Structured GANs – Supplementary

1. Generating Tiles

As a second application for manipulating the structure of GANs, we present methods for creating tiles that can be arranged repeatedly in 2D in a variety of predetermined patterns. Just like symmetry, tiling enforces a specific structure on the output. For example, in the case of simple tiling, where tiles are being placed in the same orientation on a grid, the top (left) part of the tile should merge smoothly with the bottom (right) part.

1.1. Previous Work On Texture Synthesis

Gatys [3] demonstrated how to capture texture properties from a given image and generate new images with the same texture properties. The descriptor is based on a pre-trained network, usually VGG [8]. A GRAM matrix is extracted from feature maps of certain layers. The objective compares the descriptors of the target image to those of the source image. [2, 9] perform style transfer by combining a content loss from a feature map of a deep layer of VGG. Later on, works such as [5, 10, 6, 4, 1] and others, showed how to train generative networks that are able to simultaneously generate images with texture properties that were already embedded in the train process. Works like [6, 4, 1] do so as a GAN implementation.

In contrast to previous work, we focus on tiles and not on the textured image. This allows us to develop GANS that create tiles for complex tiling patterns.

1.2. An Architecture for Generating Tiles

The idea of enforcing structure by constructing a suitable architecture, as opposed to modifying just the loss, extends beyond symmetry to the problem of tiling. The input to the tiling problem is an image I of some texture. The goal is to synthesize a patch that:

1. Has texture properties that are indistinguishable from those of patches from the source image I.

2. Has a periodic structure such that when the patch is concatenated to itself, there is no texture discontinuity in the boundary.

The most basic tiling pattern repeats each tile, as is, in multiple columns and rows. However, as Fig. 1 illustrates, there are many alternative patterns in which the patterns might be rotated or placed in more complex patterns.

As in symmetry, we employ a modified version of the generator G of the DC-GAN method [7] in order to transform a random vector z into a patch image, in this case of size 64×64. Unlike the symmetry encode case, in which the vector z encodes whether the output image is symmetric or not, for tiling, we expect all outputs to maintain the two desired properties and z is completely random.

Since it is the texture properties of the patch that we are concerned with, we encode the patch using the gram matrix extracted from the generated image as well as from all layers of D, right after the convolution, and before adding the bias, performing batch normalization and applying ReLU. Specifically,

$$GRAM_{ij}^{l} = \left\langle F_i^l, F_j^l \right\rangle,$$

where F_i^l denotes the $i-th$ feature map of layer l. A virtual layer of ones is added in order to capture first order statistics and the size of the GRAM matrix computed for layer l is, therefore, $(k_l+1)^2$, where k_l is the number of filters in this layer. All gram matrices are then normalized by the value $k_l^{1.5}$.

All GRAM fields from all the layers of D are concatenated to one descriptor, which is fed to the fully connected part of D. At each batch, 64 crops out of I of size $64 \times 64 \times 3$ are used as the "real" samples and 64 generated samples of the same size are used as the "fake" sample. The architecture of D for capturing textures is depicted in Fig. 2.

We propose two different tiling GAN methods. The first employs cyclic deconvolutions and the second tiles and crops.

Cyclic deconvolution In order to support horizontal tiling, for example, it is necessary to have the leftmost part of the patch similar to the rightmost part. This is enforced by replacing the deconvolution blocks of G with cyclic deconvolution blocks, in which the convolutions support extend beyond the edges of the feature

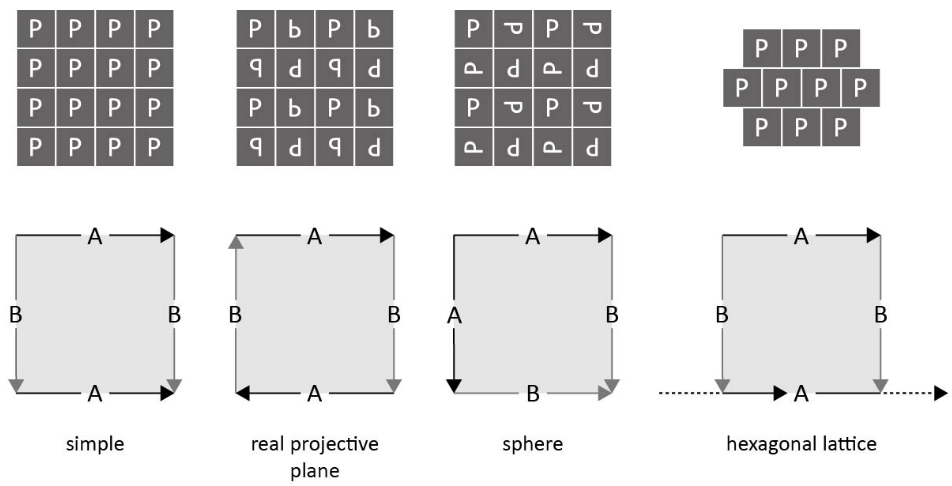

Figure 1: The four forms of tiling presented in this paper. From left to right: (simple rectangular lattice, real projective plane topology, spherical topology, hexagonal lattice)

map and warp back to the other end of the map. This is done for all layers of G. Note, that for complex tiling patterns, the cyclic deconvolution take more complex forms (see below).

Tile and randomly crop In this method, it is the discriminator that enforces the tiling property. This is done by taking the generated image, tiling it in the plane and cropping a $64 \times 64 \times 3$ patch from the result. This patch is then fed to D. If there are tiling artifacts in the crop, the discriminator will then pick up on these. During backpropagation, G is being augmented in a way that reduces the artifacts and learns the tiling pattern implicitly. See Fig. 3.

1.3. Tiling experiments

We first present, in Fig. 4, the results obtained for the simple grid tiling. As can be seen, tiling using tiles generated by the baseline DC-GAN leads to noticeable artifacts at the boundaries of the tiles, while either one of the two methods we propose avoids these artifacts.

We further experimented with less conventional tiling approaches. The results are shown in Fig. 5. The proposed methods perform well, except that the cyclic convolution method is not appropriate for the spherical topology, since it requires the conversion of a row to a column and vice versa.

A closer look at the various artifacts can be observed in Fig. 6.

2. MSE Plot for Symmetric GANs

We measure the MSE between each image and the mirror version of it. The results are shown in Fig. 7. As can be seen, the proposed methods drop to nearly 0 in the middle image, indicating that those images are symmetric to themselves. We can see that the MSE of the other methods is relatively constant and does not drop to zero. The loss-based method, with the strong symmetric constraints creates images that are symmetric throughout the range of z' values. An even stronger symmetry loss would lead to an MSE close to zero along the entire curve, with an image that is barely recognizable as a face.

References

[1] Urs Bergmann, Nikolay Jetchev, and Roland Vollgraf. Learning texture manifolds with the periodic spatial gan. *arXiv preprint arXiv:1705.06566*, 2017.

[2] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.

[3] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. In *NIPS*, 2015.

[4] Nikolay Jetchev, Urs Bergmann, and Roland Vollgraf. Texture synthesis with spatial generative adversarial networks. *arXiv preprint arXiv:1611.08207*, 2016.

[5] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711. Springer, 2016.

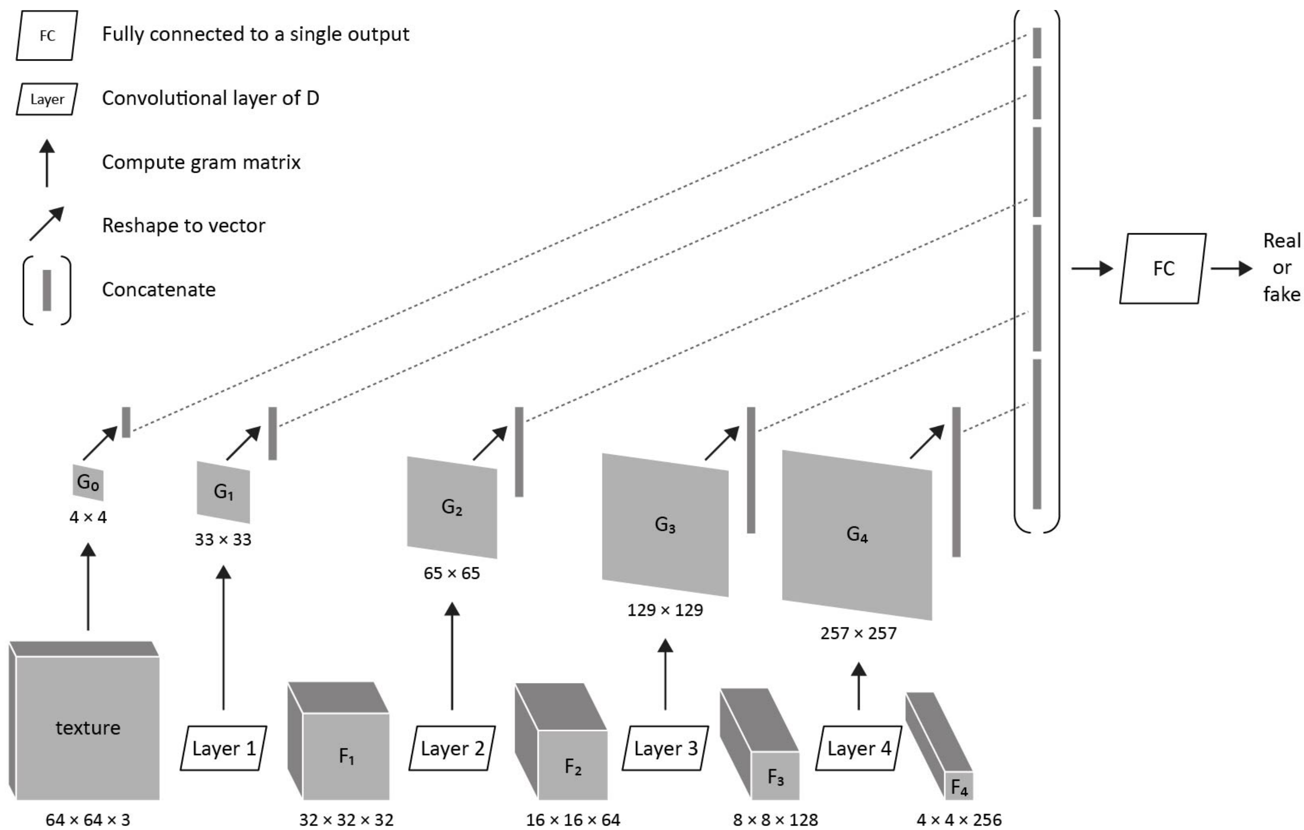

Figure 2: The architecture of D for texture synthesis.

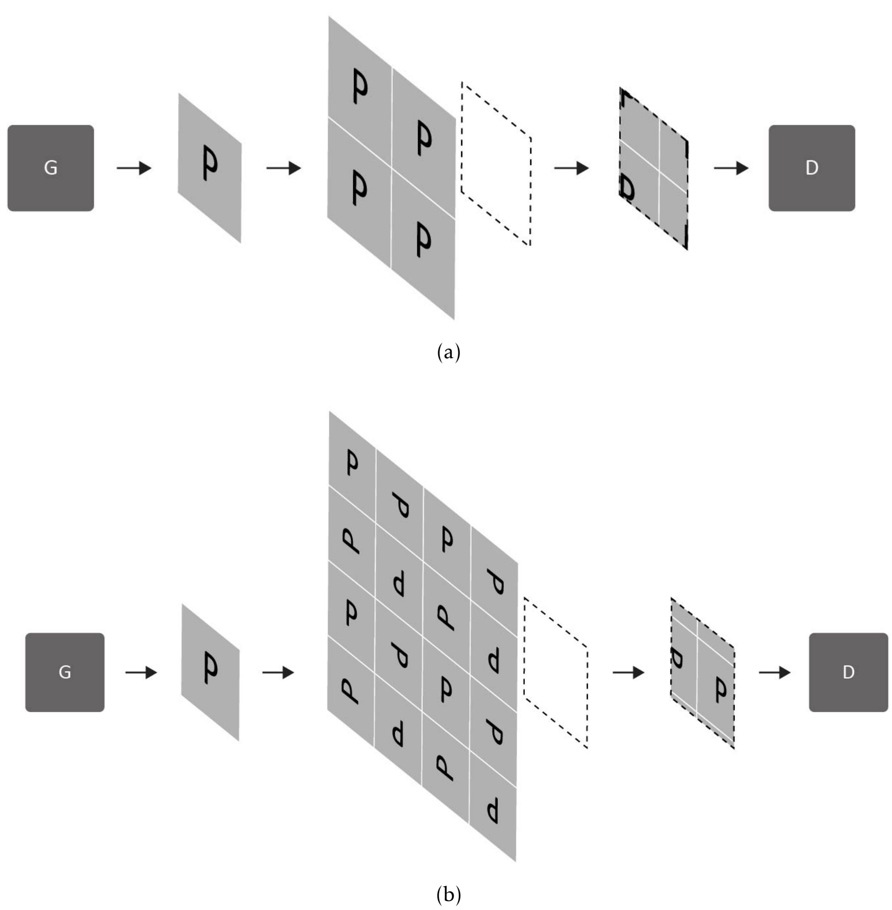

Figure 3: The tile and random crop method as applied to (a) Vanilla tiling on a grid. (b) tiling the spherical topology.

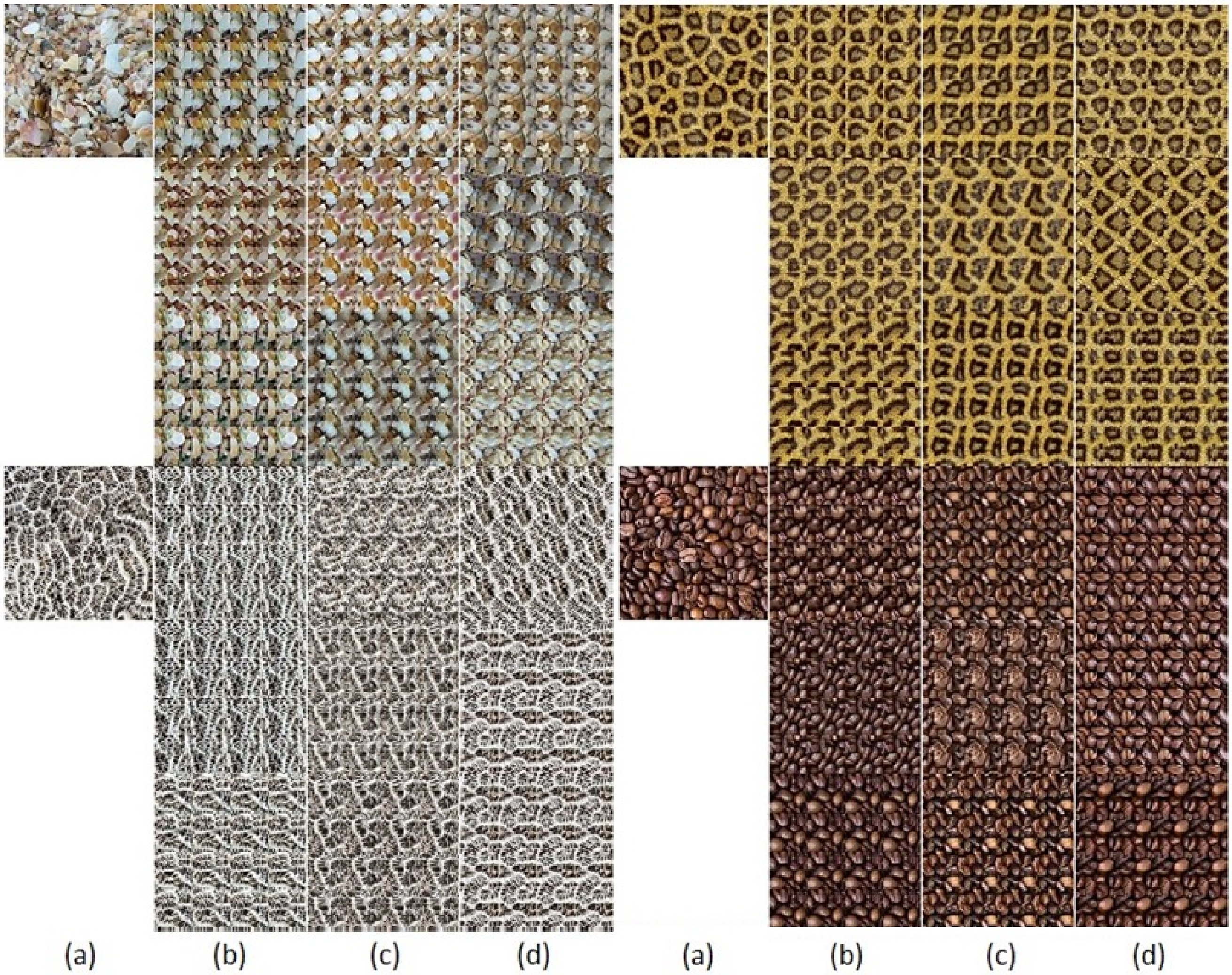

Figure 4: A comparison of the tiling methods. (a) real texture. (b) tiling outputs of DC-GAN. (c) the outcome of tiling with the cyclic deconvolution method. (d) the outcome of tiling with the tile and crop method.

[6] Chuan Li and Michael Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *European Conference on Computer Vision*, pages 702–716. Springer, 2016.

[7] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint*, 1511.06434, 2015.

[8] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[9] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *CVPR*, 2017.

[10] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. In *Int. Conf. on Machine Learning (ICML)*, 2016.

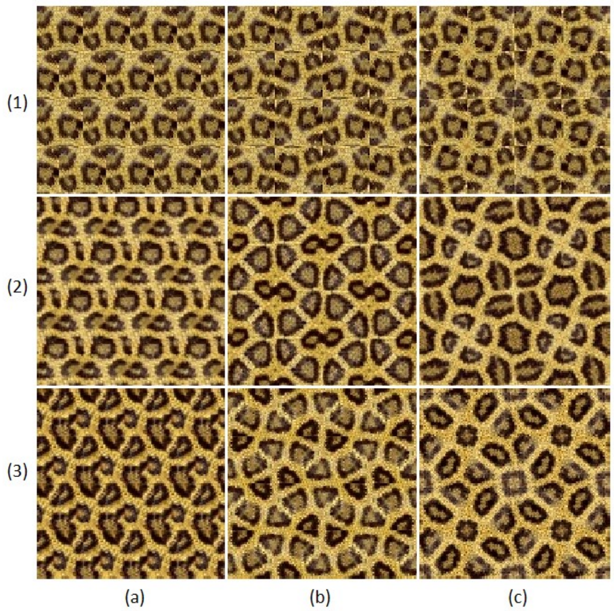

Figure 5: Row 1 shows tiling of a real texture. Row 2 shows tiling using the circular convolution method. Row 3 shows tiling of using random crop method. Column a shows tiling in a hexagonal pattern. Column b shows tiling in pattern of real projective plane topology. Column c shows tiling in pattern of spherical topology.

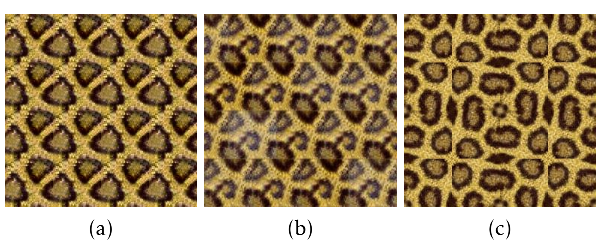

Figure 6: a collection of three repeatable artifacts observed during tiling experiments. (a) a noise texture that appears in some cases of tiling using the random crop method. (b) hexagonal tiling with the tile and crop method results in a constant tile, here each tile has a different z and yet all tiles are the same. (c) a discontinuity phenomenon typical for cyclic deconvolution combined with spherical topology.

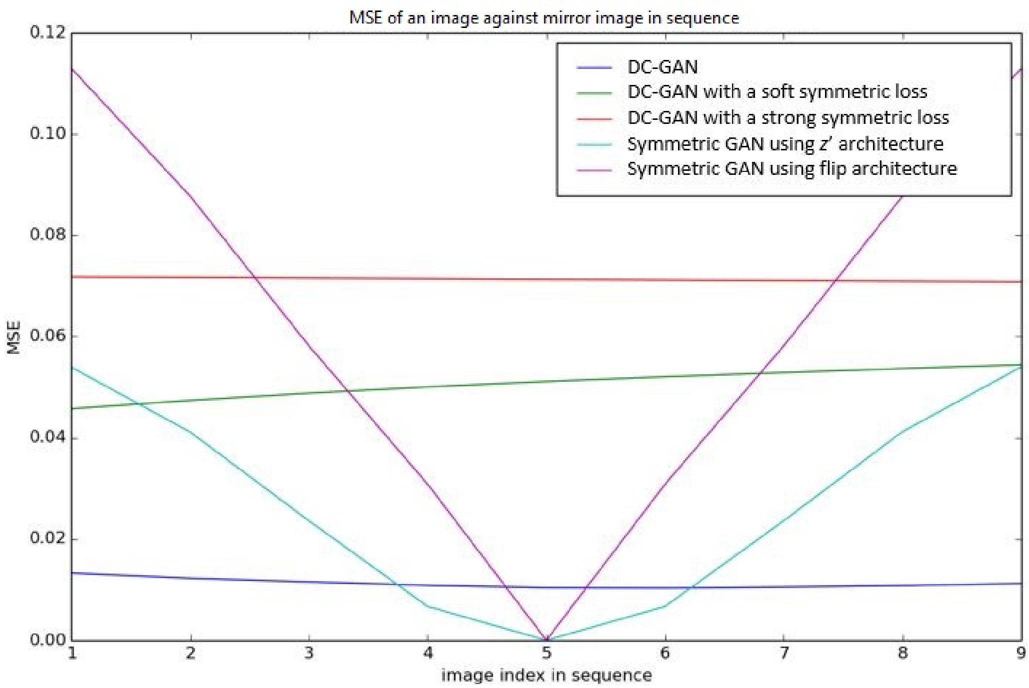

Figure 7: The MSE difference between a generated image $G(z)$ and the mirrored image of $G(z_N)$, where z_N is the vector that is supposed to generate the mirrored image, i.e., for the loss based method and the z' symmetric GAN, the first five coordinates of z are the negative of the first five coordinates of z'.