Texture Synthesis from Multiple Sources

Li-Yi Wei*
Stanford University

1 Introduction

Recent advances in texture synthesis have made it possible to synthesize textures based on examples. However, most of these algorithms can take only a single texture as the input and output a similar homogeneous texture. This is insufficient for textures that have progressively variant patterns or combined visual characteristics from several different sources.

We present a new algorithm for synthesizing textures from multiple sources. We demonstrate two specializations of our algorithm that are particularly useful: generating texture mixtures from different inputs and synthesizing solid textures from multiple 2D views.

2 Algorithm

Our algorithm is extended from [Wei and Levoy 2000]. In the following, we describe the major extentions of our algorithm over [Wei and Levoy 2000] for multiple-source synthesis.

The input of the algorithm consists of several texture sources. Those sources can be different views of a solid texture, or different patterns for a texture mixture. Each source is associated with a weight image which has the same size as the output. Those weights are user selected parameters specifying how the input sources should be mixed together. In addition, each source is associated with a neighborhood parameter, defining the size and shape of the neighborhood used during the search process for this specific source. This is necessary for texture mixture synthesis where different sources have different texton sizes, or solid texture synthesis where each view has different neighborhoods oriented with the corresponding viewing directions.

In [Wei and Levoy 2000], each output pixel is determined so that the local similarity between the input and output textures is preserved as much as possible. We would like to achieve the same goal for multi-source texture synthesis. However, since we now have more than one input textures, we have to pick the output pixel value that preserves local similarity simultaneously with all the input sources as much as possible.

Mathematically, for each output sample p we would like to find a set of input pixels $\{p_i\}$ so that the following error function is minimized:

$$E(p, \{p_i\}) = \sum_i w_i \times (\|p - p_i\|^2 + \|N_i(p) - N_i(p_i)\|^2) \quad (1)$$

where index i runs through all the input textures, p and each p_i are the output and matching input pixels, and $N_i(p)$, $N_i(p_i)$ are their neighborhoods. Note that we use different neighborhoods N_i for different sources. The error function is computed as a weighted sum of the L_2 norm between $\{p, N_i(p)\}$ and $\{p_i, N_i(p_i)\}$, and the weights $\{w_i\}$ specify the relative importance of the input textures.

To minimize the error function $E(p, \{p_i\})$, we need to determine the values p and $\{p_i\}$ so that the sum on the right hand side of Equation 1 is minimized. When only one source is present, we can directly minimize Equation 1 as follows: simply choose the p_i such that $\|N_i(p) - N_i(p_i)\|^2$ is minimal, and set p to be equal to p_i. This is exactly the algorithm presented in [Wei and Levoy 2000]. However, when multiple sources are present, we cannot solve Equation 1 directly. Instead we use an iterative procedure, alternatively setting the values of $\{p_i\}$ and p while gradually decreasing $E(p, \{p_i\})$. At the beginning of each iteration, we fix the value p and choose $\{p_i\}$ so that each individual error term $\|p - p_i\|^2 + \|N_i(p) - N_i(p_i)\|^2$ is minimized. We then keep $\{p_i\}$ fixed, and set p as the weighted average of $\{p_i\}$. It can be easily proven that both of these steps do not increase $E(p, \{p_i\})$. The process can be stopped when $E(p, \{p_i\})$ remains the same after two consecutive iterations, but experimentally we have found that 1 to 4 iterations are sufficient.

We now describe how to apply our algorithm to synthesize solid textures and texture mixtures. To synthesize texture mixtures, we use the weight images to control how each source effects the mixture result. For example, by assigning equal weighting to all the inputs, the result will be a uniform mixture of the input textures. However, by spatially varying the weight images, we can achieve special effects such as one texture gradually transforming to another (Figure 1). To synthesize solid textures from multiple views, we simply specify several images consisting of different views of a hypothesized solid texture. Each view is associated with a neighborhood oriented with respect to the specific viewing direction. Figure 2 demonstrates several synthesis results.

For further details and results, we refer the reader to [Wei 2002][Chapter 7].

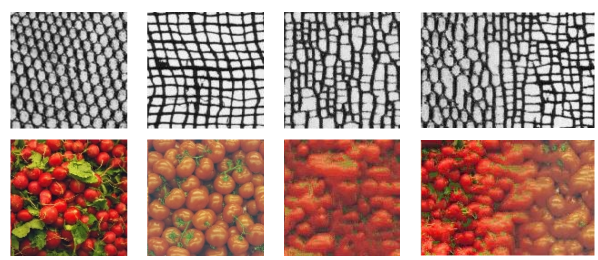

Figure 1: *Texture mixture results. For each row of images, the two input textures are shown on the left, and the corresponding synthesis results, with equal weighting and ramp weighting, are shown on the right.*

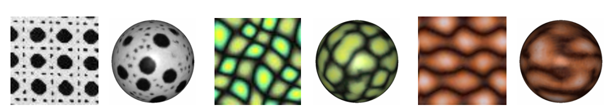

Figure 2: *Solid texture synthesis results. For each pair of images, the original texture is shown on the left, and the corresponding synthesis result, a sphere of solid texture, is shown on the right.*

References

WEI, L.-Y., AND LEVOY, M. 2000. Fast texture synthesis using tree-structured vector quantization. In *Proceedings of ACM SIGGRAPH 2000*, 479–488.

WEI, L.-Y. 2002. *Texture Synthesis by Fixed Neighborhood Searching*. PhD thesis, Stanford University.

*e-mail: liyiwei@graphics.stanford.edu