DEEP TERRAIN EXPANSION: TERRAIN TEXTURE SYNTHESIS WITH DEEP LEARNING

V.Toulatzis, I.Fudos
Department of Computer Science & Engineering, University of Ioannina
vtoulatz@cs.uoi.gr, fudos@cs.uoi.gr

Introduction

In real-world applications terrains play a cardinal role in the field of games and geospatial applications such as Geographic Information Systems (GIS). The textures of a terrain are essential for creating virtual photorealistic environments for users. In many cases, the entire texture of a region is not available in high resolution or is much smaller than the required texture to cover a terrain. Tiling of a texture across a terrain or an enlarged version of it (Fig.1) usually fails to provide an acceptable visual result.

Figure 1: Left: Original texture. Middle: Tiling. Right: Resizing.

Consequently, high quality texture synthesis is a central issue to such settings. We propose a novel methodology that extends previous work providing both synthesis and expansion/shrinkage of a texture.

Deep Expansion

Given an example texture Gatys et al. [1] used a deep learning process to generate an image matching the features of an input texture. We introduce a novel method that is inspired by the aforementioned approach but it aims on producing synthesized textures of different sizes.

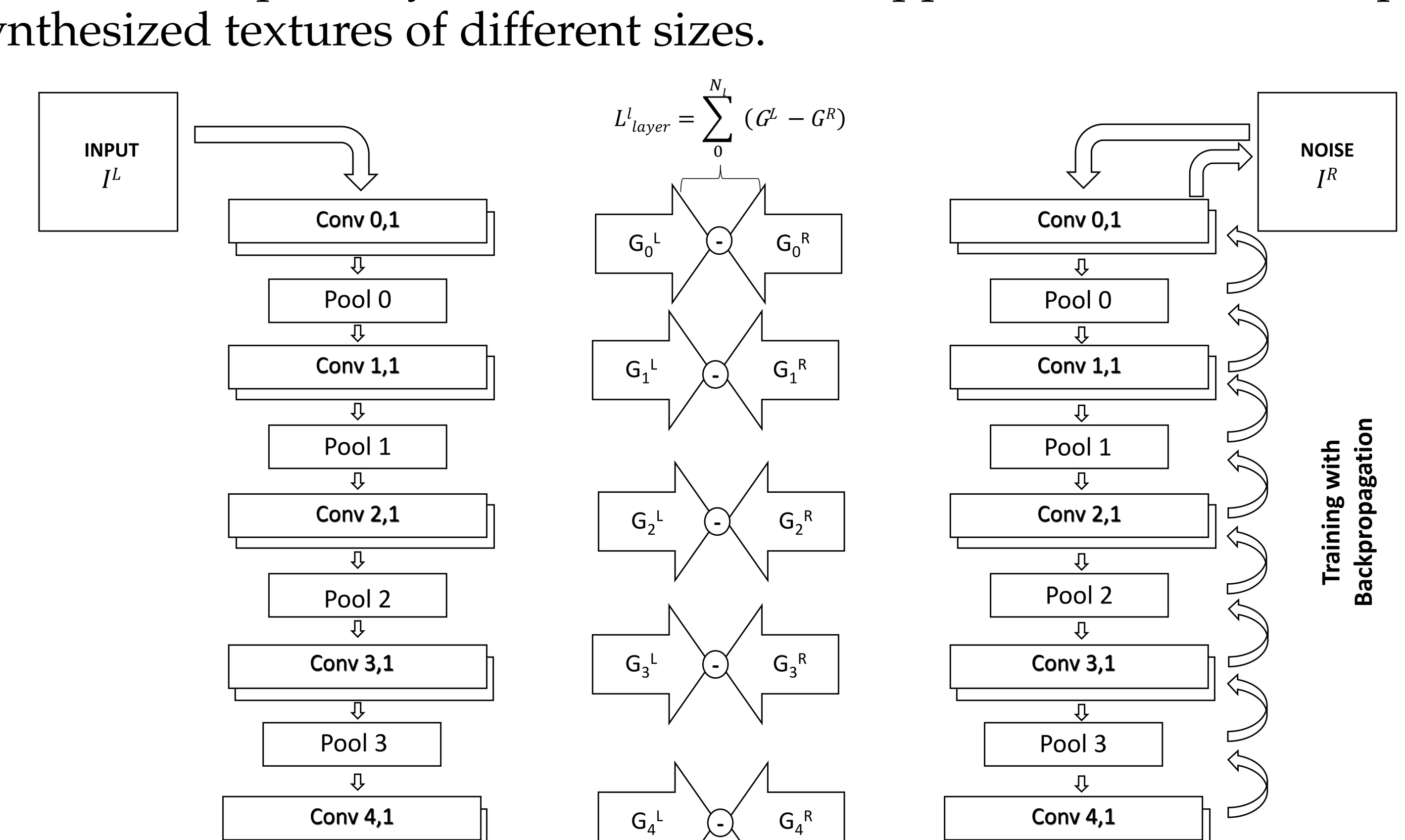

Figure 2: Texture Synthesis work-flow.

Method:

- We utilize a VGG-19 [3] pretrained network for our two separate CNN instances and gradient descent optimization to train our model to finally generate a larger or a smaller resolution terrain texture.
- Every layer l has N_f filters each of vectorized size S_f, where a computation of a feature representation of the two textures occurs by using Gram Matrices:

$$G^l_{ij} = \sum_k F^l_{ik} F^l_{jk}. \tag{1}$$

The correlations among the activations F^l_{ab} (the activation of the a^{th} filter at position b in layer l) of a general feature map matrix $F^l \in R^{N_f \times S_f}$ can be represented by a Gram.

- Then the layer loss that contributes to the total loss function is computed as mean squared displacement of the network Gramians:

$$L^l_{layer} = \frac{\sum_{ij}(G^L_{l_{ij}} - G^R_{l_{ij}})^2}{4{N_f}^2{S_f}^2}. \tag{2}$$

- The total loss function with a weight value w_l for each different layer that our network system aims to minimize is:

$$L_{total}(I^L, I^R) = \sum_{l=0}^{N_L} w_l L^l_{layer}. \tag{3}$$

Results

- Our method have been developed in Python with Caffe [2] using CUDA.
- Our experiments have been carried out on a workstation equipped with a NVIDIA GTX 1050 Ti GPU with 4GB memory.
- We have trained our model with 256×256 images to generate the new 512×512 ones for expansion and the opposite dimensions for shrinkage. For convergence of our model to the results displayed in Fig.3 & Fig.4 the training period was 2000 iterations for each input case.
- The average execution time was $\approx$ 5000 secs ($\approx$ 1 hour and 20min).

Figure 3: Synthesis & expansion of terrain textures

Figure 4: Synthesizing lower resolution textures

Conclusions

- We have introduced a novel method for texture expansion through a synthesizing process whose core is a fundamental deep learning method.
- Our approach provides larger resolution terrain textures of high quality, a property that makes them appropriate for texture mapping.
- We have additionally demonstrated that our approach is capable of creating textures of smaller resolution.

Future Work

- Compare the quality and time performance with other state-of-the-art texture expanding methods (e.g [4]).
- Extend this work to style transfer and prove through experiments the importance of smaller terrain texture synthesis in such cases.
- Our method can be extended for creating new synthesized tiles that match surrounding tiles for a wide spectrum of applications.

References

[1] L. A. Gatys, A. S. Ecker, and M. Bethge. Texture synthesis using convolutional neural networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pages 262–270, Cambridge, MA, USA, 2015. MIT Press.

[2] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[3] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv 1409.1556*, 09 2014.

[4] Y. Zhou, Z. Zhu, X. Bai, D. Lischinski, D. Cohen-Or, and H. Huang. Non-stationary texture synthesis by adversarial expansion. *ACM Trans. Graph.*, 37(4):49:1–49:13, July 2018.