Creating Texture Exemplars from Unconstrained Images

Yitzchak David Lockerman, Su Xue, Julie Dorsey, and Holly Rushmeier
Department of Computer Science
Yale University
New Haven, Connecticut, USA
Email: yitzchak.lockerman@yale.edu

***Abstract*—Texture is an essential feature in modeling the appearance of objects and is instrumental in making virtual objects appear interesting and/or realistic. Unfortunately, obtaining textures is a labor intensive task requiring parameter tuning for procedural methods or careful photography and post-processing for natural images. Many texture synthesis techniques have been developed to generate textures of arbitrary spatial extent, but these techniques require the user to first produce an exemplar consisting solely of the desired texture. We present a fast method using diffusion manifolds to locate textures in unconstrained photographs, and extract exemplar tiles. The method requires the user to only specify a single point within the image containing the desired texture and the scale of the desired texture. The user may tune the result using simple interactions. The method is non-local, in the sense that the desired texture does not have to appear in a single contiguous region in the source image. This document is a summary of our work and system. A full account is available online [1].**

***Keywords*-Texture Extraction; Image Processing;**

I. Introduction

The ability to model visual textures is essential in any computer graphics modeling system. Yet preparing textures is still a labor intensive task. One approach to allow users to create their own textures has been the development of Markov Random Field (MRF) techniques to generate textures of arbitrary extent from a small patch texture [2]. This patch, called an exemplar, is then used as the input of a texture synthesis algorithm to create an image of the texture with the required size. This process fails when a source image contains details other than the single desired texture. Lu et al. [3] created a system for removing small regions of undesired content within an image dominated by one texture.

In this work we simplify the process of preparing a texture exemplar by providing a method that automatically extracts a desired texture from an image using minimal and intuitive user input. Our technique is based on heat diffusion on a graph, inspired by diffusion manifolds [4]. To our knowledge, this is the first method that can extract non-local texture information from natural images without special preparation.

While our work was originally inspired by Lu et al.'s dominant texture algorithm,it is not directly comparable due to a number of intrinsic differences:

1) By design, the dominant texture algorithm only works on a small class of images, ones that are dominated by a single texture. Our algorithm can handle a much larger class of images, including a large selection of natural images.
2) Their algorithm worked on an offline process, with computation taking on the order of 18 **minutes**. Our algorithm makes use of simple user input, and can process images of comparable size in under 10 **seconds**, using nearly identical,vintage, hardware (and in under 5 **seconds** using current hardware).
3) The dominant texture algorithm worked on images on the order of 11 **thousand** pixels. Our algorithm can work with images with **millions** of pixels. Using vintage hardware, our algorithm takes 38 **seconds** on a 4.9 megapixel image (and under 10 **seconds** on modern hardware).

II. The Texture Exemplar Pipeline

Our technique is used in a system with a simple interface. In the system the user specifies the scale of the texture of interest within the source image. After a short (on the order of seconds) preprocess, the user can generate texture tiles interactively by clicking on a point in the source image. The set of tiles can be refined interactively, if desired, by selecting additional points, rejecting tiles, and adjusting a single parameter. A texture of arbitrary extent can be generated from these tiles using any variety of Markov random field (MRF) techniques. Fig. 1 outlines this pipeline.

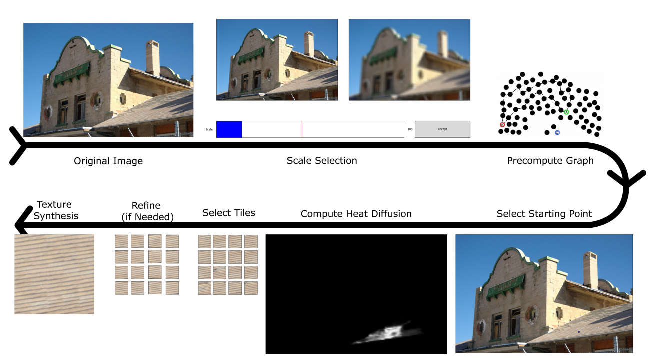

Figure 1. The steps in the pipeline.

A. Defining Texture Scale

A natural image frequently contains multiple textures at many spatial scales. This is an unavoidable problem, as many textures have details that themselves are made up of textures. For example, a brick wall contains a texture that represents the wall pattern and each brick itself has a brick texture. For these situations some sort of user interaction is needed to inform the algorithm which texture scale the user desires.

To solve this problem we present the user with an interface with a slider that adjusts the scale parameter. As the parameter is changed, the user is shown a blurred version of the image. This is done by dividing the image into tiles, then replacing each with its average color. The user is asked to find a scale where the details of the texture are no longer visible. Equivalently, they are asked to identify the level where the desired texture tiles first appear as single color pixels.

B. The Diffusion Graph

Our method is based on performing diffusion on a similarity graph between texture patches. We begin constructing the graph by dividing the image into overlapping tiles and give each a feature vector: a normalized set of moments of the pixel color distribution with the tile. That is, if μ_i^c is the ith moment of the cth of the three color components, then we choose some cutoff n and use a feature vector with $3n$ components

$$\frac{1}{i!}\mu_i^c \quad 1 \leq i \leq n$$

The feature vectors are used to create a 128-nearest neighbor, or approximate nearest neighbor, graph based on the features. We are then able to create a diffusion matrix, W, chosen to conserve energy by the formula:

$$\widetilde{W_{i,j}} = \begin{cases} e^{-\frac{||x_i - x_j||^2}{2\epsilon^2}} & x_i \text{ and } x_j \text{ are neighbors} \\ 0 & o.w. \end{cases}$$

$$W_{i,j} = \frac{\widetilde{W_{i,j}}}{\sum\limits_{i=1}^{n} \widetilde{W_{i,j}}}$$

where $\epsilon = \sqrt{\frac{1}{n}\sum\limits_{i=1}^{n} ||x_i - x_{nearest(i)}||^2}$.

C. Identifying Texture Tiles

Once the manifold is created, it can be used to select texture tiles on the fly. By focusing on the user's selection, rather than trying to classify all textures in the image into classes as done in previous work, we introduce another major efficiency.

Specifically, when the user clicks on a point, i, within the image the heat distribution over all tiles, $h_i(x_j)$, is calculated for the entire manifold. This is done by setting the value of the node i to 1 and setting the value of the rest of the nodes to 0 and then iterating the equation:

$$h^{(t+1)}(x_i) = \sum_j W_{i,j} h^t(x_j)$$

Since $W_{i,j}$ is sparse, the complexity of the above equation grows linearly with the number of tiles making the diffusion tractable even for large images.

With the values of $h_i(x_j)$ determined, tiles corresponding to the selected texture are found by simply taking a pre-set number of tiles with a maximum value of $h_i(x_j)$.

D. Exemplar Generation

It is possible to modify most MRF algorithms to use this tile set as an exemplar for generating a texture of arbitrary extent, instead of a single image. MRF texture synthesis algorithms are now extremely well studied, and many algorithms exist for many different possible situations [2].

III. Future Work and Conclusion

This work demonstrates that diffusion distance manifolds can be used at interactive rates to distinguish textures in arbitrary images or to generate texture exemplars for texture synthesis algorithms. The method is non-local and able to accumulate information from a large number of image regions. In fact, our implementation can combine patches from multiple images and is only limited by computer memory. While we provided as much information here as room permitted, a full account is available online [1].

Acknowledgment

We thank the rest of the Yale Graphics Group and professor Steven Zucker for their help, advice, support. We acknowledge Cyril Zhang for his assistance with the prototypes that predated the system described in this work and for his invaluable suggestions and discussions. Elliot Lockerman provided the photograph used in Fig. 1.

This work was funded by NSF Grant IIS-1064412

References

[1] Y. D. Lockerman, S. Xue, J. Dorsey, and H. Rushmeier, "Creating texture exemplars from unconstrained images," Yale University, Tech. Rep. TR1483, October 2013. [Online]. Available: http://graphics.cs.yale.edu/site/tr1483

[2] L. Wei, S. Lefebvre, V. Kwatra, and G. Turk, "State of the art in example-based texture synthesis," Eurographics 2009, State of the Art Report, EG-STAR, 2009.

[3] J. Lu, J. Dorsey, and H. Rushmeier, "Dominant texture and diffusion distance manifolds," in Computer Graphics Forum, vol. 28, no. 2. Wiley Online Library, 2009, pp. 667–676.

[4] B. Nadler, S. Lafon, R. Coifman, and I. Kevrekidis, "Diffusion maps, spectral clustering and eigenfunctions of Fokker-Planck operators," Arxiv preprint math/0506090, 2005.