

COL 783: Assignment 5

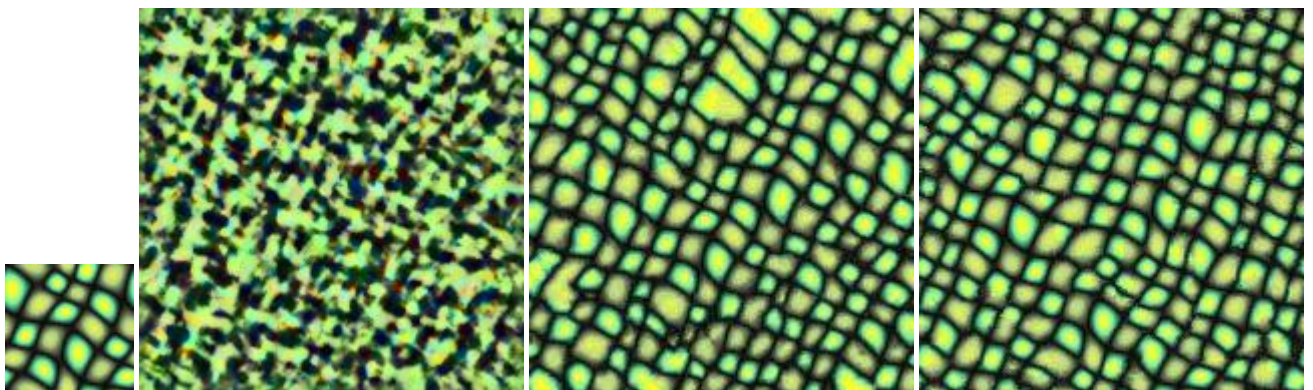
Texture Synthesis

Due date: ~~Wednesday, 20 November 2019~~ Monday, 25 November 2019

OVERVIEW

In class, we have discussed various techniques for texture analysis. In this assignment, we will explore the converse, texture *synthesis*: given an example image of a texture, create a new image that matches its visual appearance and so is perceived by a human viewer as a new sample of the same texture. A nice survey of this area is given in Raad et al., “[A survey of exemplar-based texture synthesis](#)” (2017).

Many early techniques for texture synthesis were built on top of texture analysis: the idea being that if we have extracted a sufficiently rich and comprehensive set of texture descriptors, then any other random image with same descriptors should appear to have similar texture content. These techniques work well for some random textures, but not for many others. Modern texture synthesis techniques in computer graphics are instead based on copying local neighbourhoods of pixels from the exemplar and stitching them together to synthesize a new texture. We will see examples of both techniques in this assignment.



An input image (far left) and textures synthesized from it using the methods of Heeger and Bergen, Efros and Leung, and Wei and Levoy (middle left to far right). From Wei and Levoy (2000).

REQUIREMENTS

Input texture images

I have uploaded a database of simple grayscale textures [here](#). You can get many more examples from the texture synthesis web pages of Alexei Efros: [1](#), [2](#), [3](#), and Michael Ashikhmin: [1](#). (Use the ones on the left side, which are the *input* to their algorithms!)

You should also find some other cool images of your own and run your implementations on them as well.

Random phase noise

The simplest approach is based on Fourier descriptors, i.e. the amplitudes of the Fourier coefficients of the image. Creating a random image with the same descriptors is equivalent to retaining the Fourier amplitudes while randomizing their phases. This is described well in Section 2.1 of the Raad et al. survey linked above. Implement this method and demonstrate some results where it works well and some where it works poorly.

Steerable pyramids

A more sophisticated set of descriptors is given by the steerable pyramid mentioned in class. An explicit construction of a steerable pyramid is given in Section 2.1 of Portilla and Simoncelli’s “[A parametric texture model based on joint statistics of complex wavelet coefficients](#)” (2000). The distributions of values at different subbands of the pyramid can be used as a texture descriptor. Heeger and Bergen’s “[Pyramid-based texture analysis/synthesis](#)” (1995) paper uses this idea for texture synthesis via histogram matching of each subband.

Implement the steerable pyramid as described by Portilla and Simoncelli, and use it for the texture synthesis algorithm of Heeger and Bergen. You may use built-in functions for histogram matching if available. Demonstrate both successful examples and failure cases.

Non-parametric synthesis

Another way to define a texture is as an image such that, at an appropriate scale, every neighbourhood in the image looks like every other neighbourhood. From this point of view, we can perform texture synthesis by incrementally constructing an image such that each neighbourhood in it is also similar to those in the example. Techniques for doing so were introduced by Efros and Leung, “[Texture synthesis by non-parametric sampling](#)” (1999), and Wei and Levoy, “[Fast texture synthesis using tree-structured vector quantization](#)” (2000).

Implement a non-parametric texture synthesis method along the lines of these papers. As suggested by Wei and Levoy, you should perform synthesis at multiple resolutions using the Gaussian pyramid you implemented in Assignment 3. (You are not required to implement *all* components of their algorithm, e.g. vector quantization.)

Extra credit

Implement *one* of the following papers:

- Efros and Freeman, “[Image quilting for texture synthesis and transfer](#)” (2001)
- Some nontrivial subset of Hertzmann et al., “[Image analogies](#)” (2001)

SUBMISSION AND EVALUATION

As usual, submit your code and a report describing your work. Include images demonstrating the results of each task of the assignment. More details TBA.

Students working individually are permitted to implement only *two* out of the three required texture synthesis methods, or to implement all three for extra credit.