EG UK Computer Graphics & Visual Computing (2019)
G. K. L. Tam and J. C. Roberts (Editors)

Poster

Deep Terrain Expansion: Terrain Texture Synthesis with Deep Learning

V. Toulatzis and and I. Fudos

Department of Computer Science & Engineering, University of Ioannina, Greece

Abstract

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 using an enlarged version of it usually fails to provide an acceptable photorealistic result. Consequently, high quality texture synthesis is a central issue in such settings. In this paper, we explore a novel methodology that extends previous work providing both synthesis and expansion/shrinkage of a texture.

1. Introduction

The general objective of texture synthesis is to generate new synthesized textures. Simple methods of texturing such as tiling and resizing of the initial texture usually fail to cover a surface with visually acceptable results. This is illustrated in Fig.1, in which we observe that tiling causes discontinuity problems and resizing/scaling enhances aliasing. Example-based texture synthesis has introduced

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

significant improvements in this field of research by taking advantage of feature space mapping across the layers of neural networks.

In this paper, we propose an alternative to [GEB15] in which a larger texture is generated by using the Gram Matrix representations of features in every layer of a Convolutional Neural Network (CNN). To this end, we have observed that Gram Matrices are not filter-size dependent but their size is determined solely by the network structure. Consequently, we are able to use two pretrained CNNs that attempt to converge to an acceptable resemblance level of two forwarded images that are not of equal size.

2. Related Work

There are many approaches to synthesizing textures, from simple ones such as tiling and stochastic generators to the more sophisticated example-based and non-parametric. The latter methods comprise pixel-based approaches [EL99], [WL00], patch-based techniques [EF01], [KSE*03] and optimization-based methods [KEBK05]. The most remarkable results have been accomplished by using deep learning techniques such as [GEB15] which is a fundamental work that many recent approaches have relied on. However, in real world applications there is an additional need for creating larger resolution textures. To this end, efficient expanding techniques such as [ZZB*18] and the self-tuning [KNL*15] have been developed.

3. Deep Expansion

Given an example texture [GEB15] used a deep learning process to generate an image matching features of the input. We introduce a novel method that is inspired by the aforementioned approach but it aims at producing synthesized textures of different sizes.

We have employed a VGG-19 [SZ14] pretrained network for our two separate CNN instances and gradient descent optimization to train our model. The two networks are fed with an example terrain texture and a noise image, respectively. In every layer a computation of a feature representation of the two textures occurs by using Gram Matrices (or Gramians). Each layer l has N_f filters each of vectorized size S_f. The correlations among the activations F^l_{ij} of a general feature map matrix $F^l \in R^{N_f \times S_f}$ can be represented by a Gramian $G^l_{ij} = \sum_k F^l_{ik} F^l_{jk}$. The layer loss that contributes to the total loss function with a significance value w_l is computed as mean squared displacement of the network Gramians and the total loss function is defined as $L_{total}(I^L, I^R) = \sum_{l=0}^{N_L} w_l \frac{\sum_{ij}(G^L_{l_{ij}} - G^R_{l_{ij}})^2}{4{N_f}^2{S_f}^2}$.

Our model advances the work of [GEB15] by exploiting the key observation that a Gramian of every layer is only dependent on the

DOI: 10.2312/cgvc.20191262

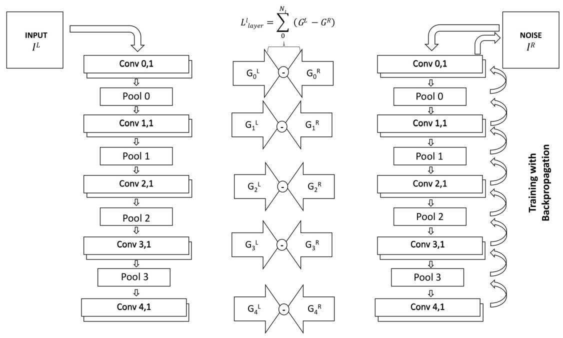

Figure 2: *Texture Synthesis work-flow.*

number of filters and not on the input size. Therefore, we utilize CNNs with identical structure and different filter sizes as shown in Fig.2. By doing so, our approach is able to handle two different resolution textures as input. As a consequence, our model extends the previous work and can generate either smaller or larger synthesized terrain textures.

4. Results & Future Work

We have developed our method in Caffe framework [JSD*14]. Our experiments have been run on a PC equipped with a NVIDIA GTX 1050 Ti GPU with 4GB memory. We train our model with 256×256 images to generate the new 512×512 ones for 2000 iterations. We used both satellite images and texture images that can be applied to any terrain. The average execution time of our model implementation was 5000 secs ($\approx$ 1 hour and 20 minutes). The results[†] of Fig.3 demonstrate that our model captures and transfers image features efficiently from a smaller feature space to a greater one and vice versa. The uniform expansion of terrain textures demonstrates that our model is capable of producing high-quality terrain textures from just one smaller exemplar.

Nevertheless, our method should be thoroughly examined via extra experiments and can be extended to creating new synthesized tiles with no discontinuities across their placement.

Figure 3: *Generating synthesized textures of different resolutions.*

References

[EF01] Efros A. A., Freeman W. T.: Image quilting for texture synthesis and transfer. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 2001), SIGGRAPH '01, ACM, pp. 341–346. URL: http://doi.acm.org/10.1145/383259.383296, doi:10.1145/383259.383296. 1

[EL99] Efros A. A., Leung T. K.: Texture synthesis by non-parametric sampling. In *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2* (Washington, DC, USA, 1999), ICCV '99, IEEE Computer Society, pp. 1033–. URL: http://dl.acm.org/citation.cfm?id=850924.851569. 1

[GEB15] Gatys L. A., Ecker A. S., Bethge M.: Texture synthesis using convolutional neural networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1* (Cambridge, MA, USA, 2015), NIPS'15, MIT Press, pp. 262–270. URL: http://dl.acm.org/citation.cfm?id=2969239.2969269. 1

[JSD*14] Jia Y., Shelhamer E., Donahue J., Karayev S., Long J., Girshick R., Guadarrama S., Darrell T.: Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093* (2014). 2

[KEBK05] Kwatra V., Essa I., Bobick A., Kwatra N.: Texture optimization for example-based synthesis. *ACM Trans. Graph. 24*, 3 (July 2005), 795–802. URL: http://doi.acm.org/10.1145/1073204.1073263, doi:10.1145/1073204.1073263. 1

[KNL*15] Kaspar A., Neubert B., Lischinski D., Pauly M., Kopf J.: Self tuning texture optimization. *Comput. Graph. Forum 34*, 2 (May 2015), 349–359. URL: http://dx.doi.org/10.1111/cgf.12565, doi:10.1111/cgf.12565. 1

[KSE*03] Kwatra V., Schödl A., Essa I., Turk G., Bobick A.: Graphcut textures: Image and video synthesis using graph cuts. In *ACM SIGGRAPH 2003 Papers* (New York, NY, USA, 2003), SIGGRAPH '03, ACM, pp. 277–286. URL: http://doi.acm.org/10.1145/1201775.882264, doi:10.1145/1201775.882264. 1

[SZ14] Simonyan K., Zisserman A.: Very deep convolutional networks for large-scale image recognition. *arXiv 1409.1556* (09 2014). 1

[WL00] Wei L.-Y., Levoy M.: Fast texture synthesis using tree-structured vector quantization. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 2000), SIGGRAPH '00, ACM Press/Addison-Wesley Publishing Co., pp. 479–488. URL: http://dx.doi.org/10.1145/344779.345009, doi:10.1145/344779.345009. 1

[ZZB*18] Zhou Y., Zhu Z., Bai X., Lischinski D., Cohen-Or D., Huang H.: Non-stationary texture synthesis by adversarial expansion. *ACM Trans. Graph. 37*, 4 (July 2018), 49:1–49:13. URL: http://doi.acm.org/10.1145/3197517.3201285, doi:10.1145/3197517.3201285. 1

† More results in supplementary material, also available here.