Dominant Texture and Diffusion Distance Manifolds

Jianye Lu Julie Dorsey Holly Rushmeier

Computer Graphics Group, Yale University

Abstract

Texture synthesis techniques require nearly uniform texture samples, however identifying suitable texture samples in an image requires significant data preprocessing. To eliminate this work, we introduce a fully automatic pipeline to detect dominant texture samples based on a manifold generated using the diffusion distance. We define the characteristics of dominant texture and three different types of outliers that allow us to efficiently identify dominant texture in feature space. We demonstrate how this method enables the analysis/synthesis of a wide range of natural textures. We compare textures synthesized from a sample image, with and without dominant texture detection. We also compare our approach to that of using a texture segmentation technique alone, and to using Euclidean, rather than diffusion, distances between texture features.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Computing Methodologies—Three-dimensional graphics and realism Color, shading, shadowing, and texture

1. Introduction

Texture describes appearance that looks homogeneous at large scales but random at small scales and has the commonly accepted characteristics of locality, repetition, and randomness. Texture has found widespread use in realistic appearance modeling and synthesis in computer graphics [DRS08]. However, not all images are suitable as a source for texture samples: some input images have extraneous elements that are not a part of target textures or that have very distracting features; some input images have variations that depend on the environment or that are designed by a user. A uniform texture is desired especially when texturing 3D objects. Spatial variations in the object texture should be under the designer's control (e.g. spelling out the word "I3D" with flowers in the grass in Fig. 10 of [Ash01]), or should be computed to be consistent with variations in shading and shadowing, when the object is illuminated. All these issues can be summarized into one key problem: given a source image, how can we identify patches that are suitable as texture samples?

In previous work, users were required to carefully prepare the texture sample image. This can be a tedious trialand-error process, and requires insight into the texture synthesis process to identify an appropriate texture sample size [KW07], as well as to eliminate areas not suitable for use.

© 2008 The Author(s)

To address this problem, we extract the *dominant texture*, a large group of homogeneous texture elements, from the input image. We use pixels grouped in small patches as texture elements (referred to as "texels" hereafter) and study their distribution in feature space based on a simple observation: variations within dominant texels are much smaller than those between a dominant texel and an outlier; in other words, elements of dominant texture are relatively much closer to each other. When a single texture image contains myriad observations of such texture elements, we can detect dominant texture by a data-driven approach without any prior knowledge. Moreover, we note that dominant texture elements are easily identified by humans despite their complicated patterns. We conjecture that such texel points of interest lie on a low-dimensional manifold within the original high-dimensional space, similar to many other perceptionrelated applications. Therefore, we suggest the diffusion distance, a non-linear approach in manifold construction, to better detect such geometric structure.

In this paper, we propose a pipeline that automatically determines texture patch size and the collection of texture patches for any given source image. Referring to Figure 1, starting from a texture image, we construct a binary mask of dominant texture using manifold analysis based on diffusion distance. Comparing texture synthesis results with and

Journal compilation © 2008 The Eurographics Association and Blackwell Publishing Ltd. Published by Blackwell Publishing, 9600 Garsington Road, Oxford OX4 2DQ, UK and 350 Main Street, Malden, MA 02148, USA.



(a) source image (b) dominant texture mask (c) synthesis with full image (d) synthesis with dominant texture

Figure 1: Given a texture image (a), we construct a binary image that masks dominant texture based on manifold density and diffusion distance to the kernel texel (b). Compared to texture synthesis with full source image in (c), synthesis only with dominant texture in (d) gives us more homogeneous appearance and less distraction.

without such a mask side-by-side, we can see that dominant texture masks greatly improve the synthesis quality. Our method eliminates the need for client-side parameter tweaking and input image editing, both of which are essential in order for most current techniques to achieve the desired results. We make the following key contributions:

- apply diffusion distance manifolds to natural texture analysis and propose practical implementations to enable their full potential;
- detect texture patch sizes and extract the collection of texture patches used for conventional texture analy-sis/synthesis approaches;
- present a pipeline that is fully automatic and robust for a wide range of natural texture images that previous work has failed to address.

The rest of this paper is organized as follows: first, we review related work in texture analysis/synthesis and manifold-based analysis in Section 2; next, we detail our manifold construction with diffusion distance in Section 3, and formally define the dominant texture extraction problem in Section 4; then, we detail our pipeline in Section 5, introduce a practical implementation in Section 6; then, we demonstrate and compare experimental results using wide range of natural texture samples with other methods in Section 7; finally, we conclude with a summary and future work in Section 8.

2. Related Work

Our work is related to previous research in texture analysis/synthesis and manifold-based data analysis.

2.1. Texture Analysis and Synthesis

Considerable work has been devoted to texture-related research, including texture classification, which retrieves similar samples in the training set given any texture image, texture segmentation, which identifies differently textured areas in one image, and example-based texture synthesis, which

generates a large patch of texture similar to a small input sample. Much of this work adopted parametric texture models and considered either global or local similarity for texture clustering [HB95, Bon97, LM01, FB03]. These approaches rely on predefined texture features, about which we have limited knowledge, and can only be applied to certain types of textures. Later, non-parametric methods were proposed for texture synthesis: a partially synthesized neighborhood is compared to given examples based on the Markov Random Field (MRF) assumption, and the best match is used as new synthesis pixel [EL99]. Combined with subsequent improvements, such methods have proven well suited for efficient and high-quality texture synthesis (see [KW07] for a comprehensive overview). However, these methods do not validate MRF properties of the input samples before synthesis, and thus fail on images with complicated natural patterns without careful preparation.

Our system combines the advantages of parametric and non-parametric methods by inserting dominant texture extraction before MRF-based synthesis. Unlike previous parametric texture models, we use raw color pixel values of texture patches as our texture feature, and construct diffusiondistance-based manifolds to better examine texture homogeneity, which has proven to work well for a wide range of natural textures without *a priori* knowledge. Detected dominant textures are fed to conventional non-parametric synthesis methods, such as *Image Quilting* [EF01], which we apply in this paper. Dominant textures can also be used by other texture analysis frameworks, such as *Near-Regular Textures* analysis [LLH04]. We show that dominant texture detection improves texture synthesis quality in 2D and on 3D objects.

Previous work also used masks to avoid mixing unrelated textures. For example, Hertzmann et al. [HJO*01] proposed "texture-by-numbers" to synthesize new images based on manually assigned labels; Zalesny et al. [ZFCG05] automatically segmented the input image into homogeneous regions with traditional gray-level co-occurrence methods before synthesis. In our work we automatically generate a dominant texture mask using raw texture patch pixel values on a manifold based on diffusion distance. Other work has developed specific models to account for or to correct certain types of texture variations. For example, Liu et al. [LLH04] modeled geometric variations due to non-planar surfaces or perspective viewing; Xue et al. [XWT*08] modeled shading variations as the product of reflectance and illuminance. Our work does not rely on any prior knowledge.

2.2. Manifold-based Analysis

Manifold-based analysis considers only short-distance relationships between data points and reconstructs the global structure by "stitching" such small pieces. Manifold models are considered more suitable for perception-related applications, where observations commonly lie only on a lowdimensional manifold in the original high-dimensional feature space [SL00]. Related approaches have been revived in the past few years, including isometric mapping (Isomap) [TdSL00], locally linear embedding (LLE) [RS00], diffusion maps based on diffusion distance [Laf04], and others. These methods have been applied widely in graphics, such as in charting in BRDF modeling [MPBM03], simplicial complexes for texture modeling [MZD05], Isomap and LLE in appearance-space texture synthesis [LH06], and an Isomaplike approach in estimating weathering degree maps from a single image [WTL*06]. In our previous work [LGG*07], we applied diffusion maps to analyze correlation between appearance patches and environmental contexts on a weathered surface. In this work we extend the application of diffusion maps to the identification of similar textures.

Coifman et al. showed the diffusion-maps-based methods to be more reliable for recovering the smooth data structure and more robust to data noise [CLL*05, LKC06]. In this paper, we demonstrate their advantage in texture element modeling by showing an actual distribution of our data points, and by comparing our dominant texture detection results with Euclidean-distance-based analysis. This general idea is similar to the appearance manifold described in [WTL*06], however the authors did not consider outliers in input samples. In subsequent work by Xue et al., they dealt primarily with shading variations [XWT*08]. In our approach, we apply diffusion distance instead of geodesic distance for manifold construction, use image patches as elements to better characterize textures, and propose a fully automatic pipeline. Pavan and Pelillo studied dominant sets in a graph using pairwise clustering and introduced a similar algorithm to ours [PP07]. However, their approach has a different mathematical starting point. In addition, we provide intuitive reasoning, a practical implementation, including key parameter selection, and demonstrate results on a wide range of input images.

3. Manifold Construction based on Diffusion Distance

To quantify how close two texels are in feature space, we need a dissimilarity measurement. Euclidean distance, while

© 2008 The Author(s) Journal compilation © 2008 The Eurographics Association and Blackwell Publishing Ltd.



Figure 2: Comparison between Euclidean and diffusion distances: point B and point C are equally distant from point A in terms of Euclidean distance (highlighted in solid green lines). Heat starting from point A reaches point B faster because of more paths connecting them (highlighted in dashed blue lines); hence, in terms of the diffusion distance, point B is considered "closer" to point A compared to point C.

$X \subset \mathbb{R}^D$	source patches, with cardinality N
$\mathscr{M} \subset \mathbb{R}^d$	embedded manifold
$X' \subset X$	subsampled patches, with cardinality $N' \ll N$
$\mathscr{M}' \subset \mathbb{R}^{d'}$	subsampled manifold from X'
$ x_i - x_j $	Euclidean distance
ε	Gaussian kernel size
$\delta(x_i, x_j)$	diffusion distance *
W _{i,j}	kernel function *
Ŵ	kernel matrix *
$\mu(x_i)$	manifold density *
A	normalized heat diffusion matrix
$h_k(x_i)$	dominant heat distribution from x_k

Table 1: Summary of notation (* A superscript of (ε) is used when the Gaussian kernel size is explicitly specified).

simple, fails to recover the geometric structure of lowdimensional manifolds embedded in high-dimensional feature space, where perception-related data usually lie [SL00].

Instead, we use diffusion distance, a non-linear distance measurement, which computes the distance between two points by simulating heat diffusion and recording the time of traversal based on random diffusion. Figure 2 compares Euclidean distance and diffusion distance measurements on a "dumbbell-shaped" data set. The diffusion distance considers the width of the region connecting two points, in a sense the number of paths connecting the two points. The diffusion distance is demonstrated to better preserve structure smoothness and is more robust to both outliers and small disturbances within the graph structure [CLL*05].

A formal definition of the diffusion distance is as follows (see Table 1 for a summary of math notation): given data set $X \subset \mathbb{R}^D$ with N points, we define its $N \times N$ kernel matrix $W^{(\varepsilon)}$ with entries of pairwise heat conduction function (or, kernel function) values as

$$w_{i,i}^{(\varepsilon)} = e^{-\|x_i - x_j\|^2 / 2\varepsilon},\tag{1}$$

where $||x_i - x_j|| = (\sum_l (x_{i,l} - x_{j,l})^2)^{1/2}$ is the Euclidean distance between x_i and x_j , and ε is some Gaussian kernel width. When x_i and x_j are far away from each other, their heat conduction rate decreases quickly to zero, which means heat can only diffuse between two near points. Then we normalize the kernel matrix to factor out sampling density and apply singular vector decomposition (SVD) to project original data x_i to a point y_i in a new feature space \mathbb{R}^N , where the Euclidean distance of $||y_i - y_j||$ approximates the diffusion distance $\delta^{(\varepsilon)}(x_i, x_j)$ in the original feature space [LKC06].

Selecting an appropriate Gaussian kernel size ε is critical to reveal geometric structure using the diffusion distance. Coifman et al. proposed an automatic way to select ε : when we sum up both sides of Equation 1 with respect to all *i*'s and *j*'s, then approximate the right side with its mean value integral and with the integral in the manifold's tangent space, we have

$$\log\left(\sum_{i}\sum_{j}w_{ij}^{(\varepsilon)}\right) \approx \frac{d}{2}\log\varepsilon + \left(\frac{N^2(2\pi)^{d/2}}{\operatorname{vol}(\mathscr{M})}\right), \quad (2)$$

where *d* is the dimension of the manifold, *N* is the number of total observations on the manifold, and vol(\mathscr{M}) is the volume of the manifold (see [CSSS08] for details). Since ε from the linearity region of the curve between $\log(\sum_i \sum_j w_{i,j}^{(\varepsilon)})$ and $\log(\varepsilon)$ introduces the least error among approximations, such ε best reveals the manifold structure based on diffusion distances (see Figure 3 for an example). This guideline makes selecting ε more efficient since we do not rely on prior knowledge of our data set or brute-force trials. Moreover, we estimate the manifold dimension *d* based on the slope of that linearity region.



Figure 3: Gaussian kernel size selection from the linearity region on a curve between $\log(\sum_{i} \sum_{j} w_{i,j}^{(\varepsilon)})$ and $\log(\varepsilon)$. See Figure 1(a) for input texture image.

Explicit construction of the manifold requires expensive computation, including SVD on a large kernel matrix. If we are only interested in diffusion distances among certain data points, the diffusion framework provides an alternative measure by a Markov random walk [CLL*05]: Suppose one unit of heat starts diffusing from data point x_k at time 0. In order to represent the heat distribution, we use a vector χ_k of length N whose element is 1 only at the k-th component and 0 everywhere else. After time τ , the heat distribution at point x_i is defined as

$$h_k(x_i) = \left(A^{\tau} \chi_k\right)_i, \tag{3}$$

where *A*'s entries are defined as $a_{i,j} = w_{i,j} / \sum_j w_{i,j}$ to normalize the outgoing degrees for heat transition, and $(\cdot)_i$ is the *i*-th component of a vector. This value measures the heat accumulation at point x_i coming from x_k through all possible connections and thus qualitatively approximates the diffusion distance $\delta(x_i, x_k)$: the higher the value of $h_k(x_i)$, the closer these two points are. For τ selection, we use $dN^{(1/d)}$, which is large enough to propagate heat to neighboring points but not to reach the heat equilibrium (where all texels hold the same amount of heat). In this paper, we use this measure to simplify our diffusion distance estimation without explicitly computing the manifold.

4. Dominant Texture and the Texel Manifold

To build up texels for manifold construction, we first apply the 2D Fourier transform to the image and detect periodicity along every 45 degrees in frequency domain. The strongest periodicity response is used to estimate the texture patch size. This scheme is particularly important for regularly structured patterns. Unlike some previous work, we split the source image into *overlapping* patches with step size 1 along each direction, providing dense samples that are critical for non-linear manifold reconstruction, and partly compensating for possible inaccuracy in patch size estimation usually found in images of natural textures. Such patches are then expanded into high-dimensional feature vectors with all pixel values. We construct a graph using texel vectors as nodes and kernel function values as weights over edges between each pair of texels.

Explicit computation of diffusion distances between dense samples is impracticable in our pipeline. Instead, we use two attributes of texels on the manifold to identify texels of interest:

Density $\mu(x_i)$: measures local compactness of a texel x_i on the manifold and can be estimated using the sum of weights over all edges connecting x_i

$$\mu^{(\varepsilon)}(x_i) = \sum_j w_{i,j}^{(\varepsilon)}.$$
(4)

Dominant heat distribution $h_k(x_i)$: as defined in Equation 3, approximates the diffusion distance $\delta(x_i, x_k)$. When



Figure 4: Texel classification: (a) and (b) show abstracted graphs of the texel distribution in feature space and classification in a space spanned by density and dominant heat distribution; (c) and (d) show corresponding graphs from real data (See Figure 1 for the source image). To visualize texels in high-dimensional space in (c), we project them onto a subspace spanned by the first three principal components using PCA; the color bar on the right shows the data density: red for high densities, blue for low densities; see http://graphics.cs.yale.edu/DominantTexture/ for animated 3-D view. We can see a large dense group on the right (corresponding to the brick wall, later classified as Type 0), a small dense group on the left (corresponding to part of the pipe, later classified as Type I), a few points connecting these two groups with medium densities (later classified as Type II), and many sparse points with low densities scattered around (later classified as Type III).

 x_k is selected from the dominant texture set, this value also implies how probable it is that x_i belongs to the same set.

Based on these two attributes, we classify texels into four different types, as follows (see Figure 4):

- **Texel Type 0 (Dominant Texels)**: texels that form the dominant texture. These texels have high densities with support from near neighbors, cover the main region in the image, and have high dominant heat distributions;
- **Texel Type I (Small-Group Outliers)**: texels that form some other texture groups covering relatively small regions in the image. These texels might still have high densities if their internal variations are sufficiently small. However, their dominant heat distributions would be significantly lower than those of texel type 0;
- Texel Type II (Bridging Outliers): texels connecting Type 0 and Type I groups. Such "texel bridges" are usually very narrow compared to the size of other texel groups, hence densities are low. On the other hand, they are closer

to the main texel group and have higher dominant heat distributions compared to texel type I;

• **Texel Type III (Scattered Outliers)**: noise scattered throughout the feature space, far from other texels. These have low densities and dominant heat distribution.

Although only dominant texels (Type 0) are explicitly detected and used in our pipeline, texel classification as outlined above provides a complete picture of texel characteristics from natural texture images for further study.

5. Pipeline

With the diffusion distance manifold we built in Section 4, we propose the following pipeline for dominant texture detection, with the data flow shown in Figure 5.

Step 1: Texture granularity detection: Given a source image \mathscr{I} , we apply the 2-D Fourier transform and estimate the proper patch size n_p in frequency domain.

^{© 2008} The Author(s) Journal compilation © 2008 The Eurographics Association and Blackwell Publishing Ltd.



Figure 5: Flow chart of dominant texture detection.

Step 2: Texel vectors preparation: We split \mathscr{I} into *N* overlapping patches of size $n_p \times n_p$ centered around each pixel, then expand such patches into *D*-dimensional vectors $\{x_i\} \subset \mathbb{R}^D$, where $D = 3n_p^2$ for a color texture image.

Step 3: Kernel matrices computation: We estimate pairwise kernel function values $w_{i,j}^{(\varepsilon)}$ between texels using a series of predefined ε candidates (see Equation 1), then construct a series of kernel matrices $W_{N \times N}^{(\varepsilon)}$.

Step 4: Kernel size selection: We plot a curve between $\log(\sum_i \sum_j w_{i,j}^{(\varepsilon)})$ and $\log(\varepsilon)$, then select ε around the linearity region and estimate the manifold dimension *d* based on the slope of that linearity region.

Step 5: Texel density estimation: We estimate the density $\mu(x_i)$ for each texel based on the kernel size ε (see Equation 4).

Step 6: Dominant Texels Detection: We first select x_k with the highest density as our "seed" texel, then scatter all texels in a 2-D plane spanned by their densities $\mu(x_i)$ and dominant heat distribution values $h_k(x_i)$ with respect to x_k (see Figure 4(d)). If texels with high densities and high dominant heat distribution values (those in the top right quadrant) consist of more than 40% of all texels, they are considered as dominant texels (Type 0). Otherwise, they are considered as small-group outliers (Type I), and are removed along with texels with high heat distribution values but low densities (those in the top left quadrant, as bridging texels, Type II). We will re-run this step with all remaining texels until we detect the dominant texel group.

In this step, thresholds for density and dominant heat diffusion are estimated automatically (see the vertical and horizontal red lines in Figure 4(d)): we order density values from low to high, then select the valley between the first and second peaks on the smoothed histogram as density threshold; if no second peak is detected, we use the value of the 0.2quantile point instead. For the heat value threshold, we build the smooth histogram in a similar way, then select the first valley beyond the 0.4-quantile point as the threshold, making sure the size of main texel group is at least 40% of the image; if no such valley is found, we simply consider all remaining texels as scattered outliers (Type 0).

Step 7: Dominant texture mask: If no dominant texels are detected, we will reject the source image. Otherwise, we create a binary dominant texture mask \mathscr{I}_{mask} by first placing 1's at the centers of dominant texels on the source image and 0's elsewhere, then applying a convolution of this binary image with a $n_p \times n_p$ square block of 1's. \mathscr{I}_{mask} masks "good pixels" that compose the dominant texture and is used along with source image \mathscr{I} in conventional texture synthesis schemes.

6. Practical Implementation

Similar to other data-driven approaches, we need a large amount of data to construct a reliable estimate about the data distribution without prior knowledge. However, direct application of the pipeline above is still limited by computational resources. We use several techniques to make our pipeline practical in terms of space and time complexity.

6.1. Linear Dimensionality Reduction

We notice from Equation 1 that diffusion distances depend only on the estimation of Euclidean distances, which can be well approximated with linear dimensionality reduction. We first apply principal component analysis (PCA) on our source texel set to reduce its dimension to $D' \ll D$, where 99.9% of the energy is still preserved. Thus, diffusion distances in Equation 1 can be estimated in a significantly lower-dimensional space without losing much accuracy.

6.2. Data Subsampling

Suppose our original data set X with N points is oversampled from a *d*-dimensional manifold \mathcal{M} , and a subset $X' \subset X$ with N' points (N' \ll N) is still sufficient to cover \mathcal{M} . We have the following equation for the Gaussian kernel sizes ε for the original data set and ε' for the simplified data set,

$$N'/N = \left(\varepsilon/\varepsilon'\right)^d.$$
 (5)

If we approximate *d* by *d'* based on *X'* (see Equation 2), then we have an approximation of ε for \mathcal{M}

$$\varepsilon_{approx} = \varepsilon' \left(N'/N \right)^{1/d'} . \tag{6}$$

In our pipeline, we randomly select a small subset $X' \subset X$ with cardinal number $N' \ll N$, estimate ε' and d' based on Equation 2, then estimate the manifold density of Xbased on diffusion distances with respect to the subset X'based on ε_{approx} in Equation 6. Similar ideas can be found in [dST04], where landmark points are used to accelerate multi-dimensional scaling. A small subset X' can greatly accelerate the computation but tends to over-simplify \mathscr{M} ; thus, d cannot be well approximated by d', and we have only a poor estimation of ε for \mathscr{M} . For our test data, an empirical number $N' \sim N^{0.7}$ balances accuracy and acceleration very well; we only need to estimate $N \times N'$ pairs of diffusion distances instead of N^2 pairs.

6.3. Graph Sparsification

During heat diffusion simulation, we need to maintain a matrix *A* of size $N \times N$ (see Equation 3), which is too large for memory. In fact, a texel will only have a few neighbors with non-negligible connection in terms of the kernel function defined in Equation 1. Therefore, we consider only *k* nearest neighbors ($k \ll N$) for each texel based on approximate nearest neighbor searching (ANN) [AMN*98], resulting in a sparse diffusion matrix *A* with $k \times N$ non-zero entries. After graph sparsification, the right hand side of Equation 3 can be rewritten as $(A \cdots (A(A\chi_k)))_i$ and computed efficiently as a series of multiplications between a sparse matrix and a vector. In our implementation, a conservative $k = \log^2 N$ balances efficiency and accuracy very well.

7. Results

We built our pipeline on a Windows platform with a 2.80GHz CPU and 2GB memory. We implemented steps 2 and 3 in C++ because these steps involve large amounts

© 2008 The Author(s) Journal compilation © 2008 The Eurographics Association and Blackwell Publishing Ltd. of simple data processing and we were concerned with efficiency. We implemented other steps in Matlab for better module flexibility and result visualization. For a general color image of size 125×94 and patch size 11, it takes approximately 3 minutes for image patch preprocessing, 15 minutes for kernel matrix computation, and less than 10 seconds on dominant texture detection based on density and dominant heat distribution.

7.1. Dominant Texture Detection Results

We apply our pipeline to a wide range of natural textures that contain different extraneous elements, including artificial objects, deterioration patterns like scratches and cracks, flow patterns, shadows, rust, dirt, biological growth, and others (see Figure 6). All these examples are taken from natural scenes with rich texture information, but are difficult to use as texture samples without some preprocessing. Our pipeline detects dominant textures in most of these cases without prior knowledge.

7.2. Comparison of Texture Synthesis with and without Dominant Texture Detection

We generate binary masks for dominant textures based on the detection results in Section 7.1, then synthesize large 2-D texture patches from these input images using *Image Quilting* [EF01]. We compare synthesis results with and without such a mask side-by-side in the first two columns in Figure 6. Our masks remove outliers while still preserving rich variations of the dominant texture. In other words, our pipeline automatically builds up a "clean" texture sample image for synthesis purposes, a step which is missing in all previous non-parametric synthesis approaches.

One might argue that those outliers sometimes enrich texture variations and can be viewed as part of the texture pattern. However, there are two important reasons that necessitate dominant texture detection: first, structured elements in the input image (such as pipes and windows in the first two examples in Figure 6) do not follow MRF-assumption and cannot be repeated randomly; second, global texture variations are highly correlated to the environment and cannot be synthesized with local neighborhoods. Simple repetition of such patterns is very distracting (see the last two examples in Figure 6). Our approach can be viewed as a form of preprocessing for other synthesis approaches oriented toward global variations [WTL*06, LGG*07], where global variations over uniform textures under user control are possible.

To validate the second point, we synthesize textures from 2-D images onto 3-D surfaces by first splitting and flattening 3-D meshes into 2-D patches then synthesizing on the texture atlas. We compare synthesis results with and without the dominant texture mask in Figure 7. When using full source images, we cannot guarantee consistency between appearance and geometry, or object shadow directions with scene



Figure 6: We compare dominant texture detection and corresponding texture synthesis results of four image samples. From left to right are synthesis with (a) full source image, (b) dominant texture based on diffusion distances, (c) dominant texture based on inverse Euclidean distances (see Section 7.3), and (d) the largest texture segment from nCut (see Section 7.4).



Figure 7: *Texture synthesis on 3D objects. From top to bottom: source texture image and dominant texture mask, synthesis with full source texture image (note the inconsistency between texture and geometry, and between shadows from the texture map and those from scene relighting), and synthesis with only dominant texture.*

illumination. Realistic renderings can only be achieved by post-processing scenes with "clean" textures as shown at the bottom of Figure 7.

7.3. Comparison with Euclidean-Distance-based Methods

To examine the value of using diffusion distances, we replace the kernel function in Equation 1 with the inverse of Euclidean distance, as follows

$$w_{i,j} = \frac{1}{\|x_i - x_j\|}.$$
(7)

We compare both detection results side-by-side in the second and the third columns in Figure 6. In most cases, diffusion distance manifolds provide more accurate detection results with more good texel candidates for texture synthesis.

7.4. Comparison with Texture Segmentation

To examine the value of building the texel manifold, we also compare our dominant texture detection results with image segmentation results from *Multiscale Normalized Cuts Segmentation* [CBS05] with two segments in the rightmost column in Figure 6. We note that unlike our approach, *nCut* is

© 2008 The Author(s) Journal compilation © 2008 The Eurographics Association and Blackwell Publishing Ltd. not completely automatic: the user must specify the number of texture segments. In general, their results tend to preserve large areas of uniform texture, usually missing small isolated outliers or further splitting uniform areas into smaller patches. Again, our pipeline gives us more accurate detection results.

7.5. Failure Cases

Our pipeline gives poor results for two types of cases (see Figure 8). Diffusion distance manifolds tend to connect texel groups with smooth and broad transitions and therefore cannot correctly respond to such images; our pipeline also fails for complicated crack patterns that have rich variations in scale, orientation, and gap sizes. To our knowledge, such patterns have not yet been well addressed by any data-driven approach.

8. Conclusion and Future Work

In this paper, we address the problem of dominant texture detection in texture modeling and synthesis. We construct a graph with texture elements and a kernel function, then propose an automatic pipeline to detect the dominant texture using diffusion distance with acceleration techniques. Experimental results show that our system covers a wide range of natural texture samples without domain knowledge.



Figure 8: Failure samples with dominant texture masks embedded: image with smooth transition between solid and spotted paint textures; cracks of different orientation and gap sizes.

All source data and additional examples can be found at http://graphics.cs.yale.edu/DominantTexture/. In future work, we will study the explicit shapes of texture manifolds, and apply state-of-the-art numerical methods to accommodate even larger data sets to cover more complicated and generalized texture samples.

Acknowledgement The authors would like to thank Steven Gortler for commenting on an early version of the paper and Franco Woolfe for invaluable discussions. This material is based upon work supported by the National Science Foundation under Grant No. 0528204.

References

- [AMN*98] ARYA S., MOUNT D. M., NETANYAHU N. S., SIL-VERMAN R., WU A. Y.: An optimal algorithm for approximate nearest neighbor searching fixed dimensions. J. ACM 45, 6 (1998), 891–923.
- [Ash01] ASHIKHMIN M.: Synthesizing natural textures. In I3D '01: Proceedings of the 2001 symposium on Interactive 3D graphics (2001), pp. 217–226.
- [Bon97] BONET J. S. D.: Multiresolution sampling procedure for analysis and synthesis of texture images. In SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques (1997), pp. 361–368.
- [CBS05] COUR T., BENEZIT F., SHI J.: Spectral segmentation with multiscale graph decomposition. In CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Vol 2 (2005), IEEE Computer Society, pp. 1124–1131.
- [CLL*05] COIFMAN R. R., LAFON S., LEE A. B., MAGGIONI M., NADLER B., WARNER F., ZUCKER S. W.: Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *Proceedings of the National Academy of Science 102* (May 2005), 7426–7431.
- [CSSS08] COIFMAN R., SHKOLNISKY Y., SIGWORTH F., SINGER A.: Graph laplacian tomography from unknown random projections. *IEEE Transactions on Image Processing 17*, 10 (2008), 1891–1899.
- [DRS08] DORSEY J., RUSHMEIER H., SILLION F.: Digital Modeling of Material Appearance. The Morgan Kaufmann Series in Computer Graphics. Boston: Morgan Kaufmann / Elsevier, 2008.
- [dST04] DE SILVA V., TENENBAUM J. B.: Sparse multidimensional scaling using landmark points, 2004. Manuscript.
- [EF01] EFROS A. A., FREEMAN W. T.: Image quilting for texture synthesis and transfer. In SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques (2001), ACM Press, pp. 341–346.

- [EL99] EFROS A. A., LEUNG T. K.: Texture synthesis by nonparametric sampling. In *ICCV '99: Proceedings of the International Conference on Computer Vision - Vol 2* (1999), IEEE Computer Society, pp. 1033–1038.
- [FB03] FISCHER B., BUHMANN J. M.: Path-based clustering for grouping of smooth curves and texture segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 25, 4 (2003), 513–518.
- [HB95] HEEGER D. J., BERGEN J. R.: Pyramid-based texture analysis/synthesis. In SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques (1995), ACM Press, pp. 229–238.
- [HJO*01] HERTZMANN A., JACOBS C. E., OLIVER N., CUR-LESS B., SALESIN D. H.: Image analogies. In SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques (2001), ACM Press, pp. 327–340.
- [KW07] KWATRA V., WEI L.-Y.: Example-based texture synthesis. Course in SIGGRAPH, 2007.
- [Laf04] LAFON S. S.: *Diffusion maps and geometric harmonics*. PhD thesis, Yale University, May 2004.
- [LGG*07] LU J., GEORGHIADES A. S., GLASER A., WU H., WEI L.-Y., GUO B., DORSEY J., RUSHMEIER H.: Contextaware textures. ACM Trans. Graph. 26, 1 (2007), 3.
- [LH06] LEFEBVRE S., HOPPE H.: Appearance-space texture synthesis. In SIGGRAPH '06: ACM SIGGRAPH 2006 Papers (New York, NY, USA, 2006), ACM Press, pp. 541–548.
- [LKC06] LAFON S., KELLER Y., COIFMAN R. R.: Data fusion and multicue data matching by diffusion maps. *IEEE Trans. Pattern Anal. Mach. Intell.* 28, 11 (2006), 1784–1797.
- [LLH04] LIU Y., LIN W.-C., HAYS J. H.: Near regular texture analysis and manipulation. ACM Transactions on Graphics (SIG-GRAPH 2004) 23, 3 (August 2004), 368 – 376.
- [LM01] LEUNG T., MALIK J.: Representing and recognizing the visual appearance of materials using three-dimensional textons. *Int. J. Comput. Vision 43*, 1 (2001), 29–44.
- [MPBM03] MATUSIK W., PFISTER H., BRAND M., MCMIL-LAN L.: A data-driven reflectance model. ACM Trans. Graph. 22, 3 (2003), 759–769.
- [MZD05] MATUSIK W., ZWICKER M., DURAND F.: Texture design using a simplicial complex of morphable textures. ACM Trans. Graph. 24, 3 (2005), 787–794.
- [PP07] PAVAN M., PELILLO M.: Dominant sets and pairwise clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence 29*, 1 (2007), 167–172.
- [RS00] ROWEIS S. T., SAUL L. K.: Nonlinear dimensionality reduction by locally linear embedding. *Science 290*, 5500 (Dec 2000), 2323–2326.
- [SL00] SEUNG H. S., LEE D. D.: The manifold ways of perception. *Science* 290, 5500 (Dec 2000), 2268–2269.
- [TdSL00] TENENBAUM J. B., DE SILVA V., LANGFORD J. C.: A global geometric framework for nonlinear dimensionality reduction. *Science* 290, 5500 (Dec 2000), 2319–2323.
- [WTL*06] WANG J., TONG X., LIN S., PAN M., WANG C., BAO H., GUO B., SHUM H.-Y.: Appearance manifolds for modeling time-variant appearance of materials. *ACM Trans. Graph.* 25, 3 (2006), 754–761.
- [XWT*08] XUE S., WANG J., TONG X., DAI Q., GUO B.: Image-based material weathering. *Computer Graphics Forum 27*, 2 (Apr. 2008), 617–626.
- [ZFCG05] ZALESNY A., FERRARI V., CAENEN G., GOOL L. V.: Composite texture synthesis. *Int. J. Comput. Vision* 62, 1-2 (2005), 161–176.

© 2008 The Author(s) Journal compilation © 2008 The Eurographics Association and Blackwell Publishing Ltd.