synthetic textures and maximizing the similarity of the geometric style between the input texture and the generated seamless textures (see examples in Figure 7 and Figure 8). Particularly, SP3F-GAN provides high-quality output for textures with non-uniform illumination (e.g., chequered pattern), irregular textures (e.g., striped, bubbly pattern and fabric with wrinkles), gradient color textures (e.g., grid and smeared pattern) and non-stationary pattern).

### 4.3.2 Quantitative Comparison

Table 1. Quantitative comparison between texture quilting method, PSGAN, the self-organising textures and the proposed SP3F-GAN. We offer the average results of various perceptual metrics across a diverse range of textures. Higher is better for SSIM, while lower is better for LPIPS

| | ↑SSIM | ↓LPIPS |
|---|---|---|
| Texture quilting (Efros & Freeman, 2001) | 0.2163 | 0.4920 |
| Center cropped Texture quilting’s 2*2 tiling | 0.1897 | 0.6162 |
| PSGAN (Bergmann et al., 2017) | 0.1763 | 0.5957 |
| Self-organising (Niklasson et al., 2021) | 0.1753 | 0.5328 |
| Center cropped Self-organising’s 2*2 tiling | 0.2326 | 0.4531 |
| Center cropped PSGAN’s 2*2 tiling | 0.1896 | 0.5236 |
| SP3F-GAN | 0.2354 | **0.4327** |
| Center cropped SP3F-GAN’s 2*2 tiling | **0.2457** | 0.5063 |

During the quantitative evaluation, we used common metrics such as the Structural Similarity Index Metric (SSIM) (Wang et al., 2004) and Learning Perceptual Image Block Similarity (LPIPS) (Zhang et al., 2018) to compare the performance of the synthesized textures and the ground truth (GT) textures. The SSIM metric was developed by Wang et al. (2004) and assesses structural similarity in the pixel space, making it appropriate for evaluating synthesized textures. Additionally, we used LPIPS, a perceptual distance metric in deep image space that was developed by Zhang et al. (2018). In earlier research, this metric has been extensively employed to evaluate generative models (Huang et al., 2018; Karras et al., 2020; Zhan et al., 2022). To ensure fair comparisons, we conducted experiments using the Describable Textures Dataset (DTD) and compared our method with traditional SOAT texture quilting method (Efros & Freeman, 2001), the deep learning-based method PSGAN (Bergmann et al., 2017) and self-organising textures (Niklasson et al., 2021). we compared the composite image generated by our method with the corresponding GT texture, and our proposed approach achieved superior SSIM and LPIPS scores. It is noted that the input parameters of the texture quilting algorithm (e.g., block size, number of blocks, etc.) are according to the default settings in the paper. The network parameters and sizes of PSGAN and self-organizing textures are exactly the same as those defined in the paper. Additionally, we compared the 2×2 tiled generated textures with the GT textures. However, since the metrics used in this experiment require both images to have the same resolution, we cropped the 2×2 tiled generated image to match the resolution of the corresponding GT texture. Interestingly, our method outperformed texture quilting method, PSGAN and self-organising textures method in terms of both SSIM and LPIPS scores, as shown in Table 1.This indicates that our approach effectively preserves the stylistic and semantic content of the generated textures by leveraging various loss functions. Furthermore, our method's spatial manipulation algorithm enables seamless boundaries between tiles, surpassing texture quilting method, PSGAN and self-organising textures method in terms of structural similarity. This indicates that the proposed SP3F-GAN achieves superior results in terms of maintaining the structural coherence and integrity of the synthesized textures.

## 4.4 Limitations and Discussion

During the experimental analysis discussion, we found that the proposed SP3F-GAN also has some limitations. For example, our method cannot handle one texture with several different geometric styles (includes multiple semantic objects) simultaneously, as shown in Figure 9 row i. Additionally, in cases where the input texture lacks sufficient regularity, the adversarial generation process of SP3F-GAN may not yield satisfactory results. This can lead to the disruption and reconstruction of the local smeared pattern of the face, as seen in Figure 9 row ii. Furthermore, when dealing with non-stationary images (row iii) or highly complex patterns with intricate color and geometric shapes, such as the paisley pattern in Figure 9 row iv, SP3F-GAN can generate seamless textures but at the cost of fidelity in the generated texture.

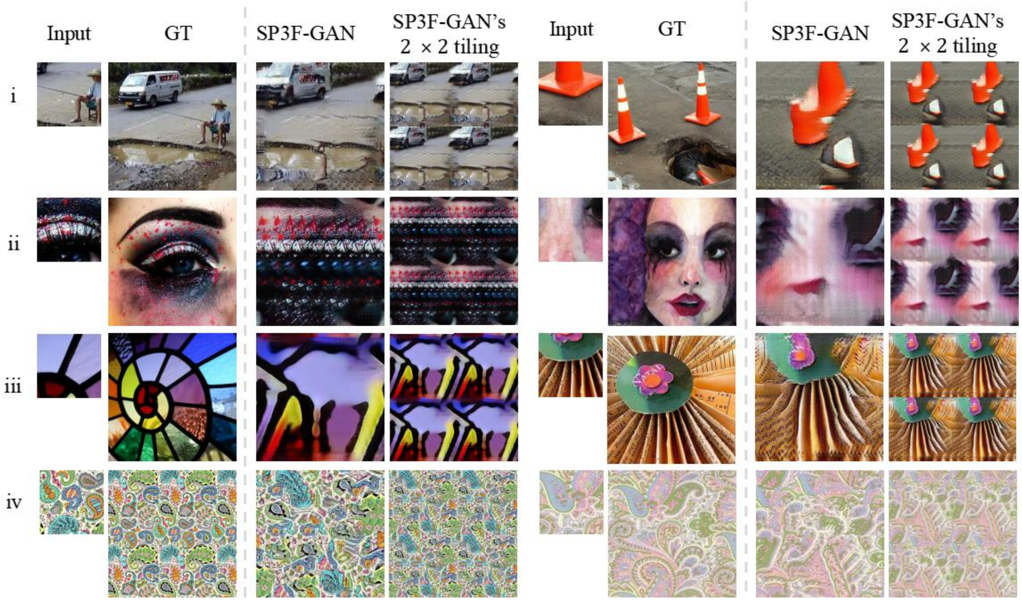


Figure 9. Limitations of the proposed seamless texture maps synthesis model. From left to right: input texture, GT texture, synthesized texture

## 5. CONCLUSION AND FUTURE WORK

In this paper, we propose an automatic method called SP3F-GAN (Seamless Patterns for 3D Fashion GAN) for synthesizing seamless textures for multi-layer texture maps. The proposed SP3F-GAN combines recent advancements in deep texture synthesis, adversarial neural networks, and latent space feature editing to achieve seamless texture synthesis, enabling an end-to-end tileable texture synthesis method without manual input. In addition, we introduce a total variation component in the loss function to penalize texture discontinuities, further improving the tileability of synthesized textures. Through extensive experiments on the describable texture dataset (DTD), we demonstrate that the SP3F-GAN generates seamless texture maps with superior visual quality, particularly in terms of preserving fine details. When compared to traditional methods, our approach achieves global seamlessness in the synthesized textures. And, in comparison to recent advanced GAN-based and Neural Cellular Automata models, SP3F-GAN better preserves the semantic properties of textures while being able to synthesize multiple texture maps simultaneously. About the proposed method model itself, it is allowed to edit the feature map of the latent space to suit different image generation tasks. In addition, the SP3F-GAN can be transferred to other generative models (e.g., diffusion model) because the adversarial expansion framework, although powerful, also has some potential shortcomings. In the future, we will try to optimize visual quality in terms of detail preservation by adjusting the network architecture or using a more stable generative model.

## ACKNOWLEDGEMENT

The work described in this paper was funded in part by a grant from the Laboratory for Artificial Intelligence in Design (Project Code: RP1-1) under the InnoHK Research Clusters, Hong Kong Special Administrative Region Government and by the Innovation and Technology Commission of Hong Kong, under grant ITP/028/21TP.

## REFERENCES

Akl, A. et al. (2018). A survey of exemplar-based texture synthesis methods. *Computer Vision and Image Understanding*, Vol. 172, pp. 12-24.

Barnes, C., Shechtman, E., Finkelstein, A. and Goldman, D. B. (2009). PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics*, Vol. 28, No. 3.

Bergmann, U., Jetchev, N. and Vollgraf, R. (2017). Learning texture manifolds with the periodic spatial GAN. *arXiv preprint arXiv:1705.06566*.

Cimpoi, M. (2014). Describing textures in the wild. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. https://doi.org/10.1109/CVPR.2014.461

*CLO3D* (2023). Retrieved in 2023 from https://www.clovirtualfashion.com/

Cohen, M. F., Shade, J., Hiller, S. and Deussen, O. (2003). Wang tiles for image and texture generation. *ACM transactions on graphics (TOG)*, Vol. 22, No. 3, pp. 287-294.

Dong, W., Zhou, N. and Paul, J.-C. (2007). Optimized tile-based texture synthesis. *Proceedings of Graphics Interface 2007*, pp. 249-256.

Efros, A. A. and Freeman, W. T. (2001). *Image Quilting for Texture Synthesis and Transfer*. *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pp. 341-346.

Efros, A. A. and Leung, T. K. (1999). Texture synthesis by non-parametric sampling. *Proceedings of the seventh IEEE international conference on computer vision*. https://doi.org/10.1109/ICCV.1999.790383

Frühstück, A., Alhashim, I. and Wonka, P. (2019). Tilegan: Synthesis of large-scale non-homogeneous textures. *ACM transactions on graphics (TOG)*, Vol. 38, No. 4.

Galerne, B., Gousseau, Y. and Morel, J.-M. (2010). Random phase textures: Theory and synthesis. *IEEE transactions on image processing*, Vol. 20, No. 1, pp. 257-267.

Gatys, L., Ecker, A. S. and Bethge, M. (2015). Texture synthesis using convolutional neural networks. *Advances in neural information processing systems*, Vol. 28.

Goodfellow, I. J. (2014). Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, *1406*.

Guo, Y. (2020). Materialgan: Reflectance capture using a generative svbrdf model. *arXiv preprint arXiv:2010.00114*.

Hertz, A., Hanocka, R., Giryes, R. and Cohen-Or, D. (2020). Deep geometric texture synthesis. *arXiv preprint arXiv:2007.00074*.

Huang, X., Liu, M.-Y., Belongie, S. and Kautz, J. (2018). Multimodal unsupervised image-to-image translation. *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 179-196.

Isola, P., Zhu, J.-Y., Zhou, T. and Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/CVPR.2017.632

Jetchev, N., Bergmann, U. and Seward, C. (2017). GANosaic: Mosaic creation with generative texture manifolds. *arXiv preprint arXiv:1712.00269*.

Jetchev, N., Bergmann, U. and Vollgraf, R. (2016). Texture synthesis with spatial generative adversarial networks. *arXiv preprint arXiv:1611.08207*.

Karras, T., Aila, T., Laine, S. and Lehtinen, J., (2017). Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*.

Karras, T., Laine, S., & Aila, T. (2019). A style-based generator architecture for generative adversarial networks. *Proceedings of the IEEE/CVF Conference on Computer vision and Pattern Recognition*,

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., & Aila, T. (2020). Analyzing and improving the image quality of stylegan. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.* https://doi.org/10.1109/TPAMI.2020.2970919

Kaspar, A. (2015). Self tuning texture optimization. *Computer Graphics Forum*, Vol. 34, Issue 2. http://dx.doi.org/10.1111/cgf.12565

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Krizhevsky, A., Sutskever, I. and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, Vol. 25.

Kwatra, V. et al. (2003). Graphcut textures: Image and video synthesis using graph cuts. *ACM transactions on graphics (TOG)*, Vol. 22, No. 3, pp. 277-286.

Larsen Greiner, T. A. et al. (2022). Unsupervised deep learning with higher-order total-variation regularization for multidimensional seismic data reconstruction. *Geophysics*, Vol. 87, No. 2, V59-V73.

Levin, A., Zomet, A., Peleg, S. and Weiss, Y. (2004). Seamless image stitching in the gradient domain. *Computer Vision-ECCV 2004: 8th European Conference on Computer Vision*. Prague, Czech Republic, Vol. 3024, pp. 377-389.

Li, C. and Wand, M. (2016). Combining markov random fields and convolutional neural networks for image synthesis. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2479-2486

Li, Z., Sunkavalli, K., and Chandraker, M. (2018). Materials for masses: SVBRDF acquisition with a single mobile phone image. *Proceedings of the European Conference on Computer Vision (ECCV)*.

Liang, L. et al. (2001). Real-time texture synthesis by patch-based sampling. *ACM Transactions On Graphics (TOG)*, Vol. 20, No. 3, pp. 127-150.

Lin, J., Sharma, G. and Pappas, T. N. (2023). Toward universal texture synthesis by combining texton broadcasting with noise injection in StyleGAN-2. *e-Prime-Advances in Electrical Engineering, Electronics and Energy*, Vol. 3, 100092.

Liu, G. et al. (2020). Transposer: Universal texture synthesis using feature maps as transposed convolution filter. *arXiv preprint arXiv:2007.07243*.

Lu, S., Mok, P. Y. and Jin, X. (2017). A new design concept: 3D to 2D textile pattern design for garments. *Computer-Aided Design*, Vol. 89, pp. 35-49.

Mardani, M. et al. (2020). Neural ffts for universal texture image synthesis. *Advances in neural information processing systems*, Vol. 33, pp. 14081-14092.

Montes, R. and Ureña, C. (2012). *An overview of BRDF models*. University of Grenada (Technical Report LSI-2012-001).

Niklasson, E., Mordvintsev, A., Randazzo, E. and Levin, M. (2021). Self-organising textures. *Distill*, Vol. 6, No. 2. http://doi.org/10.23915/distill.00027.003

Portilla, J. and Simoncelli, E. P. (2000). A parametric texture model based on joint statistics of complex wavelet coefficients. *International Journal of Computer Vision*, Vol. 40, pp. 49-70.

Rabin, J., Peyré, G., Delon, J. and Bernot, M. (2012). Wasserstein barycenter and its application to texture mixing. *Scale Space and Variational Methods in Computer Vision: Third International Conference, SSVM 2011*, Ein-Gedi, Israel. Revised Selected Papers 3.

Rodriguez-Pardo, C. and Garces, E. (2022). SeamlessGAN: Self-Supervised Synthesis of Tileable Texture Maps. *arXiv preprint arXiv:2201.05120*.

Ronneberger, O., Fischer, P. and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015: 18th International Conference*. Munich, Germany, pp. 234-241.

Shapiro, L. S. (1995). *Affine analysis of image sequences*. Cambridge University Press.

Sterken, C. and Manfroid, J. (1992). Color transformation. In C. Sterken and J. Manfroid (eds.) *Astronomical Photometry: A Guide*. Springer Dordrecht, pp. 119-125.

Wang, Z., Bovik, A. C., Sheikh, H. R. and Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, Vol. 13, No. 4, pp. 600-612.

Zhan, F. et al. (2022). Marginal contrastive correspondence for guided image generation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/CVPR52688.2022.01040

Zhang, R. et al. (2018). The unreasonable effectiveness of deep features as a perceptual metric. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/CVPR.2018.00068

Zhou, Y. et al. (2018). Non-stationary texture synthesis by adversarial expansion. *arXiv preprint arXiv:1805.04487*.

Zhu, J.-Y., Park, T., Isola, P. and Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. *Proceedings of the IEEE International Conference on Computer Vision*. https://doi.org/10.1109/ICCV.2017.244