e-Prime - Advances in Electrical Engineering, Electronics and Energy 3 (2023) 100092

Contents lists available at ScienceDirect

e-Prime - Advances in Electrical Engineering, Electronics and Energy

journal homepage: www.elsevier.com/locate/prime

Toward universal texture synthesis by combining texton broadcasting with noise injection in StyleGAN-2

Jue Lin [a,*], Gaurav Sharma [b], Thrasyvoulos N. Pappas [a]

[a] ECE Dept., Northwestern University, Evanston, IL, USA
[b] ECE Dept., University of Rochester, Rochester, NY, USA

ARTICLE INFO

Keywords:
Texture analysis and synthesis
Generative adversarial network
2010 MSC:
68U10
94A12

ABSTRACT

We present a universal texture synthesis approach that incorporates a novel multiscale texton broadcasting module in the StyleGAN-2 framework. The texton broadcasting module introduces an inductive bias, enabling generation of a broader range of textures, from those with regular structures to completely stochastic ones. To train and evaluate the proposed approach, we construct a comprehensive high-resolution dataset, NUUR-Texture500, that captures the diversity of natural textures as well as stochastic variations within each perceptually uniform texture. Experimental results demonstrate that the proposed approach yields significantly better quality textures than the state of the art. The ultimate goal of this work is a comprehensive understanding of texture space.

1. Introduction

Texture is an important visual attribute for human perception and image analysis, as it provides critical information for material appearance, understanding, and characterization [1]. Texture understanding, and texture analysis/synthesis in particular, is important for a variety of applications, including image analysis and compression, computer graphics, virtual reality, and human-computer interaction. The study of texture analysis/synthesis must take into account the stochastic nature of texture and human perception, which is the ultimate judge of texture quality. Accordingly, a number of authors have proposed algorithms for texture analysis/synthesis that are based on multiscale frequency decompositions, which have been used to model early visual processing in the brain [2–8]. On the other hand, the stochastic nature of texture necessitates a statistical approach for texture analysis. The most complete parametric approach for texture analysis/synthesis has been proposed by Portilla and Simoncelli [8], who developed a statistical model for synthesizing a broad set of textures based on a steerable filter decomposition. Even though their goal was to provide a universal statistical model that parametrizes the space of visual textures, it falls short of successfully modeling all textures. Thus, the complete mathematical and perceptual characterization of texture remains an open problem.

The resurgence of neural networks has stimulated broad interest in both academia and industry, promising to push the frontiers in a wide variety of research areas [9–11]. One of the most successful models is the generative adversarial network (GAN), which has yielded impressive results in numerous applications [12–14], such as generation of human faces, anime characters, objects and scenes, image-to-image translation, image super-resolution, and inpainting. However, the problem of texture [modeling] synthesis has not received as much attention. There has been work on texture recognition and classification (e.g., [15,16]), which is a different problem, even though it can be considered as another aspect of texture modeling. The focus of this work is on utilizing the GAN framework for texture synthesis, and ultimately, a more complete mathematical and perceptual characterization of the space of visual textures. We present a new approach for universal texture synthesis that introduces a multiscale texton broadcasting module in the StyleGAN-2 framework, which enables the generation of a wide variety of textures, both regular and stochastic.

A generally accepted definition of visual texture is an image that is spatially homogeneous and usually contains repeated elements, often with random variations in position, orientation, and color [8]. The repeated elements in a texture are commonly referred to as *textons,* a term introduced by Bela Julesz, one of the pioneers of texture analysis and perception [17]. Texture appearance can range from completely regular periodic structure to completely random variations, and typically consists of both periodic structure and stochastic variations. However, in our experiments we found that the failure of the

* Corresponding author.
E-mail address: jue.lin@u.northwestern.edu (J. Lin).

https://doi.org/10.1016/j.prime.2022.100092
Received 16 June 2022; Received in revised form 30 September 2022; Accepted 25 November 2022
Available online 5 December 2022

StyleGAN-2 framework to capture the periodic aspect of textures manifests itself in two ways: first, the trained model generates disproportionately fewer periodic textures when randomly sampled from the latent space; and second, once we identify a latent vector corresponding to a periodic texture, injecting the model with different samples of multi-scale noise cannot produce distinguishable texture crops, that is, the synthesized periodic structure is often "anchored" in a fixed location. We will analyze both of these shortcomings and will show that they can be mitigated by the introduction of the texton broadcasting module.

The main contributions of this work are the following:

- We propose a novel multi-scale texton broadcasting module for StyleGAN-2 that, in combination with the noise injection, provides appropriate inductive bias to enable universal high-quality texture synthesis, spanning from regular to completely stochastic textures.
- Through extensive experiments, we demonstrate that the proposed texton broadcasting module significantly enhances texture generation, improving the representation of different textures as well as variations of individual textures, that is, different crops of a perceptually uniform textures, which we call *identical textures* [18]. In addition, we introduce a metric for quantifying the diversity of the synthesized identical textures.
- For effective training and evaluation, we created NUUR-Texture500 [19], a comprehensive dataset of high-resolution textures representative of the diversity of natural textures as well as the variations within individual textures.
- We demonstrate that unlike conventional analysis-synthesis techniques, traversals between two anchor textures in our trained latent space exhibit smooth transitions between homogeneous textures instead of incoherent spatial mixtures.

Figure 1 illustrates sample results obtained with the proposed approach. The three textures shown are obtained by sampling the latent space representation and are chosen to highlight the ability of the proposed technique to produce a wide range of textures, from stochastic to regular (periodic). Also, though the generative model is trained on 256 $\times$256 texture crops, it can produce textures on an arbitrary canvas size, as illustrated by the examples where the three synthesized textures are shown for canvases with sizes of 256 $\times$ 256 and 512 $\times$ 512 pixels.

2. Related work

2.1. Texture analysis and synthesis

As mentioned earlier, traditional parametric approaches for texture analysis/synthesis have been primarily based on subband decompositions. Heeger and Bergen [4] matched the histograms of a steerable filter decomposition to achieve impressive texture synthesis results; however, their approach is limited to stochastic textures. Portilla and Simoncelli [8] developed a more elaborate model that relies on a wide variety of subband statistics for synthesizing a much broader set of textures. Their goal is to parametrize the space of visual textures based on a universal statistical model. Parametric approaches based on Markov Random Fields (MRFs) have also shown considerable potential for texture synthesis [20,21]. Exemplar-based texture synthesis can be found in [22–25]; however, this does not involve any texture modeling.

In recent years, a diverse collection of deep learning-based approaches have also been proposed for texture analysis/synthesis [26–28]. The majority of these approaches rely on convolutional neural networks, which have enjoyed wide success in image processing and computer vision tasks. Gatys et al. [26] use a pretrained VGG-19 network [10] to extract feature vectors at multiple spatial scales in the network hierarchy from a given texture and then, starting with a random image, synthesize another texture that matches, for each scale, the Gram-matrix of feature vector inner products. Ulyanov et al. [28,29] use a fast feed-forward generative network to achieve similar performance. Li et al. [30] further develop a feed-forward generative network to synthesize multiple diverse textures. Rodriguez-Pardo et al. [31] use content loss [32] to extract a template of periodic textures.

In addition to the mainstream convolutional neural architectures, image synthesis has also utilized the recently introduced implicit neural representation (INR) methods [33–36]. Instead of relying on successive convolutional layers that progressively grow the image size, these methods use a multilayer perceptron to predict image RGB values corresponding to input pixel coordinates [37–40]. Although, these methods have not been developed and explored specifically for texture synthesis, particular instances [41] have provided capabilities of interest in texture synthesis, such as the ability to extrapolate beyond defined image boundaries.

2.2. Generative adversarial networks

Goodfellow et al. [42] introduced an adversarial formulation for training a generative model, whereby a second discriminator network provides feedback by determining whether a generated image comes from the actual data distribution or not. The WGAN [43] uses the Wasserstein (or earth mover) distance between probability distributions, which improves stability of learning and alleviates mode collapse. Further improvements come from alternative formulations of Lipschitz continuity for the WGAN, e.g., gradient penalty [44] and spectral normalization [45].

The application of GANs to texture synthesis was introduced by Jetchev et al. [46], who proposed the spatial GAN for synthesis of textures of arbitrary size. However, like Gatys et al. [26], the functionality of the spatial GAN is limited to producing equivalent textures, that is, it generates one model per texture. The periodic spatial GAN (PSGAN) by Bergmann et al. [47] represents the first attempt to learn a latent space that is capable of generating periodic textures by injecting a periodic pattern (with a random phase term) at the bottom of the generator network. However, they trained on a very small dataset and, as we will show below, the quality of the resulting textures is mixed.

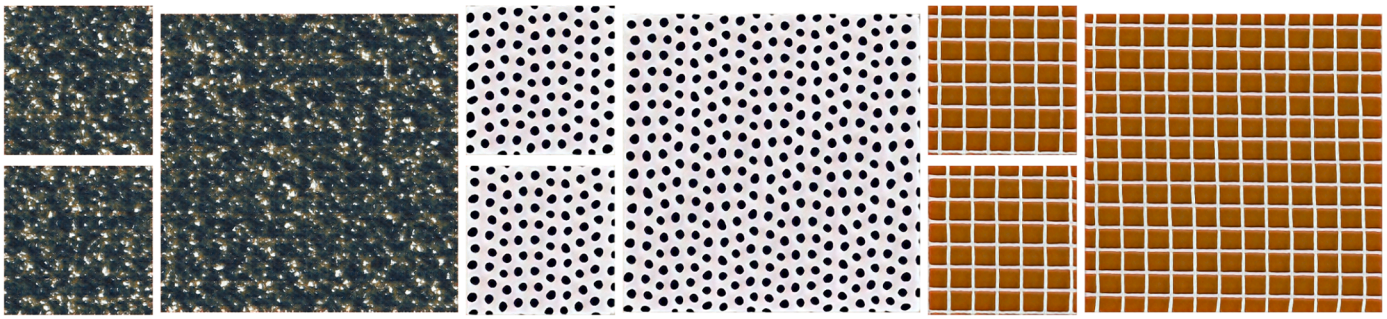


Fig. 1. Sample textures synthesized using the proposed approach for canvas sizes of 256 $\times$ 256 and 512 $\times$ 512 pixels.

2.3. StyleGAN models

Building on the progressive-GAN [48], StyleGAN, proposed by Karras et al. [12], introduces an intermediate latent space, which is used to adjust the style of the image at each convolution layer, and also adds explicit noise injection at each layer. This allows the disentanglement of global features (like pose, face shape, and human identity) and local stochastic variations (like hair and skin texture). StyleGAN-2 [49] was proposed to address some noticeable blob-like visual artifacts in StyleGAN, by redesigning the normalization and eliminating progressive training.

Following the work of Karras et al., which was applied to faces, objects, or scenes, one line of research sought to interpret the latent space induced by StyleGAN-like models. Built upon loss functions containing location information or pretrained attribute classifiers, an input image can be inverted into a latent code $\mathbf{z}$, and visual property manipulation can be achieved via navigating in the latent space [50,51]. However, due to lack of an equivalent model, a proper GAN inversion technique is still missing for textures. Although loss functions for faces, objects, or scenes have been proposed, they rely on pixel-to-pixel correspondences, ill-suited for textures because of their stochastic nature.

Another line of research sought to investigate the impact of the internal components of the StyleGAN models. Xu et al. [52] found that zero padding implicitly encodes location, which works for faces, objects, or scenes but is not desirable for textures. Choi et al. [53] addressed the spatial bias in StyleGAN-2 by adding sinusoidal embeddings, commonly used in transformers [54–56].

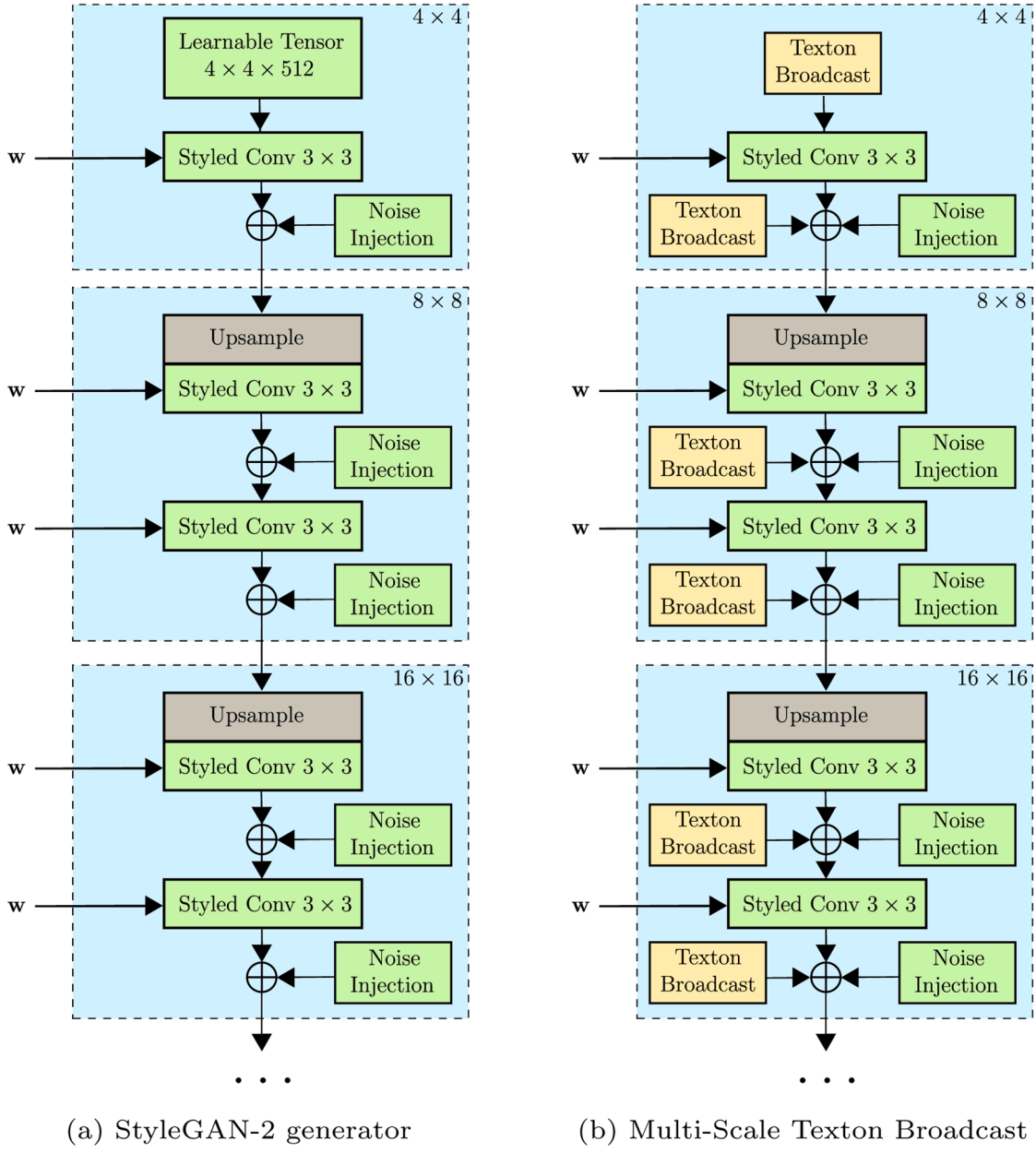


Fig. 2. Comparison between StyleGAN-2 and our proposed modifications. All feature maps within the same blue area, enclosed with dashed line, share the same spatial size, indicated on the upper right corner. The "⊕" is an element-wise sum. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

3. Proposed method

3.1. Preliminaries

We denote by $\Omega = \{\Omega_1, \Omega_2, \cdots\}$ the set of all textures. Intuitively, each texture Ω_i has a set of basic elements or *textons* [57] (e.g., a brick in a wall). We can aggregate textons from Ω and establish a *universal texton codebook* T_Ω [58]. Any texture Ω_i can be represented as a spatial repetition of textons drawn from T_Ω, and the selected textons are adjusted to fit certain properties (e.g., shapes). Moreover, stochastic variation is often present within each Ω_i (e.g., layout of bricks). We can therefore model texture distribution in two parts: *inter*-texture distribution $P_{\Omega_i \sim \Omega}(\Omega_i)$ for distinct textures, and *intra*-texture distribution $P_{\mathbf{I} \sim \Omega_i}(\mathbf{I}_i)$ conditioned on the same texture Ω_i, where we assume Ω_i is sufficiently large and $\mathbf{I}$ is a random crop from Ω_i.

Conceptually, a StyleGAN-2 generator $G_{\boldsymbol{\theta}}$ can approximate both $P_{\Omega_i \sim \Omega}(\Omega_i)$ and $P_{\mathbf{I} \sim \Omega_i}(\mathbf{I})$ with $P_{\mathbf{z},\mathbf{n}}(G_{\boldsymbol{\theta}}(\mathbf{z},\mathbf{n}))$ and $P_{\mathbf{n}|\mathbf{z}}(G_{\boldsymbol{\theta}}(\mathbf{z},\mathbf{n})|\,\mathbf{z})$ respectively, where $\mathbf{z}$ is drawn from a D-dimensional normal distribution and $\mathbf{n}$ is the multi-scale spatial noise. Due to randomness in the recurrence of textons, most textures exhibit *stochasticity* as well as *periodicity*. The stochasticity is well captured by $\mathbf{n}$, producing subtle changes in textured regions, e.g., hair and skin. However, our experiments show that the StyleGAN-2 generator is biased towards synthesizing stochastic textures, even though regular textures constitute a comparable portion of the training set. We refer to this failure to adequately represent the space of different textures as *inter-texture mode collapse*. Moreover, the injected noise $\mathbf{n}$ is inadequate for rendering alternative visually equivalent versions of periodic textures, such as those obtained by cropping with different spatial shifts from a large texture canvas. We refer to this failure to adequately represent variations of a texture in the synthesized set as *intra-texture mode collapse*. Empirically we found that the first problem is relatively easier to address.

The intra-texture mode collapse suggests a strong entanglement between spatial location and latent space. Such entanglement is arguably acceptable for images in other domains, e.g., face positioned in the center, but the spatial layout of textons should be stochastic. Most operations in StyleGAN-2 (see Fig. 2a) do not explicitly encode spatial information, e.g., convolutions, upsampling. We have identified the bottom $512 \times 4 \times 4$ tensor and zero-padding in coarse layers as the causes of spatial anchoring of visible structures (see Fig. 5a).

3.2. Texton broadcasting module

To capture the periodic nature of textures, we design a *texton broadcasting (TB)* module that simulates the spatial repetition of physical textons, as illustrated in Fig. 3. First, a trainable texton $\mathbf{v}_i$ is replicated along spatial dimensions. Then the intensity of each $\mathbf{v}_i$ is modulated with respect to a *broadcast map (BM)*, modeled as a 2D sinusoidal wave:

$$\mathrm{BM}_i(h,w) = \mathbf{A}_i \sin\left(2\pi\varsigma(\mathbf{f}_i)^T \begin{bmatrix} h \\ w \end{bmatrix} + \boldsymbol{\varphi}_i + \Delta\right) + \mathbf{B}_i \tag{1}$$

where $\forall i \in \{1, 2, \ldots, P\}$, $\mathbf{f}_i = [f_{ih}, f_{iw}]^T$, $\boldsymbol{\varphi}_i$, $\mathbf{A}_i$ and $\mathbf{B}_i$ represent the frequency, initial phase, amplitude, and offset of the 2D sine, all of which are trainable parameters, and P denotes the total number of textons in a TB module. We use $[h,w]^T \in \{1,2,\ldots,H\} \times \{1,2,\ldots,W\}$ as the spatial coordinate vector; H and W can be dynamically sized but are fixed during training. We use an element-wise sigmoid function $\varsigma(\cdot)$ to map $[f_{ih}, f_{iw}]^T$ into the interval $(0,1)$, since discrete-time frequency is periodic in ω with period 2π, i.e., $\sin(\omega n) = \sin((\omega + 2\pi)n), \forall n \in \mathbb{Z}$. Note that Δ is uniformly sampled from $[0, 2\pi)$ to simulate a random shift, and is shared among all BM_i within the same module to have a unified phase control. The output $\mathbf{Y}$ of a module is the sum across all broadcast/modulated $\mathbf{v}_i$:

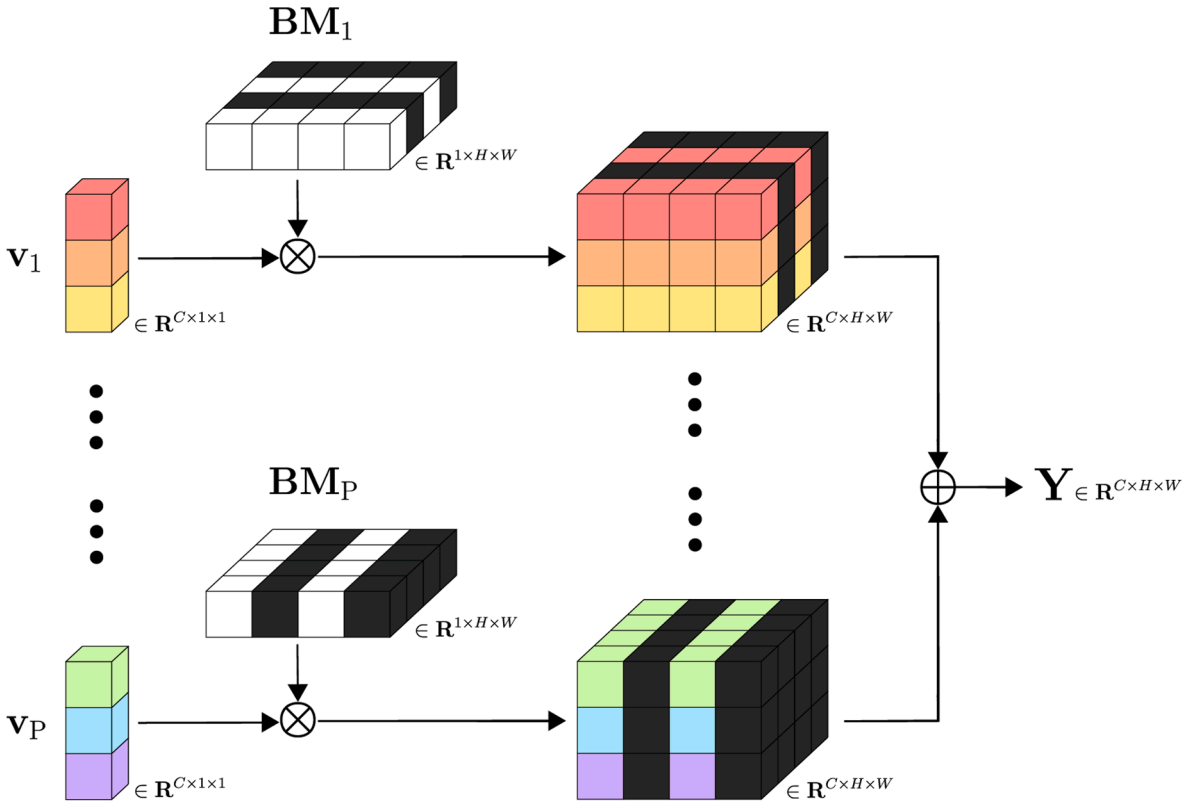


Fig. 3. Mechanism of Texton Broadcasting. Each cubelet represents a scalar, the value of which is indicated by its color. For ease of illustration, we assume a black cubelet $= 0$, a white cubelet $= 1$, and other colors are arbitrary. The $\otimes$ duplicates $\mathbf{v}_i$ spatially and $\mathbf{BM}_i$ along channel dimension, followed by an element-wise multiplication. The $\oplus$ is an element-wise sum.

$$\mathbf{Y}(c,h,w) = \sum_{i=1}^{P} \mathbf{v}_i(c) \otimes \mathrm{BM}_i(h,w) \tag{2}$$

where $\otimes$ combines replication of all textons $\mathbf{v}_i$ with the modulation by BM_i, as illustrated in Fig. 3. Note that the use of a random phase Δ is critical, otherwise the module is just another deterministic spatial anchor.

3.3. Multi-scale texton broadcasting

By replacing the bottom $512 \times 4 \times 4$ tensor with a TB module, the new model is now capable of producing images of variable sizes. However, the zero padding at the bottom layers still causes a residual spatial anchoring effect when synthesizing images of larger spatial size. To mitigate this issue, we couple each noise injection (NI) module with a TB module (see Fig. 2b), up to and including layers with spatial size of 64×64. Such hierarchical placement of NI and TB is aligned with the multi-scale nature of textures. The spatial size H and W of each TB module is configured to match the spatial size at its corresponding layer except the bottom one, which is set to 4×4 to produce a final 256×256 image compatible with the discriminator during training.

3.4. Training objective functions

The inter-texture mode collapse is in fact closely related to the general definition of mode collapse in the literature, where a model yields only a few distinguishable images. We adopt the Wasserstein distance as the loss function, and impose a gradient penalty [44] to enforce the Lipschitz continuity on the discriminator network (or critic) $D_{\boldsymbol{\phi}}$, parametrized by $\boldsymbol{\phi}$. The losses for $D_{\boldsymbol{\phi}}$ and $G_{\boldsymbol{\theta}}$ are given as:

$$\mathscr{L}(\boldsymbol{\phi}) = \mathbb{E}_{(\mathbf{z},\mathbf{n},\Delta_G)}\{D_{\boldsymbol{\phi}}(G_{\boldsymbol{\theta}}(\mathbf{z},\mathbf{n},\Delta_G))\} - E_{\Omega_i \sim \Omega}\{\mathbb{E}_{\mathbf{I}|\Omega_i}\{D_{\boldsymbol{\phi}}(\mathbf{I})\}\} \tag{3}$$

$$\mathscr{L}(\boldsymbol{\theta}) = -\mathbb{E}_{(\mathbf{z},\mathbf{n},\Delta_G)}\{D_{\boldsymbol{\phi}}(G_{\boldsymbol{\theta}}(\mathbf{z},\mathbf{n},\Delta_G))\} \tag{4}$$

where $\mathbf{z}$ and $\mathbf{n}$ are drawn from a normal distribution, Δ_G denotes the set of all Δ injected across $G_{\boldsymbol{\theta}}$. Note that in (3), multiple crops $\mathbf{I}$ are sampled from the same texture Ω_i, which helps $D_{\boldsymbol{\phi}}$ learn intra-texture distribution explicitly.

3.5. GAN inversion for textures

Once a reliable generator is obtained, texture analysis can be performed via GAN inversion. Inverting a latent space trained on textures poses distinct challenges. Most generative models produce images with well-defined objects of interest, e.g., a face or bed. However, for textures, everything is of interest and the stochastic placement of textons renders commonly used location-wise losses, such as L2 loss and content loss [32], inapplicable. Instead, we use the style loss by Gatys et al. [26] who extract the Gram matrices of feature maps from a pretrained VGG-16 network Φ[10]. The process of searching for an optimal latent code $\mathbf{z}^*$ for any texture Ω_i is given via:

$$\mathscr{L}(\mathbf{I}_1,\mathbf{I}_2) = \sum_l \left[\frac{\Phi_l(\mathbf{I}_1)\Phi_l(\mathbf{I}_1)^T - \Phi_l(\mathbf{I}_2)\Phi_l(\mathbf{I}_2)^T}{C_l \times N_l^2}\right]^2 \tag{5}$$

$$\mathbf{z}^* = \underset{\mathbf{z}}{\operatorname{argmin}} \mathbb{E}_{(\mathbf{n},\Delta_G,\mathbf{I}\sim\Omega_i)}[\mathscr{L}(G_{\boldsymbol{\theta}}(\mathbf{z},\mathbf{n},\Delta_G),\ \mathbf{I})] \tag{6}$$

where l is layer index, and Φ_l, N_l, C_l represent feature maps, spatial size and channel at the lth layer of Φ, respectively. Note that a common practice in the literature of GAN inversion is to use $\mathbf{w}$ or $\mathbf{w}^+$ as the optimization variable instead of the raw latent code $\mathbf{z}$, where $\mathbf{w}$ comes from the style mapping network with $\mathbf{z}$ as input, and $\mathbf{w}^+$ is the aggregation of all $\mathbf{w}$ across different layers.

4. Experimental results

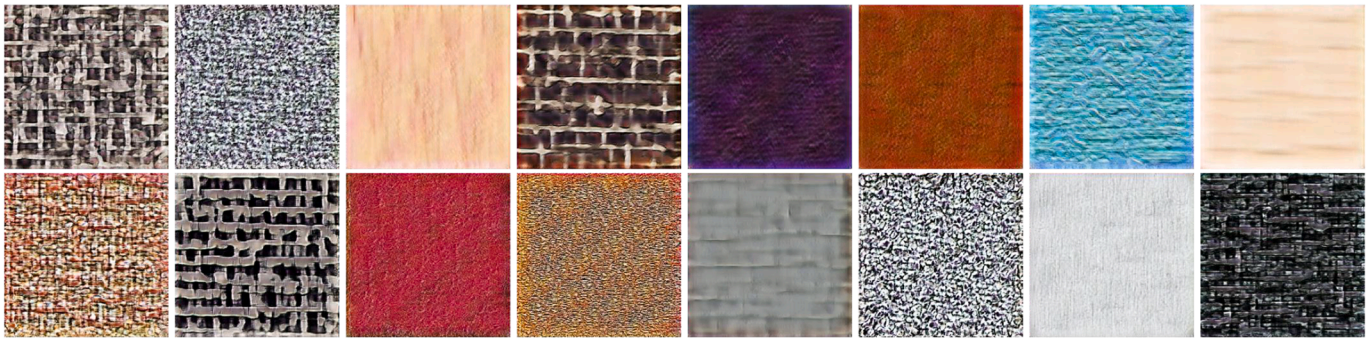
4.1. Dataset construction

There exist multiple texture datasets in the literature. However, most of these datasets are not suitable for our texture synthesis application setting and do not conform to the definition of texture in the introduction, which we, and other texture analysis-synthesis researchers, have adopted motivated by the application scenario. For example, the Describable Texture Dataset (DTD) [59], the Amsterdam Library of Textures (ALOT) [60], the Flickr Material Dataset (FMD) [61], the Materials in Context 2500 (MINC-2500) [62], and the Ground Terrain in Outdoor Scenes (GTOS) [63], contain images with multiple textures objects, and are thus not spatially homogeneous textures. Moreover, some of the existing datasets (e.g., CUReT [64,65] and KTH-TIP [66]) either lack inter-texture diversity or sufficient spatial size for obtaining multiple independent crops, which are essential for training the network. To address such limitations, we collect a more comprehensive dataset of textures, which we make publicly available under the name NUUR-Texture500 [19]. We include a texture image into the dataset if it fits the following criteria: (1) It is perceptually uniform; (2) it contains sufficient independent 256×256 crops to learn the intra-texture distribution; (3) each 256×256 crop contains enough texton repetitions (at least 5 in each dimension) to form a texture; and (4) it is under either Creative Commons Public Domain license (CC0) or custom website license for free academic use. After extensive search on multiple stock image websites, we obtained 500 quality texture images, ranging from natural to artificial, periodic to stochastic, and fine-grained to coarse, and each image contains roughly 20 to 50 independent crops.

4.2. Training settings

In our experiments, we uniformly sampled 2 crops of 256×256 from each texture in a mini-batch of 8 distinct textures, to explicitly enforce the learning of intra-texture distribution on the discriminator network $D_{\boldsymbol{\phi}}$. The same trick can be applied to the generator network $G_{\boldsymbol{\theta}}$ by feeding multiple noise samples $\mathbf{n}$ conditioned on the same latent variable $\mathbf{z}$, but this resulted in a slower convergence of the generator and no significant performance gain. Hence, to prevent the mode collapse, we adopted the Wasserstein distance in (3) and (4) as the loss with gradient penalty $= 0.01$ to impose Lipschitz continuity, and the discriminator parameters $\boldsymbol{\phi}$ were updated twice, followed by one generator update (3×10^5 generator iterations in total). We disabled the mixing regularization and the path-length regularization as they are time-consuming. For other hyperparameters, we followed the default protocols of StyleGAN-2, including latent space dimensionality $D = 512$, learning rate $= 0.002$, Adam optimizer, and exponential moving average of $G_{\boldsymbol{\theta}}$. All experiments were run on a Nvidia GeForce RTX 3090 with 10,496 CUDA cores and 24 GB GPU memory. Training the generator requires approximately 4 weeks, and average time required to generate a texture was 16.7 ms (computed over 10,000 samples).

Regarding the settings of TB, each module has $P = 16$ learnable texton vectors. We applied the TB module to all layers of spatial size up to 64×64 to prevent high-frequency artifacts. The spatial size of the bottom TB module was set to 4×4 during training, and the number of channels of each texton vector was fixed at $C = 512$. The spatial size as well as channel size of all remaining TB modules were designed to match the feature maps of their preceding blocks of Styled Conv as shown in Fig. 2b. At test time, the final output image size can be varied by simply modifying the spatial size of the bottom module.



(a) PSGAN: suboptimal texture synthesis

(b) StyleGAN-2: visually competitive image quality, but severe intra-texture mode collapse, as will be discussed in later sections

(c) Proposed approach: well defined textures, good balance between stochastic and structured/periodic textures

Fig. 4. Sample textures generated by models trained on the NUUR-Texture500 dataset.

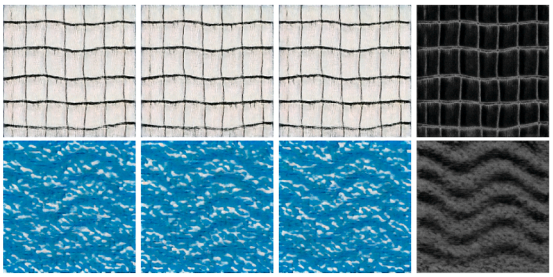
We compare our approach with PSGAN, which also aims to model periodic textures, and the baseline StyleGAN-2. All methods were trained on our dataset, and we provide a list of textures sampled from each method. As shown in Fig. 4, the proposed module outperforms PSGAN and StyleGAN-2 in terms of diversity and image quality. We will provide more samples in the supplementary material. We also evaluate the FID [67] for each model in Table 1. The distribution of the training set was computed by sampling 20 crops from each texture. For each method, 1000 latent codes were sampled, and 20 different texture crops were obtained for each code. Each method was repeated 5 times, and the 95% confidence intervals are provided.

Table 1
Quantitative evaluation of different methods.

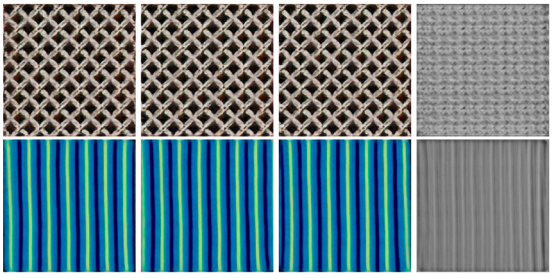
	FID ↓
PSGAN [47]	133.72±2.46
StyleGAN-2 [49]	72.48±1.86
Proposed	**70.05 ± 1.41**
Training set	1.58±1.10

4.2.1. *Intra-texture mode collapse*

Fig. 5 shows that the proposed approach yields a significant improvement in terms of intra-texture diversity, that is, the reconstructed textures do not all correspond to the same crop. To emphasize this point, the figure also shows a map of pixel-wise standard deviations $\sigma_{\mathbf{z}} \in \mathbb{R}^{H \times W}$ conditioned on $\mathbf{z}$ defined by:



(a) StyleGAN-2: Synthesized textures and $\sigma_{\mathbf{z}}$ maps show that most pixels stay invariant



(b) Proposed approach: Synthesized textures and $\sigma_{\mathbf{z}}$ maps show better pixel-wise variation

Fig. 5. Different synthesized textures from the same latent vector, and the associated standard deviation maps $\sigma_{\mathbf{z}}$ on the right. Higher intensity in the $\sigma_{\mathbf{z}}$ map indicates lower likelihood of anchoring artifacts, while darker regions indicate intra-texture mode collapse.

$$\sigma_{\mathbf{z}} = \sqrt{\mathbb{E}_{\mathbf{n}|\mathbf{z}}\left[\left(G_{\boldsymbol{\theta}}(\mathbf{z},\mathbf{n}) - \mathbb{E}_{\mathbf{n}|\mathbf{z}}[G_{\boldsymbol{\theta}}(\mathbf{z},\mathbf{n})]\right)^2\right]} \tag{7}$$

The low intensities in the $\sigma_{\mathbf{z}}$ maps of Fig. 5a indicate point-by-point alignment of the generated textures, and are thus an indication of intra-texture mode collapse, while the significantly higher intensities in the $\sigma_{\mathbf{z}}$ maps of Fig. 5b indicate that the textures are less likely to suffer from anchoring artifacts. For better visualization, all of the $\sigma_{\mathbf{z}}$ maps have been scaled in amplitude by a factor of 2.

To quantify the intra-texture mode collapse of a synthesized texture, we now introduce a novel and intuitive measure we call *thresholded invariant pixel percentage (TIPP)*, which is calculated as

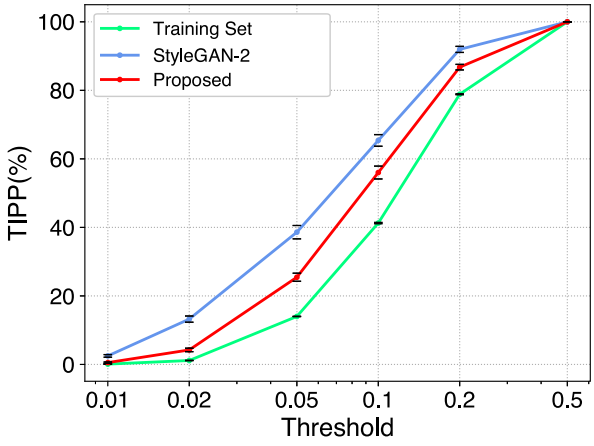


Fig. 6. TIPP measure of intra-texture mode collapse shows that proposed method outperforms StyleGAN-2 across a wide range of thresholds t.

$$\text{TIPP}_t(\mathbf{z}) = \frac{1}{H \times W} \sum_{h}^{H} \sum_{w}^{W} \mathbf{1}(\sigma_{\mathbf{z}}[h,w] \leq t) \tag{8}$$

where $\sigma_{\mathbf{z}} \in \mathbb{R}^{H \times W}$ is the pixel-wise standard deviation defined in (7), t is the threshold, and $\mathbf{1}(\cdot)$ is an indicator function that returns 1 if the condition is met and 0 otherwise. Intuitively, TIPP_t calculates the percentage of pixels with $\sigma_{\mathbf{z}} < t$. Pixels with low standard deviation have a strong invariance, and thus a higher TIPP value indicates worse spatial anchoring artifact.

To evaluate each model, we sampled 1000 latent codes $\mathbf{z}$, synthesized 20 crops per $\mathbf{z}$, and calculated TIPP averaged over the codes $\mathbf{z}$ for different thresholds. TIPP(%) can also be calculated for the training set, where the averaging is over crops rather than latent codes. For that, we sampled 20 crops from each texture in the training set. Fig. 6 shows that the proposed method consistently outperforms StyleGAN-2 in terms of intra-texture mode collapse.

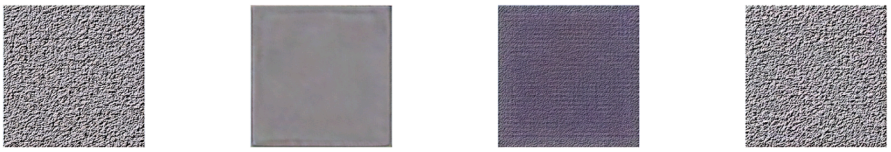
4.2.2. *Inter-texture mode collapse*

We designed an experiment to investigate inter-texture mode collapse. We selected 8 textures of distinguishable properties, and independently sampled 8 fixed latent codes, from which the generator samples during training. Our expectation was that StyleGAN-2 would overfit the data, allocating each latent code to a different texture, which would indicate strong disentanglement between latent code and noise injection. To our surprise, as shown in Fig. 7a, the original StyleGAN-2 strategy with non-saturating loss performs poorly on this small set. We then applied the Wasserstein distance combined with adding Gaussian noise ($\sigma = 0.01$) to the discriminator input and found that it improves diversity as shown in Fig. 7b. Therefore, we adopted the Wasserstein distance and discriminator noise as the default training configuration for both the proposed method and StyleGAN-2 (included in the results shown in Fig. 4).

(a) Original training strategy of StyleGAN-2 (non-saturating loss) shows inter-texture mode collapse

(b) StyleGAN-2 with Wasserstein distance + discriminator noise

Fig. 7. Inter-texture mode collapse experiments.



(a) Ground truth (b) L2 loss (c) Content loss (d) Gram matrices

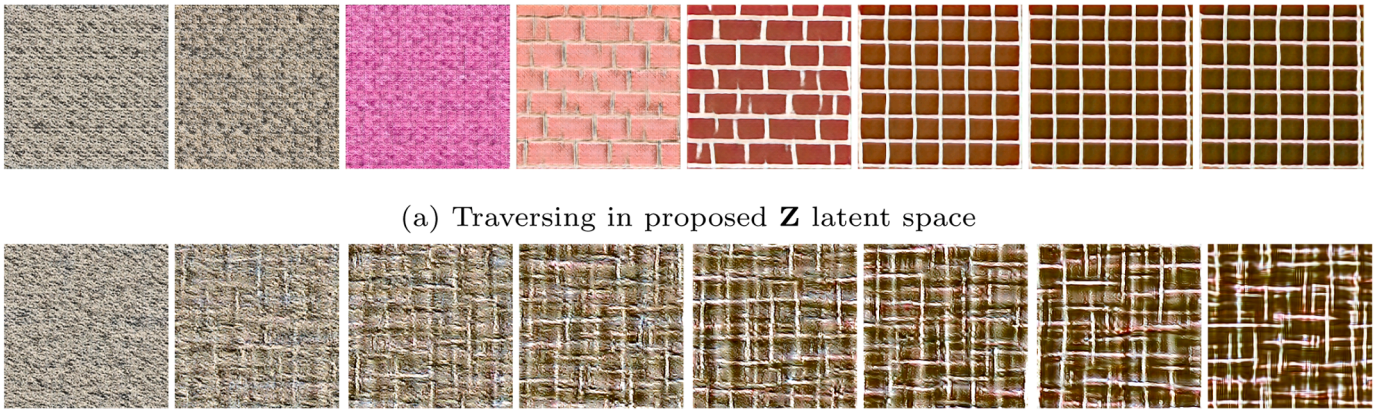
Fig. 8. Loss functions for texture GAN inversion.

4.3. GAN inversion

In this experiment, we used *relu*1_2, *relu*2_2, *relu*3_3 and *relu*4_3 layers to extract feature maps and compute Gram matrices via (6), where *reluX_Y* denotes the output of the *Y*th Rectified Linear Unit(ReLU) at the *X*th spatial scale in a VGG network. For completeness, we also investigated the efficacy of mean-squared loss and content loss [32]. For all losses, the inversion was performed on the same optimization variable $\mathbf{w}$. We avoided the use of $\mathbf{w}^+$ as we prefer a unified global representation for textures. Other shared hyperparameters include learning rate of 0.001, a total of 5000 iterations, and Adam optimizer with default settings. Results are shown in Fig. 8 and are consistent with our expectations, as the characterization of a texture should minimize location-wise correspondence due to its stochastic nature.

4.4. Latent space interpolation

Interpolation in the latent space can be performed by traversing in the $\mathbf{Z}$ or $\mathbf{W}$ space, which is mapped from input $\mathbf{z}$ via a MLP. In both learned latent spaces, there is a gradual transition from one texture to the other via intermediate uniform textures, as shown in Fig. 9. In contrast, the interpolations produced by the classic Portilla and Simoncelli [8] method consist of a mixture of the endpoint images rather than homogeneous intermediate textures.



(a) Traversing in proposed $\mathbf{Z}$ latent space

(b) Based on Portilla and Simoncelli parametrization

Fig. 9. Comparison between texture interpolation based on proposed latent space and Portilla and Simoncelli parametrization [8].

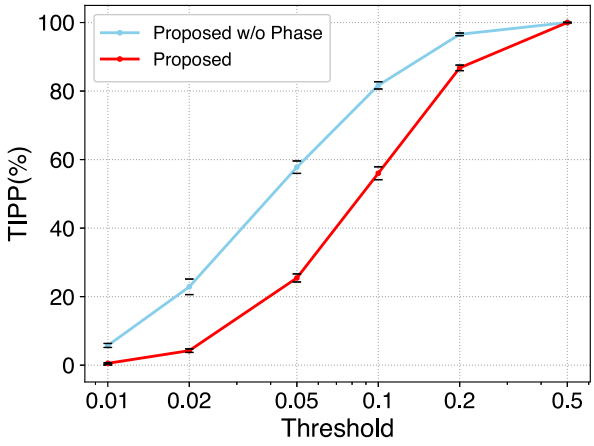


Fig. 10. Disabling random phase Δ significantly degrades TIPP.

4.5. Ablation studies

4.5.1. Is random phase noise Δ needed?

To demonstrate the necessity of Δ, we re-trained the model with fixed phase. As expected, the module degenerates to another form of spatial anchoring, severely damaging the intra-texture diversity. As shown in Fig. 10, such setting degrades TIPP by a considerable margin.

4.5.2. Multi-Scale texton broadcasting

We demonstrated the importance of multi-scale texton broadcasting by removing all but the bottom TB modules. Such an ablated model is capable of generating quality textures with the same size as the training images. However, when generalized to arbitrary sizes, only the 4 corners, shown in Fig. 11a, resemble their low-resolution counterpart, while the model fails to render the central area. We attribute this to the zero-padding at bottom layers, consistent with the study in [52] where zero-padding is shown to have an implicit encoding of location. By introducing the TB module in a multi-scale fashion, the influence of zero-padding can be substantially reduced.

4.5.3. Mapping latent code to TB via a MLP?

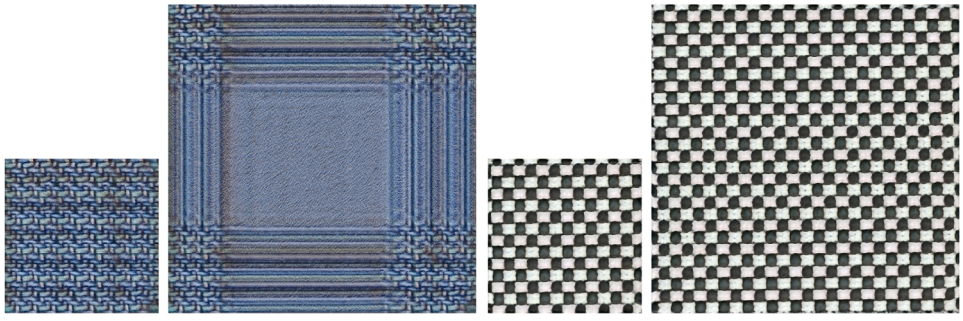
We also conducted experiments with the trainable parameters in the TB module linked to the latent code as in PSGAN via a MLP [47], e.g., $f_i = MLP_{f_i}(\mathbf{z})$. We empirically found that such a model is unstable to train and struggles to converge. We hypothesize that if the module parameters are conditioned on the latent code, then the entanglement between location and latent space is aggravated because the resulting broadcast maps are directly affected by the latent code, and the latent space no longer works in a channel-wise manner, contrary to the design philosophy of StyleGAN.

5. Discussion

While the proposed approach yields promising results, understanding textures remains a challenging task with multiple open subproblems. Despite its powerful representational abilities, the proposed model fails to encompass the training set of 500 textures. The immense diversity across textures as well as within textures poses unique challenges, which results in a relatively high FID for our trained texture synthesis network compared with networks trained for image synthesis in other domains. The FID metric itself relies on the VGG network trained for object recognition and is not ideal for evaluating texture synthesis. Development of a metric better-suited for evaluating texture synthesis methods remains an open problem. Incorporation of texture synthesis within a broader analysis-synthesis framework is also of significant interest as is the exploration of the learned latent space. A recent improvement to StyleGAN-2 has been proposed as StyleGAN-3 [68], which incorporates signal processing improvements to enhance translation/rotation equivariance. One of the changes incorporated in StyleGAN-3 is the elimination of noise injection, which improves generative performance for human face images. However, for the proposed texture synthesis application, this also eliminates an important source of intra-texture variation, i.e., variation that leads to alternative versions of a texture that are visually equivalent in their appearance. Within the scope of the present work, StyleGAN3 was therefore not utilized. Exploring the viability of StyleGAN-3 for texture synthesis is of future interest as is the expansion of the dataset and exploration of the other directions outlined here.

6. Conclusion

We conducted an in-depth investigation of texture synthesis using the StyleGAN-2 network, which demonstrates that the native architecture underrepresents both inter- and intra-texture diversity. Specifically, synthesized periodic textures manifest severe spatial anchoring, whereby there is essentially only one instance of such textures in the synthesized set, with no intra-texture diversity. We showed that the addition of the proposed multiscale texton broadcasting module significantly improves the inter- and intra-texture diversity and also directly provides extrapolation capability, i.e., the ability to generate textures on arbitrarily sized canvases. Remarkably, the latent space of textures, obtained using the proposed texton broadcasting module augmentation for StyleGAN-2, exhibits smooth transitions between homogeneous textures instead of the incoherent mixtures observed with some alternative classical texture synthesis techniques. The proposed work is a step toward universal texture synthesis and a deep neural network framework for unified texture analysis-synthesis. To facilitate further work in this field, we have made available the source code for our implementation [69].



(a) TB Module Only at the Bottom (b) Multi-scale TB Modules

Fig. 11. Multi-scale texton broadcasting experiments.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data and code are publicly available and cited in the article.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.prime.2022.100092.

References

[1] E.H. Adelson, On seeing stuff: the perception of materials by humans and machines, in: B.E. Rogowitz, T.N. Pappas (Eds.), Human Vision and Electronic Imaging VI, Vol. 4299 of Proc. SPIE, San Jose, CA, 2001, pp. 1–12.

[2] D. Cano, T.H. Minh, Texture synthesis using hierarchical linear transforms, Signal Process. 15 (1988) 131–148.

[3] M. Porat, Y.Y. Zeevi, Localized texture processing in vision: analysis and synthesis in Gaborian space, IEEE Trans. Biomed. Eng. 36 (1) (1989) 115–129.

[4] D.J. Heeger, J.R. Bergen, Pyramid-based texture analysis/synthesis. Proc. Int. Conf. Image Processing (ICIP) Vol. III, 1995, pp. 648–651.Washington, DC

[5] J. Portilla, R. Navarro, O. Nestares, A. Tabernero, Texture synthesis-by-analysis based on a multiscale early-vision model, Opt. Eng. 35 (8) (1996) 2403–2417.

[6] S. Zhu, Y.N. Wu, D. Mumford, Filters, random fields and maximum entropy (FRAME): towards a unified theory for texture modeling. IEEE Conf. Computer Vision and pattern Recognition, 1996, pp. 693–696.

[7] J.S. De Bonet, P.A. Viola, A non-parametric multi-scale statistical model for natural images, Adv. Neural Info. Process. Syst. 9 (1997).

[8] J. Portilla, E.P. Simoncelli, A parametric texture model based on joint statistics of complex wavelet coefficients, Int. J. Comput. Vis. 40 (1) (2000) 49–71.

[9] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, Commun. ACM 60 (6) (2017) 84–90.

[10] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition. 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings, 2015.

[11] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

[12] T. Karras, S. Laine, T. Aila, A style-based generator architecture for generative adversarial networks. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 4396–4405.

[13] P. Isola, J.-Y. Zhu, T. Zhou, A.A. Efros, Image-to-image translation with conditional adversarial networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 5967–5976.

[14] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, W. Shi, Photo-realistic single image super-resolution using a generative adversarial network. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 105–114.

[15] M. Cimpoi, S. Maji, I. Kokkinos, A. Vedaldi, Deep filter banks for texture recognition, description, and segmentation, Int. J. Comput. Vis. 118 (1) (2016) 65–94.

[16] Z. Chen, F. Li, Y. Quan, Y. Xu, H. Ji, Deep texture recognition via exploiting cross-layer statistical self-similarity. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 5227–5236.

[17] B. Julesz, Textons, the elements of texture perception and their interactions, Nature 290 (1981) 91–97.

[18] J. Zujovic, T.N. Pappas, D.L. Neuhoff, Structural texture similarity metrics for image analysis and retrieval, IEEE Trans. Image Process. 22 (7) (2013) 2545–2558.

[19] J. Lin, G. Sharma, T. Pappas, NUUR-Texture500: A Diverse Dataset of High Resolution Homogeneous Textures (2022). 10.5281/zenodo.7127079.

[20] E. Levina, P.J. Bickel, Texture synthesis and nonparametric resampling of random fields, Ann. Stat. 34 (4) (2006) 1751–1773.

[21] R. Paget, I. Longstaff, Texture synthesis via a noncausal nonparametric multiscale Markov random field, IEEE Trans. Image Process. 7 (6) (1998) 925–931.

[22] A.A. Efros, T.K. Leung, Texture synthesis by non-parametric sampling. Proc. Seventh Intl. Conf. Computer Vision (ICCV), Vol. 2, Kerkyra, Greece, 1999, pp. 1033–1038.

[23] V. Kwatra, A. Schödl, I. Essa, G. Turk, A. Bobick, Graphcut textures: image and video synthesis using graph cuts, ACM Trans. Graphics SIGGRAPH 22 (3) (2003) 277–286.

[24] M.E. Gheche, J.-F. Aujol, Y. Berthoumieu, C.A. Deledalle, Texture reconstruction guided by a high-resolution patch, IEEE Trans. Image Process. 26 (2) (2017) 549–560.

[25] X. You, W. Guo, S. Yu, K. Li, J.C. Príncipe, D. Tao, Kernel learning for dynamic texture synthesis, IEEE Trans. Image Process. 25 (10) (2016) 4782–4795.

[26] L.A. Gatys, A.S. Ecker, M. Bethge, Texture synthesis using convolutional neural networks. Proceedings of the 28th International Conference on Neural Information Processing Systems Vol. 1, MIT Press, Cambridge, MA, USA, 2015, pp. 262–270.

[27] Z.-M. Wang, M.-H. Li, G.S. Xia, Conditional generative convnets for exemplar-based texture synthesis, IEEE Trans. Image Process. 30 (2021) 2461–2475.

[28] D. Ulyanov, V. Lebedev, A. Vedaldi, V.S. Lempitsky, Texture networks: feed-forward synthesis of textures and stylized images. Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19–24, 2016, Vol. 48 of JMLR Workshop and Conference Proceedings, JMLR.org, 2016, pp. 1349–1357.

[29] D. Ulyanov, A. Vedaldi, V. Lempitsky, Improved texture networks: maximizing quality and diversity in feed-forward stylization and texture synthesis. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 4105–4113.

[30] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, M.H. Yang, Diversified texture synthesis with feed-forward networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 266–274.

[31] C. Rodriguez-Pardo, S. Suja, D. Pascual, J. Lopez-Moreno, E. Garces, Automatic extraction and synthesis of regular repeatable patterns, Comput. Graphics 83 (2019) 33–41.

[32] J. Johnson, A. Alahi, L. Fei-Fei, Perceptual losses for real-time style transfer and super-resolution, in: B. Leibe, J. Matas, N. Sebe, M. Welling (Eds.), Computer Vision – ECCV 2016, Springer International Publishing, 2016, pp. 694–711.

[33] Z. Chen, H. Zhang, Learning implicit fields for generative shape modeling. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 5932–5941.

[34] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, A. Geiger, Occupancy networks: learning 3D reconstruction in function space. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 4455–4465.

[35] J.J. Park, P. Florence, J. Straub, R. Newcombe, S. Lovegrove, DeepSDF: learning continuous signed distance functions for shape representation. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 165–174.

[36] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, G. Wetzstein, Implicit neural representations with periodic activation functions. Advances in Neural Information Processing Systems Vol. 33, Curran Associates, Inc., 2020, pp. 7462–7473.

[37] P. Henzler, N.J. Mitra, T. Ritschel, Learning a neural 3D texture space from 2D exemplars. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 8353–8361.

[38] M. Oechsle, L. Mescheder, M. Niemeyer, T. Strauss, A. Geiger, Texture fields: learning texture representations in function space. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 4530–4539.

[39] T. Portenier, S.A. Bigdeli, O. Goksel, GramGAN: deep 3D texture synthesis from 2D exemplars, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems Vol. 33, Curran Associates, Inc., 2020, pp. 6994–7004.

[40] I. Anokhin, K. Demochkin, T. Khakhulin, G. Sterkin, V. Lempitsky, D. Korzhenkov, Image generators with conditionally-independent pixel synthesis. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 14273–14282.

[41] I. Skorokhodov, S. Ignatyev, M. Elhoseiny, Adversarial generation of continuous images. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 10748–10759.

[42] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets. Proceedings of the 27th International Conference on Neural Information Processing Systems Vol. 2, MIT Press, Cambridge, MA, USA, 2014, pp. 2672–2680.

[43] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein generative adversarial networks. Proceedings of the 34th International Conference on Machine Learning, Vol. 70 of Proceedings of Machine Learning Research, PMLR, 2017, pp. 214–223.

[44] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A. Courville, Improved training of Wasserstein GANs. Proceedings of the 31st International Conference on Neural Information Processing Systems, Curran Associates Inc., Red Hook, NY, USA, 2017, pp. 5769–5779.

[45] T. Miyato, T. Kataoka, M. Koyama, Y. Yoshida, Spectral normalization for generative adversarial networks. 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30, - May 3, 2018, Conference Track Proceedings, 2018.

[46] N. Jetchev, U. Bergmann, R. Vollgraf, Texture synthesis with spatial generative adversarial networks. NeurIPS Workshop on Adversarial Training, 2016.

[47] U. Bergmann, N. Jetchev, R. Vollgraf, Learning texture manifolds with the periodic spatial GAN. Proceedings of the 34th International Conference on Machine Learning, Vol. 70 of Proceedings of Machine Learning Research, PMLR, 2017, pp. 469–477.

[48] D. Zhang, A. Khoreva, Progressive augmentation of GANs. Proceedings of the 33rd International Conference on Neural Information Processing Systems, Curran Associates Inc., Red Hook, NY, USA, 2019, pp. 6249–6259.

[49] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, T. Aila, Analyzing and improving the image quality of StyleGAN. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 8107–8116.

[50] R. Abdal, Y. Qin, P. Wonka, Image2StyleGAN: how to embed images into the StyleGAN latent space?. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 4431–4440.

[51] Y. Shen, J. Gu, X. Tang, B. Zhou, Interpreting the latent space of GANS for semantic face editing. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 9240–9249.
[52] R. Xu, X. Wang, K. Chen, B. Zhou, C.C. Loy, Positional encoding as spatial inductive bias in GANs. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 13564–13573.
[53] J. Choi, J. Lee, Y. Jeong, S. Yoon, Toward spatially unbiased generative models. 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 14233–14242.
[54] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need. Proceedings of the 31st International Conference on Neural Information Processing Systems, Curran Associates Inc., Red Hook, NY, USA, 2017, pp. 6000–6010.
[55] J. Gehring, M. Auli, D. Grangier, D. Yarats, Y.N. Dauphin, Convolutional sequence to sequence learning. Proceedings of the 34th International Conference on Machine Learning, Vol. 70 of Proceedings of Machine Learning Research, PMLR, 2017, pp. 1243–1252.
[56] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An image is worth 16x16 words: Transformers for image recognition at scale. 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3–7, 2021, 2021.
[57] B. Julesz, Binocular depth perception of computer-generated patterns, Bell Syst. Tech. J. 39 (1960) 1125–1162.
[58] T. Leung, J. Malik, Representing and recognizing the visual appearance of materials using three-dimensional textons, Int. J. Comput. Vis. 43 (1) (2001) 29–44.
[59] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, A. Vedaldi, Describing textures in the wild. 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 3606–3613.
[60] G.J. Burghouts, J.M. Geusebroek, Material-specific adaptation of color invariant features, Pattern Recognit. Lett. 30 (3) (2009) 306–313.
[61] L. Sharan, R. Rosenholtz, E.H. Adelson, Accuracy and speed of material categorization in real-world images, J. Vis. 14 (9) (2014) 12.
[62] S. Bell, P. Upchurch, N. Snavely, K. Bala, Material recognition in the wild with the materials in context database. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3479–3487.
[63] J. Xue, H. Zhang, K. Nishino, K.J. Dana, Differential viewpoints for ground terrain material recognition, IEEE Trans. Pattern Anal. Mach. Intell. 44 (3) (2022) 1205–1218.
[64] K.J. Dana, B. van Ginneken, S.K. Nayar, J.J. Koenderink, Reflectance and texture of real-world surfaces, ACM Trans. Graphics 18 (1) (1999) 1–34.
[65] K.J. Dana, B. Van Ginneken, S.K. Nayar, J.J. Koenderink, CUReT: Columbia-Utrecht reflectance and texture database, www1.cs.columbia.edu/CAVE/software/curet/.
[66] M. Fritz, E. Hayman, B. Caputo, J.O. Eklundh, The KTH-TIPS database (2004).
[67] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, S. Hochreiter, GANs trained by a two time-scale update rule converge to a local Nash equilibrium. Proceedings of the 31st International Conference on Neural Information Processing Systems, Curran Associates Inc., 2017, pp. 6629–6640.
[68] T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, T. Aila, Alias-free generative adversarial networks. Advances in Neural Information Processing Systems Vol. 34, Curran Associates, Inc., 2021, pp. 852–863.
[69] J. Lin, G. Sharma, T. Pappas, Source code for: toward universal texture synthesis by combining texton broadcasting with noise injection in StyleGAN-2(2022). https://github.com/JueLin/textureSynthesis-stylegan2-pytorch.

Jue Lin is currently a PhD candidate in Department of Electrical and Computer Engineering at Northwestern University, Evanston, IL, USA. He received his bachelor's degree of Engineering from Sun Yatsen University, China and bachelor's degree of Engineering from Hong Kong Polytechnic University with First Class Honour. His current research interests are Texture Analysis and Synthesis, Image and Multimedia Signal Processing, Perceptual Models for Image Analysis, Computer vision and Deep Learning.

Gaurav Sharma received the BE degree in electronics and communication engineering from the Indian Institute of Technology, Roorkee (formerly, University of Roorkee), the master's degree in applied mathematics from North Carolina State University (NCSU), Raleigh, NC, USA, and electrical communication engineering from the Indian Institute of Science, Bengaluru, India, and the PhD degree in electrical and computer engineering from NCSU. From 1996 to 2003, he was with Xerox Research and Technology, Webster, NY, USA, first as a member of research and technology staff and then as a Principal Scientist and a Project Leader. From 2008 to 2010, he was the Director of the Center for Emerging and Innovative Sciences (CEIS), a New York state supported center for promoting joint university-industry research and technology development, which is housed at the University of Rochester. He is currently with the Department of Electrical and Computer Engineering, the Department of Computer Science, and the Department of Biostatistics and Computational Biology, University of Rochester. His research interests include data analytics, signal and image processing, computer vision, color imaging, media security, and communications. He is a fellow of IEEE, of SPIE, and of the Society for Imaging Science and Technology (IS&T). He has served as the Editor-in-Chief (EIC) for the IEEE Transactions on Image Processing, from 2018 to 2020, and the Journal of Electronic Imaging (JEI), from 2011 to 2015.

Thrasyvoulos N. Pappas received the S.B., S.M., and Ph.D. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, MA, in 1979,1982, and 1987, respectively. From 1987 until 1999, he was a Member of the Technical Staff at Bell Laboratories, Murray Hill, NJ. Since 1999 he has been with the Department of Electrical and Computer Engineering at Northwestern University. His research interests are in human perception and electronic media, and in particular, image quality and compression, image analysis, content-based retrieval, model-based halftoning, and tactile and multimodal interfaces. Dr. Pappas is a Life Fellow of the IEEE and Fellow of SPIE and IS&T. He has served as Vice President-Publications (2015–17) and elected member of the Board of Governors (2004–07) of the Signal Processing Society of IEEE, editor-in-chief of the IEEE Transactions on Image Processing (2010–12), chair of the IEEE Image and Multidimensional Signal Processing Technical Committee (2002–03), and technical program co-chair of ICIP-01 and ICIP-09. He has also served as co-chair of the SPIE/IS&T Conference on Human Vision and Electronic Imaging (1997–2018). He is currently editor-in-chief of the IS&T Journal of Perceptual Imaging.