PYRAMID-BASED TEXTURE ANALYSIS/SYNTHESIS

David J. Heeger

Department of Psychology
Stanford University
Stanford, CA 94305
heeger@white.stanford.edu
http://white.stanford.edu

James R. Bergen

SRI David Sarnoff Research Center
Princeton, NJ 08544
jrb@sarnoff.com

ABSTRACT

This paper describes a method for synthesizing images that match the texture appearance of a given digitized sample. This synthesis is completely automatic and requires only the "target" texture as input. It allows generation of as much texture as desired so that any object can be covered. The approach is based on a model of human texture perception, and has potential to be a practically useful tool for image processing and graphics applications. Some of this work has been reported previously [8].

1. INTRODUCTION

Textures have often been classified into two categories, deterministic textures and stochastic textures. A deterministic texture is characterized by a set of primitives and a placement rule (e.g., a tile floor). A stochastic texture, on the other hand, does not have easily identifiable primitives (e.g., granite, bark, sand). Many real-world textures have some mixture of these two characteristics (e.g. woven fabric, woodgrain, plowed fields).

Much of the previous work on texture analysis and synthesis can be classified according to what type of texture model was used. Some of the successful texture models include reaction-diffusion [17, 18], frequency domain [9], fractal [5, 10], and statistical/random field [2, 4, 6, 7, 11, 13] models. Some (e.g., [6]) have used hybrid models that include a deterministic (or periodic) component and a stochastic component. In spite of all this work, scanned images and hand-drawn textures are still the principle source of texture maps in computer graphics.

This paper focuses on the synthesis of stochastic textures. Our approach is motivated by research on human texture perception. Current theories of texture discrimination are based on the fact that two textures are often difficult to discriminate when they produce a similar distribution of responses in a bank of (orientation and spatial-frequency selective) linear filters [1, 3]. The method described here, therefore, synthesizes textures by matching distributions (or histograms) of filter outputs. This approach depends on the principle (not entirely correct) that all of the spatial information characterizing a texture image can be captured in the first order statistics of an appropriately chosen set of linear filter outputs. Nevertheless, this model (though incomplete) captures an interesting set of texture properties.

Computational efficiency is one of the advantages of this approach compared with many of the previous texture analysis/synthesis systems. The algorithm involves a sequence of simple image processing operations: convolution, subsampling, upsampling, histograming, and nonlinear transformations using small lookup tables. These operations are fast, simple to implement, and amenable to special purpose hardware implementations (e.g., using DSP chips).

2. PYRAMID TEXTURE MATCHING

The pyramid-based texture analysis/synthesis technique starts with an input (digitized) texture image and a noise image (typically uniform white noise). The algorithm modifies the noise to make it look like the input texture. It does this by making use of an invertible image representation known as an *image pyramid*, along with a function that matches the histograms of two images.

2.1. Steerable Pyramid

An *image pyramid* is a particular type of subband transform. The defining characteristic of an image pyramid is that the basis/projection functions are translated and dilated copies of one another (translated and dilated by a factor or 2^j for some integer j). The subbands are computed by convolving and subsampling. For each successive value of j, the subsampling factor is increased by a factor of 2. This yields a set of subband images of different sizes (hence the

Supported by an NIMH grant (MH.0228), an NSF grant (IRI9320017), and an Alfred P. Sloan Research Fellowship to DJH.

0-8186-7310-9/95 $4.00 © 1995 IEEE

name image pyramid) that correspond to different frequency bands.

In an independent context, mathematicians developed a form of continuous function representation called *wavelets* (see [16] for an introduction to wavelets), that are very closely related to image pyramids. Both wavelets and pyramids can be implemented in an efficient recursive manner.

Textures that have oriented or elongated structures are not captured by radially symmetric basis functions. To synthesize anisotropic textures, we adopt the steerable pyramid transform. [12, 15]. This transform decomposes the image into several spatial frequency bands. In addition, it further divides each frequency band into a set of orientation bands.

The steerable pyramid, unlike most discrete wavelet transforms used in image compression algorithms, is non-orthogonal and overcomplete; the number of pixels in the pyramid is much greater than the number of pixels in the input image. This is done to minimize the amount of aliasing within each subband. Avoiding aliasing is critical because the pyramid-based texture analysis/synthesis algorithm treats each subband independently.

The steerable pyramid is self-inverting; the filters used for analysis (building the pyramid) are the same as those use for synthesis (collapsing the pyramid). This allows the reconstruction (synthesis) to be efficiently computed despite the non-orthogonality.

A C code implementation of the steerable pyramid is available at:
`http://www.cis.upenn.edu/~eero/home.html`.

Psychophysical and physiological experiments suggest that image information is represented in visual cortex by orientation and spatial-frequency selective filters. The steerable pyramid captures some of the oriented structure of images similar to the way this information is represented in the human visual system.

2.2. Histogram Matching

Histogram matching is a generalization of histogram equalization. The algorithm takes an input image and coerces it via a pair of lookup tables to have a particular histogram. The two lookup tables are: (1) the cumulative distribution function (cdf) of one image, and (2) the inverse cumulative distribution function (inverse cdf) of the other image. These two lookup tables are used by the histogram matching function to modify an image to have the same histogram as another image.

2.3. Texture Matching

Our texture matching algorithm works by modifying an input noise image so that it looks like an input texture image. First, match the histogram of the noise image to the input texture. Second, make pyramids from both the (modified) noise and texture images. Third, loop through the two pyramid data structures and match the histograms of each of the corresponding pyramid subbands. Fourth, collapse the (histogram-matched) noise pyramid to generate a preliminary version of the synthetic texture. Matching the histograms of the pyramid subbands modifies the histogram of the collapsed image. In order to get *both* the pixel and pyramid histograms to match we iterate, rematching the histograms of the images, and then rematching the histograms of the pyramid subbands.

Whenever an iterative scheme of this sort is used there is a concern about convergence. In the current case we have not formally investigated the convergence properties of the iteration, but our experience is that it always converges. However, stopping the algorithm after several (5 or so) iterations is critical. As is the case with nearly all discrete filters, there are tradeoffs in the design of the steerable pyramid filters (e.g., filter size versus reconstruction accuracy). Since the filters are not perfect, iterating too many times introduces artifacts due to reconstruction error.

The core of the algorithm is histogram matching which is a spatially local operation. How does this spatially local operation reproduce the spatial characteristics of textures? The primary reason is that histogram matching is done on a representation that has intrinsic spatial structure. A *local* modification of a value in one of the pyramid subbands produces a *spatially correlated* change in the reconstructed image. In other words, matching the *pointwise* statistics of the pyramid representation does match some of the *spatial* statistics of the reconstructed image. Clearly, only spatial relationships that are represented by the pyramid basis functions can be captured in this way so the choice of basis functions is critical. As mentioned above, the steerable pyramid basis functions are a reasonably good model of the human visual system's image representation.

Some examples are shown in figure 1 (see for many more examples).

3. CONCLUSION AND EXTENSIONS

This paper presents a technique for creating an image that looks like a digitized texture image. The advantage of this approach is its simplicity; you do not have to be an artist and you do not have to understand a complex texture synthesis model/procedure. You just crop a textured region from a digitized image and run a program to produce as much of that texture as you want.

The approach presented in this paper, like other texture synthesis techniques, has its limitations (see [8] for examples). The analysis captures some but not all of the perceptually relevant structure of natural textures. Hence, this approach should be considered one of many tools for texturing objects in computer graphics.

Pyramid-based texture analysis/synthesis can also be used to produce solid textures for creating textured 3-d objects without the distortions inherent in texture mapping (see [8] for examples).

The pyramid-based texture scheme may be useful for image data compression. In our current implementation, each subband histogram is encoded with 256 bins. However the subband histograms of many ''natural'' images have a characteristic shape [14], suggesting that a very small number of parameters may be sufficient.

Finally, if a small number of parameters suffice, then it may also be possible to write an interactive tool for texture synthesis, with a slider for each parameter in the representation.

4. REFERENCES

[1] J R Bergen. Theories of Visual Texture Perception. In D Regan, editor, *Spatial Vision*, pages 114--133. CRC Press, 1991.

[2] R Chellappa and R L Kashyap. Texture Synthesis Using 2-D Noncausal Autoregressive Models. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 33:194--203, 1985.

[3] C Chubb and M S Landy. Orthogonal Distribution Analysis: A New Approach to the Study of Texture Perception. In M. S. Landy and J. A. Movshon, editors, *Computational Models of Visual Processing*, pages 291--301. MIT Press, Cambridge, MA, 1991.

[4] G C Cross and A K Jain. Markov Random Field Texture Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5:25--39, 1983.

[5] A Fournier, D Fussel, and L Carpenter. Computer Rendering of Stochastic Models. *Communications of the ACM*, 25:371--384, 1982.

[6] J M Francos, A Z Meiri, and B Porat. A Unified Texture Model Based on a 2D Wold-Like Decomposition. *IEEE Transactions on Signal Processing*, 41:2665--2678, 1993.

[7] A Gagalowicz. Texture Modelling Applications. *The Visual Computer*, 3:186--200, 1987.

[8] D J Heeger and J R Bergen. Pyramid-Based Texture Analysis/Synthesis. Proceedings of SIGGRAPH 95. In *Computer Graphics*, ACM SIGGRAPH, 1995, in press.

[9] J P Lewis. Texture Synthesis for Digital Painting. Proceedings of SIGGRAPH 84. In *Computer Graphics*, volume 18, pages 245--252. ACM SIGGRAPH, 1984.

[10] J P Lewis. Generalized Stochastic Subdivision. *ACM Transactions on Graphics*, 6:167--190, 1987.

[11] T Malzbender and S Spach. A Context Sensitive Texture Nib. In N M Thalmann and D Thalmann, editors, *Communicating with Virtual Worlds*, pages 151--163. Springer-Verlag, New York, 1993.

[12] P Perona. Deformable Kernels for Early Vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1995. To appear May 1995.

[13] K Popat and R W Picard. Novel Cluster-Based Probability Model for Texture Synthesis, Classification, and Compression. In *Proceedings of SPIE Visual Communications and Image Processing*, pages 756--768, 1993.

[14] D L Ruderman and W Bialek. Statistics of Natural Images: Scaling in the Woods. *Physical Review Letters*, 73:814--817, 1994.

[15] E P Simoncelli, W T Freeman, E H Adelson, and D J Heeger. Shiftable Multi-Scale Transforms. *IEEE Transactions on Information Theory, Special Issue on Wavelets*, 38:587--607, 1992.

[16] G Strang. Wavelets and Dilation Equations: A Brief Introduction. *SIAM Review*, 31:614--627, 1989.

[17] G Turk. Generating Textures on Arbitrary Surfaces Using Reaction-Diffusion. Proceedings of SIGGRAPH 91. In *Computer Graphics*, volume 25, pages 289--298. ACM SIGGRAPH, 1991.

[18] A Witkin and M Kass. Reaction-Diffusion Textures. Proceedings of SIGGRAPH 91. In *Computer Graphics*, volume 25, pages 299--308. ACM SIGGRAPH, 1991.

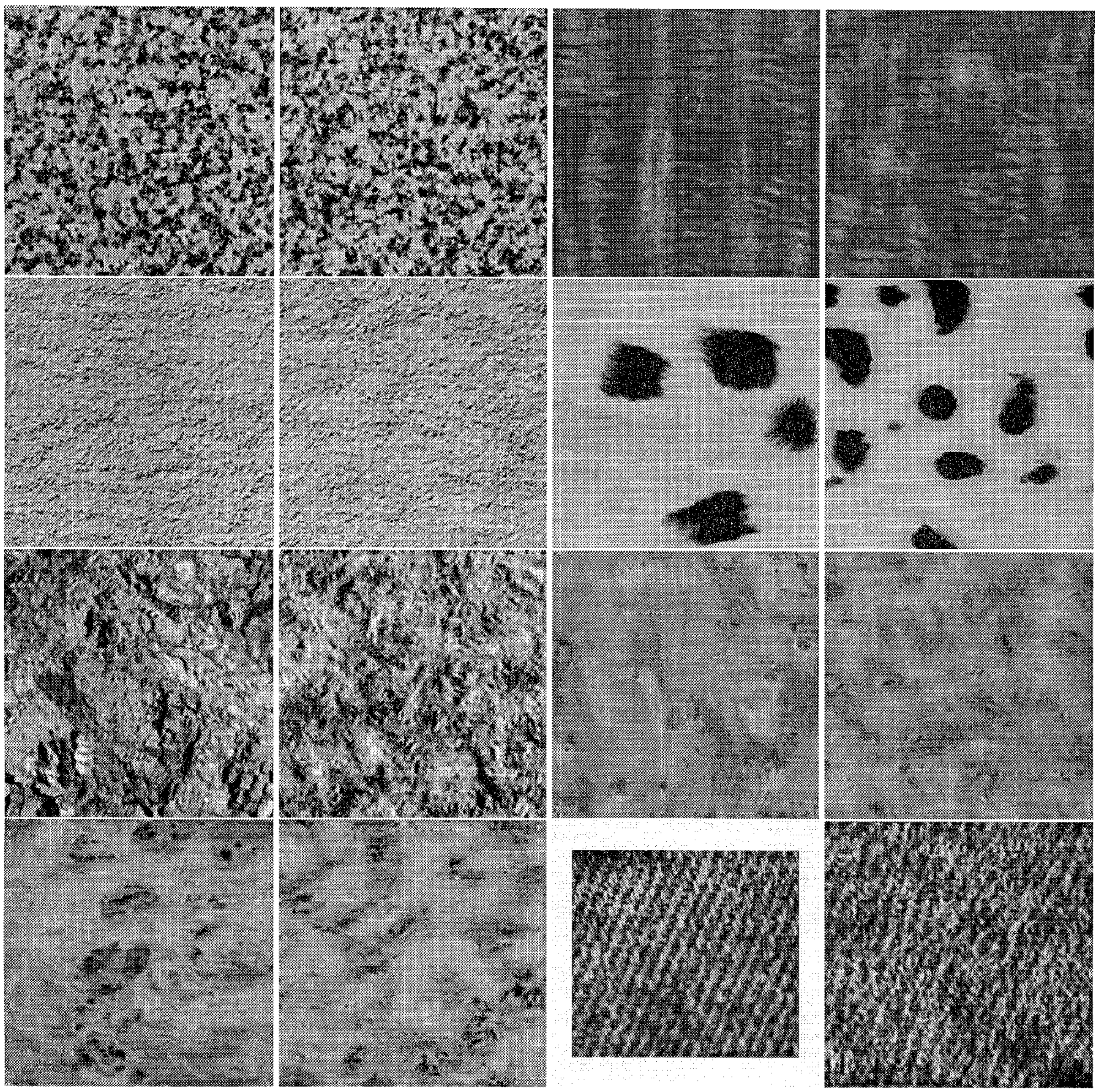

Figure 1: In each pair left image is original and right image is synthetic: granite, figured sepele wood, stucco, panda fur, slag stone, figured yew wood, burled mappa wood, denim.