جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology

Anna Frühstück, Ibraheem Alhashim, Peter Wonka (KAUST)

TileGAN: Synthesis of Large-Scale Non-Homogeneous Textures

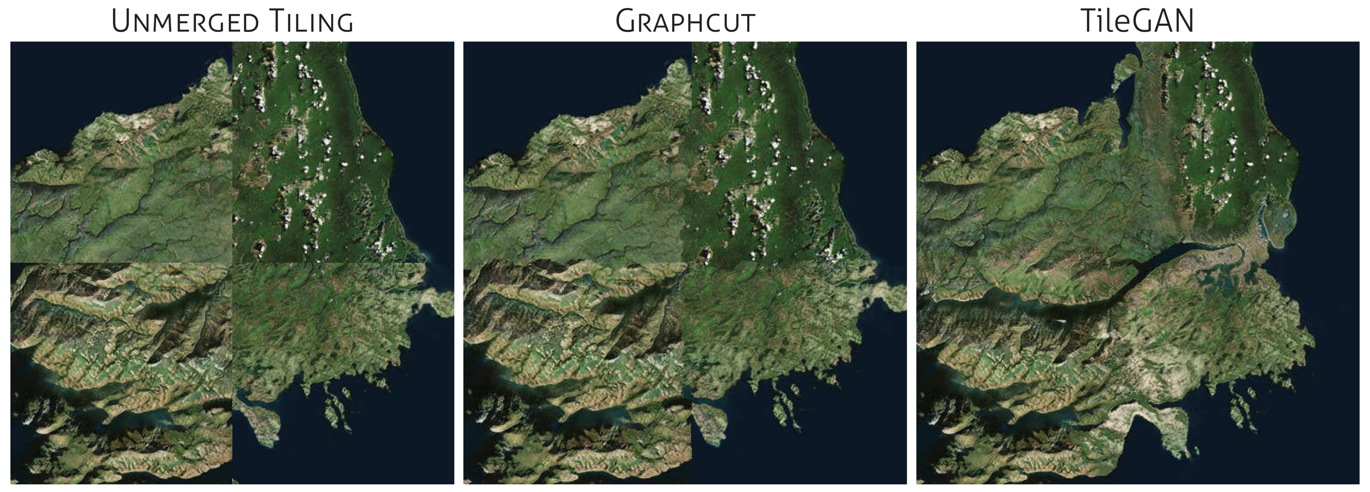
thrive SIGGRAPH2019 LOS ANGELES • 28 JULY – 1 AUGUST

Problem

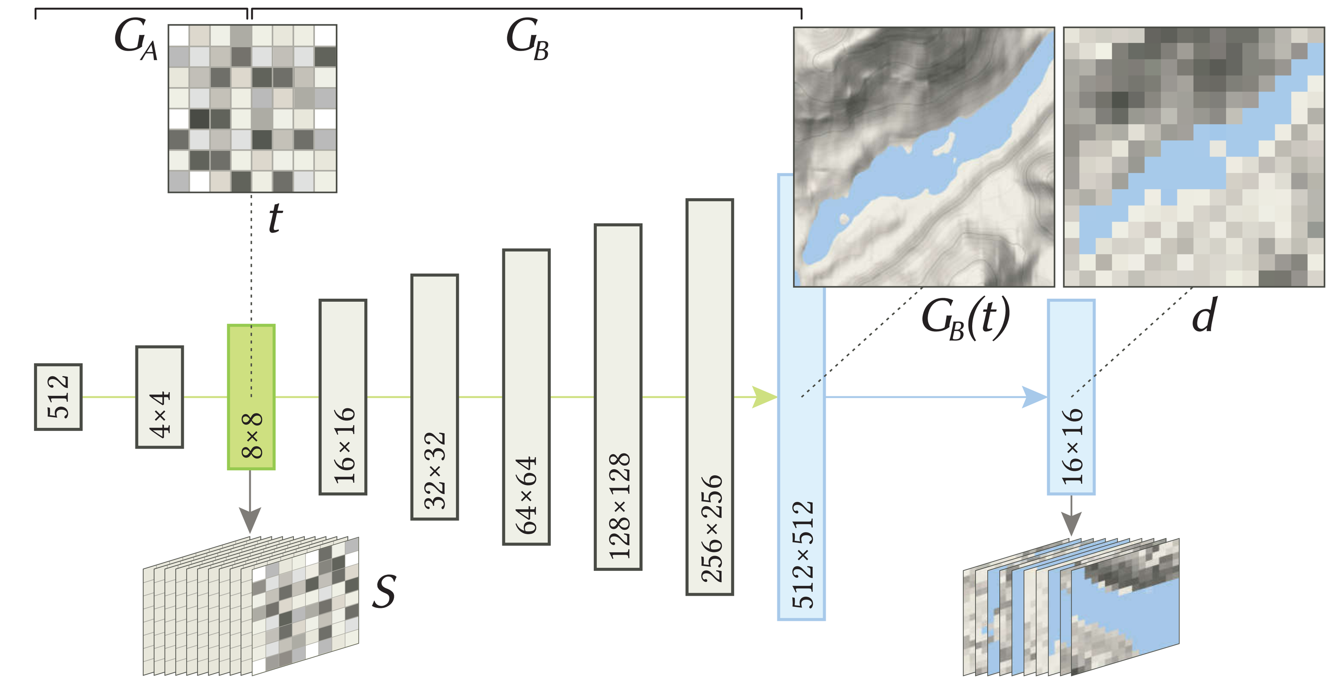
Textures are important for many computer graphics applications. With increasingly powerful graphics hardware and screen sizes, the need for high-resolution textures is ubiquitous. Despite these developments, large-scale texture synthesis remains extremely challenging due to the challenge of affording variability and generating features at multiple scales. Techniques that are able to generate reasonable results on small scales often exhibit artifacts and repetitions when scaling them to large output sizes.

Recently, Generative Adversarial Networks (GANs) [Goodfellow2014] have been successfully applied to the task of synthesizing plausible images of rich detail, but the outputs are typically limited in size.

A commonly used strategy for generating large-scale outputs is tiling smaller textures, but it is challenging to achieve a tiling with seamless transitions between adjacent tiles, even when blending the edges, such as with GraphCut [Kwatra2003].

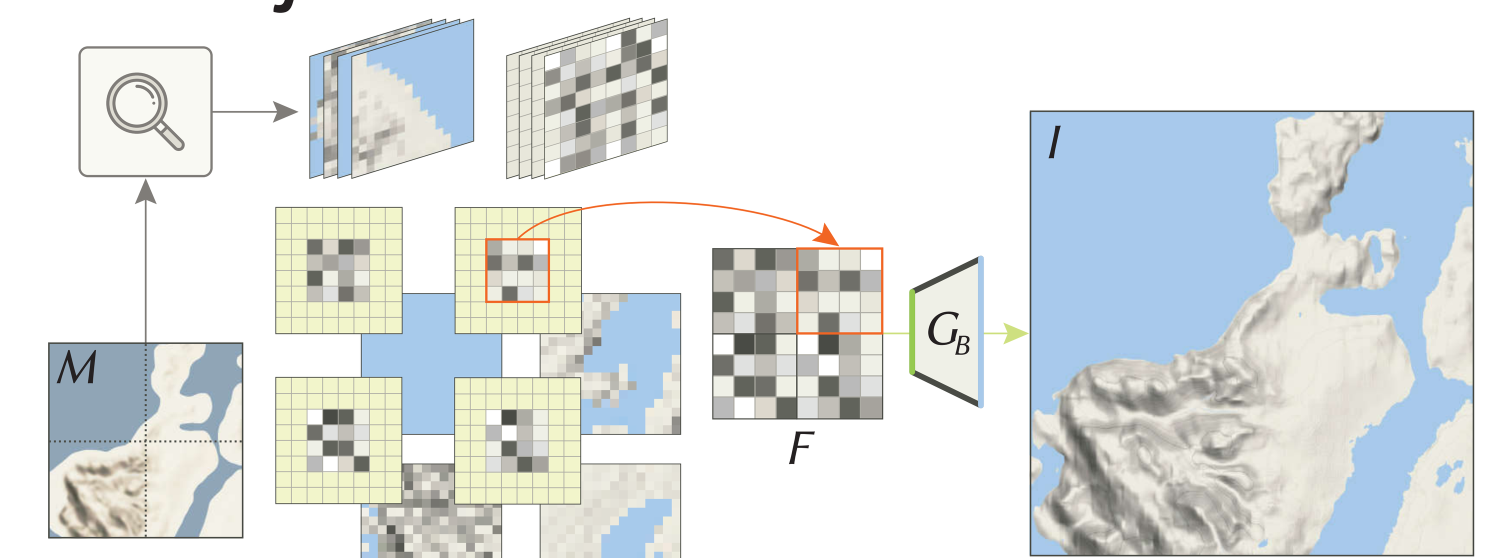


Generator Architecture



The generator network of a standard ProGAN [Karras2018] is split into two stages G_A and G_B. The intermediate latents t are extracted, and can then be tiled and manipulated. These latent tiles are stored along with a downsampled version d of the output in the lookup database S.

Guided Synthesis



A guidance image M can be used to guide the texture synthesis. M is split into blocks and tiles that are expected to generate similar output are queried for each block from the database S. The selected latents can be cropped to control the size and appearance of the generated result. The latents are tiled to generate the latent field F. Finally, F is processed by G_B to produce the final output I. The coherence of the output can be improved by applying a Markov Random Field (MRF) formulation to the field F.

Our Method at a Glance

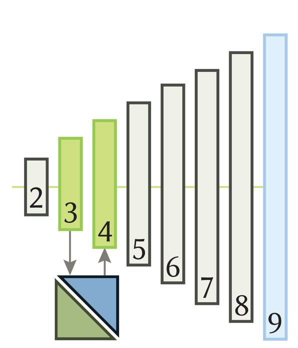
We synthesize textures with interesting features at very large-scale by exploiting the convolutional nature of GANs. We extract latents from the trained generator network, arrange the latents in a tiled grid, and feed the grid back into the network. The resulting output texture features smoothly blended transitions between adjacent texture tiles. Our technique can generate high-resolution textures of arbitrary output size.

Training

We train our generator network on texture data using Progressive Growing of GANs (ProGAN). We have curated a variety of datasets of 20K — 60K high-quality 512×512px texture tiles from publicly available resources such as aerial images, telescope imagery, maps and high-resolution scans of famous artworks.

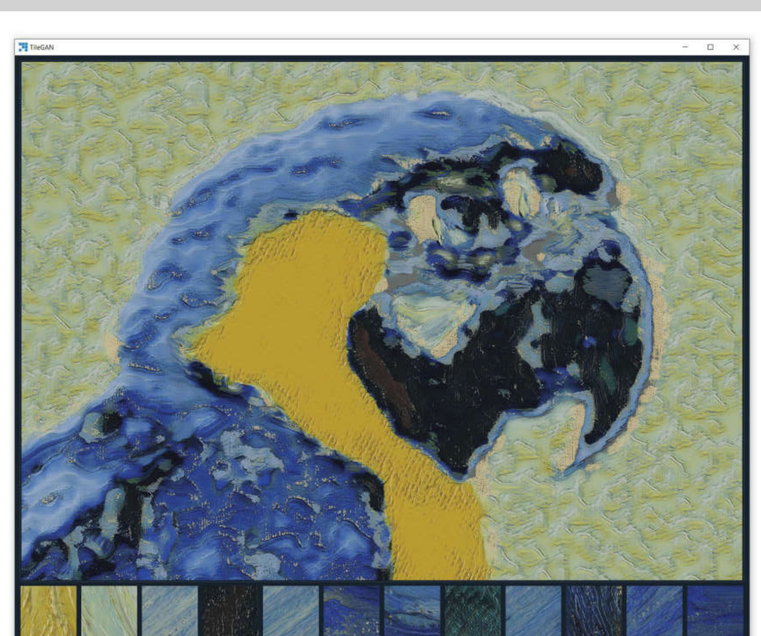
Merging Latents

The trade-off between the amount of permitted change for merging vs. the preservation of the original features can be controlled by selecting the level at which the generator network is split and the latents are tiled. In the below example, the effect of the merge level is demonstrated along a diagonal split between two very different tiles.



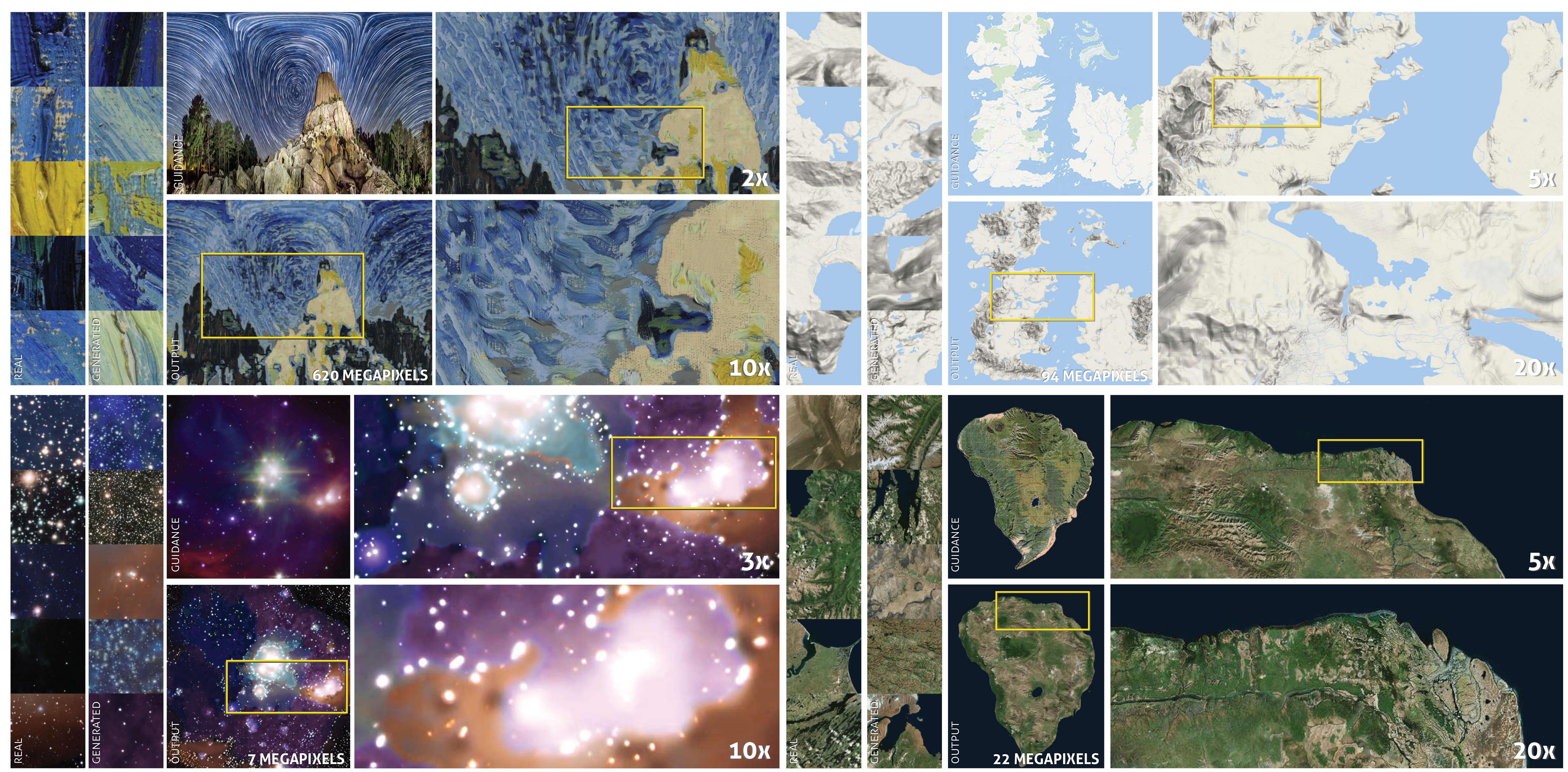
Artistic Control

We demonstrate our approach in our application where users can generate and interactively edit textures based on guidance images or randomized input fields. The underlying latent field of the texture can be modified by sampling new latents via drag-and-drop from clusters from the latent database. We also allow for other editing operations, such as cloning, perturbing and morphing between two latents.



Results

The flexibility of our technique facilitates the synthesis of textures of very large size containing a wide range of features at multiple levels of detail. We have generated results of up to 1.5 Gigapixels. We show a variety of results of our algorithm below. Each result shows a selection of real texture tiles (1st column), synthesized output tiles (2nd column), the guidance image and the high-resolution output of our algorithm (3rd column) and finally zoomed-in details of the texture (4th column).



[Kwatra2003] Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk, and Aaron Bobick. 2003. **Graphcut Textures: Image and Video Synthesis Using Graph Cuts.** *ACM Trans. Graph. 22.*
[Goodfellow2014] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. **Generative Adversarial Nets.** In *Advances in Neural Information Processing Systems 27.*
[Karras2018] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. 2018. **Progressive Growing of GANs for Improved Quality, Stability, and Variation.** In *International Conference on Learning Representations.*

The presentation of our technical paper is on Tuesday, July 31, session "High Performance Rendering" (Room 153) · anna.fruehstueck@kaust.edu.sa · github.com/afruehstueck/TileGAN