

# Texture Synthesis for Digital Painting

John Peter Lewis  
Massachusetts Institute of Technology

**Abstract** The problem of digital painting is considered from a signal processing viewpoint, and is reconsidered as a problem of directed texture synthesis. It is an important characteristic of natural texture that detail may be evident at many scales, and the detail at each scale may have distinct characteristics. A “sparse convolution” procedure for generating random textures with arbitrary spectral content is described. The capability of specifying the texture spectrum (and thus the amount of detail at each scale) is an improvement over stochastic texture synthesis processes which are scalebound or which have a prescribed  $1/f$  spectrum. This spectral texture synthesis procedure provides the basis for a digital paint system which rivals the textural sophistication of traditional artistic media. Applications in terrain synthesis and texturing computer-rendered objects are also shown.

CR Categories and Subject Descriptors: I.3.3 [Computer Graphics]: Picture/Image generation; I.3.7 [Computer Graphics]: Three Dimensional Graphics and Realism - color, shading, shadowing, and texture.

Additional Key Words and Phrases: texture synthesis, painting, terrain models, fractals.

## 1 Introduction

The motivation for texture synthesis lies in the observation that the absence of texture is often responsible for the characteristic stark or geometric appearance of computer generated images, and more generally, in the recognition that current image synthesis techniques are inappropriate for rendering many natural phenomena. [Reeves, 1983].

The lack of texture is felt particularly in digital painting. While it appears initially that a “put-that-color-there” paint program (in which any region of the display screen may be shaded with any desired color) is a comprehensive definition of painting, its limitations are severe, as can be seen by the following argument: The number of distinct color regions in a digitized (non-computer) image is generally found to be a significant fraction of the total number of pixels. In a typical medium resolution (250,000 pixel) digitized image there are on the order of 105 distinct color regions, depending on the color resolution of the digitizer and image buffer. This can be visually demonstrated by applying a unique mapping of the pixel values of a digitized image onto relatively incoherent or random values [Figure 1]. Thus, on the order of 105 manual operations would be required to digitally paint a medium resolution image of comparable complexity using a put-that-color-there procedure. As a result many digital paintings incorporate digitized images or use geometrical patterns to achieve visual complexity, while in other cases the jaggie, low resolution appearance resulting from the put-that-color-there procedure provides the “digital” character of the painting.

In contrast much of the detail in traditional media is generated as a desired effect of the painting process. Paints, brushes, and painting surfaces are carefully selected for their characteristic effects. It

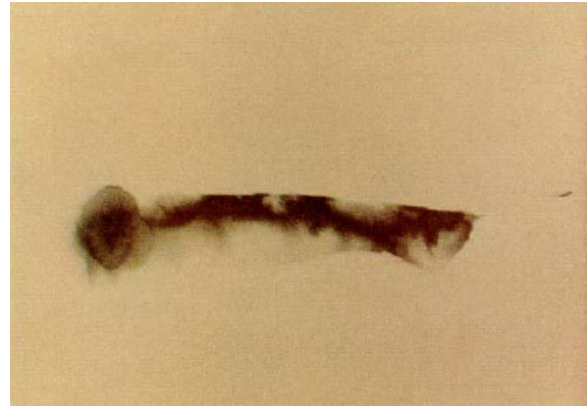


Figure 2: A watercolor brushstroke.

may be said that, whereas the paint program ignores the nuances of the artists gesture to produce a uniform, analytical mark, an expressive (physical) painting medium amplifies the artists gesture. Figure 2 is presented to emphasize that digital painting is far from a “solved problem”, as a single brushstroke in an traditional (and ostensibly primitive) medium produces a quantity and character of textural information which would be difficult to digitally generate.

Computer artists make the point that digital simulation of traditional artistic media is not an appropriate goal. While this is agreed, it is also inappropriate to identify the intrinsic character of computer images with hardware and software primitives which were originally developed to accommodate information display rather than image synthesis. Since this medium is essentially defined in software, organic effects are no less intrinsic than geometrical effects though the former will require new techniques.

The ultimate form of computer-resisted image creation will certainly depart from the painting metaphor. The immediate potential of a programmable painting is nevertheless attractive, remembering that a good illustrator can render an arbitrary scene in the time sometimes required to ray trace relatively simple scenes. As an initial approximation the problem of developing an expressive digital painting medium will be viewed as a problem of texture synthesis. Texture is defined here as object detail which we do not care to explicitly specify though some of its aggregate characteristics may be known. This definition encompasses many natural phenomena as well as the small-scale detail usually associated with the term. The size and location of a geographic feature may be of interest, for example, but it is not desirable to design or measure its surface beyond a certain scale. The textural threshold is the scale beyond which image detail may be replaced by other, similar detail without affecting the viewers perception of subject.

While this is a broad definition of texture, the consideration of texture as an essentially planar phenomenon is restrictive. It will be seen that texture fields may be interpreted as height fields for terrain synthesis, and the development of three-dimensional general-

izations of planar texture synthesis methods is conceivable.

## 2 Texture Models

Texture models trade depth for generality in attempting to reproduce the appearance of a texture without considering its underlying structure. Natural textures exhibit local variations which are modelled as a random process although the responsible phenomena are not random from other viewpoints. A number of stochastic texture models have been proposed and are discussed in the references. A brief description of several important models follows.

Time-series modelling considers periodicities in the row scan of an image; this approach is fundamentally limited in that it cannot easily describe the structure of the texture in the dimension(s) perpendicular to the scan.

Planar random point processes generate a texture consisting of a distribution of points on a planar background. Bombing processes generalize the planar point process by replacing the point primitive with a shape possessing orientation, color and other characteristics [Schacter & Ahuja, 1979].

Cell growth processes partition the plane into cells. The Voronoi tessellation is an exemplary cell grown process which distributes cell nuclei by a point process; a cell is defined as the boundary of the collection of points which are closest to a particular nucleus. The resulting texture resembles natural cellular structures. [Mezei, Puzin, and Conroy].

Syntactic texture models equate tokens of a formal grammar with structural primitives of the texture. A highly structured but non-deterministic texture may be generated if probabilities are assigned to the rewrite or expansion rules of the grammar [Lu and Fu, 1978].

Two-dimensional Markov random field models consider the conditional probability of color values over a sample region of the texture [Hassner and Sklansky, 1980; Cross and Jain, 1983]. The Markovian property is modified by defining the transition probabilities on a neighborhood of adjacent or non-adjacent pixels, so the term describes a finite memory process rather than a planar Markov chain. This method has achieved good results in simulating prototype textures. Unfortunately it is not practical for textures sampled at high resolution since the number of conditional probabilities is  $G^S$  if  $G$  is the number of quantized color levels and  $S$  is the number of adjacent pixels to be considered. Several authors have used Brownian sheets to model rugged terrain [Fournier, Fussell, and Carpenter, 1983; Mandelbrot, 1982]. Though not originally conceived of as a texture model, the height values of the Brownian sheet may be reinterpreted as color values of a planar texture.

## 3 Texture and Scale

An important characteristic of natural texture is that textural detail may occur at more than one scale. Detail may be noticeable at all scales from the textural threshold to the limit of visual activity. For example a field of pebbles has an overall shape; closer inspection shows the contour of individual pebbles, each of which has its own surface texture [Figure 3]. With the exception of the Brownian model of terrain, the texture models mentioned above describe textures with detail at only one scale (this is true of Markov textures because of practical limitations on the size of the region determining the transition probabilities). An overview of these textures affirms that the generating process is stationary and devoid of global

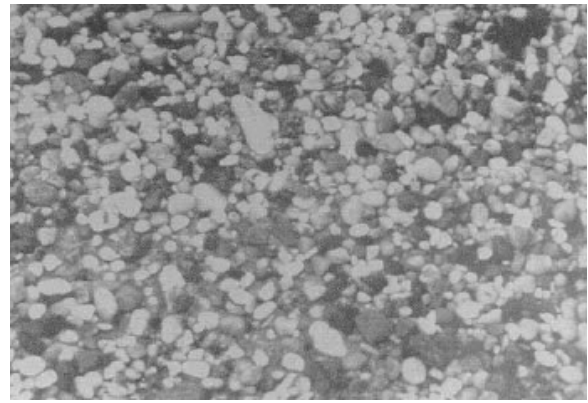


Figure 3: A field of pebbles.

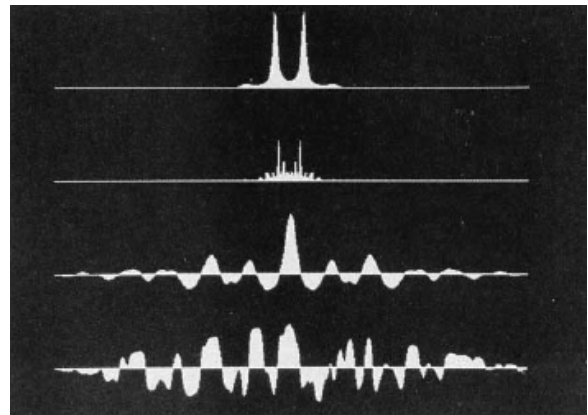


Figure 4: Windowed plot of the grey levels in one row of Figure 3 (bottom), auto correlation, power spectrum, and ten pole autoregressive spectrum estimate obtained by linear prediction (top).

character while close examination can reveal only the constituent pixels of the texture.

The Brownian model of terrain does have the property that closer observation yields more detail. This application of Brownian motion resulted from Mandelbrot's interest in fractals, a collection of mathematical objects united by the criterion of recursive self-similarity: the objects are defined as analytic or probabilistic functions of scale [Mandelbrot, 1982]. Brownian motion can be considered self-similar since the statistical moments of any sample are similar to any other sample, when adjusted by a scaling factor. In fact unmodified Brownian motion generates a very rugged terrain. Mandelbrot has proposed a revised model in which  $B(t)$  is filtered to adjust its amplitude spectrum to  $f^{-q}$  (the unfiltered  $B(t)$  has the parameter  $q=1$ ). The filtering is conceived as a Riemann-Liouville integral of  $B(t)$  [Oldham and Spanier, 1974]: (equation here) (The  $dB(x)$  form of the integral is used to sidestep the problems of formally defining the derivative of  $B(t)$  or the integral thereof using the limit calculus). This is seen to be a convolution of the derivative of  $B(t)$  (white noise) with a filter  $h(t) \propto t^{q-1}$ . The (equation) scaling factor cancels the numerator in the transform of  $h(t)$ , (equation here) (for integer values  $n$ ). When  $q$  is an integer  $RL(b(t))$  reduces to  $q$ -fold repeated integration of  $B(t)$  with the consequent smoothing effect of multiplying the spectrum of  $B(t)$  by  $1/f^q$ . Fractional values of  $q$  produce an intermediate amount of smoothing. (In practice the filtering is carried out by multiplying the spectrum of a Gaussian white noise by the desired exponent and inverse transforming

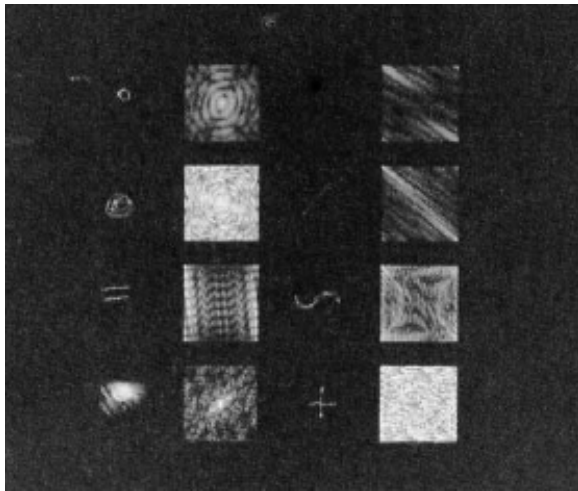


Figure 5: Painted patterns (left) and textures realized by Fourier transformation (right). The lower texture on the right is the phase of the transform while the remaining textures are transform magnitudes.

[Voss, 1983]).

## 4 Frequency Domain Interpretation of Texture

What is important here is that the preceding discussion should suggest the frequency domain interpretation of texture. Consider again the pebbles in Figure 3. The dominant structure defined by the repetition of similarly sized pebbles is reflected by peaks in the power spectrum centered at the frequency which is the inverse of the typical pebble size. [Figure 4]. The attenuated high-frequency portion of the spectrum reflects the relatively smooth surface texture of the pebbles.

This example illustrates the wide bandwidth and varied spectra of natural textures, reflecting the variety of natural structures. Although the fractal  $1/f$  spectra are analytically tractable and have resulted in very attractive computational methods [Fournier, Fussell, and Carpenter, 1982] they are not universal. The following procedure generates random textures with arbitrary spectra.

## 5 Texture Synthesis Procedure

### 5.1 Spectrum Painting

The spectral interpretation of texture suggests the direct specification of the spectrum as input to the texture synthesis procedure, with the texture generated by inverse Fourier transformation. For “organic” textures the spectrum may be as difficult to analytically define as the texture itself. Instead, the desired spectrum is painted using a digital paint program [Figure 5]. (This is a “bootstrap” procedure since the paint program incorporates the textures which it creates).

Two quadrants of the amplitude spectrum are painted and the remaining two quadrants are generated by symmetry. The phase spectrum is painted or set to random values. Inverse transformation gen-

erates a complex texture, with the real and imaginary components having a similar visual character. The magnitude, log magnitude, or phase of the inverse transform may also serve as textures.

Using the near duality between forward and inverse Fourier transforms, it is also effective to implement the spectrum painting procedure by the forward transform from a real, asymmetric pattern. The resulting texture exhibits the quadrant symmetry of a spectrum,  $F(u,v) = F^*(-u, -v)$  which may be undesirable.

Experience with the spectrum painting method shows that it is possible to acquire an intuitive feel for the relation between characteristics of a painted spectrum or pattern and its transform, and the author can reliably paint amplitude spectra to simulate some simple textures including canvas and wood grain. Painting phase spectra is more difficult because the histogram display method produces an apparent discontinuity between two  $\pi$  and zero radians. The method is fairly robust, however, and variations such as setting the phase to zero or at random have produced useful textures.

When the spectrum painting method is not intuitive it nevertheless produces rich and organic textures which would be difficult to paint in the spatial domain using current techniques. One is tempted to suggest an inverse symmetry of visual complexity, by which perceptually simple figures in one domain possess visually sophisticated transforms. This is in part a visual application of the uncertainty principle, which states that a signal and its transform cannot both be “of short duration”. Thus, a pattern consisting of a few non-zero points transforms to a texture which extend over the transform area [Figure 5].

### 5.2 Sparse Convolution

The texture sample realized by spectrum painting has several undesirable properties, however:

1. It is periodic.
2. If standard fast Fourier transform software is used then the texture is contained in a square area whose pixel resolution is a power of two.
3. The texture has well-defined edges which are not characteristic of natural textures.

In the second stage of the texture synthesis procedure a random texture field is produced from the texture sample. The sample is windowed to remove edge discontinuities by multiplying the sample by the radial Gaussian (equation). The texture field is initialized to a constant value and developed by weighted additions of the displaced sample. The weights and displacements are given by a white random number generator. This procedure is equivalent to an out-of-order convolution of the windowed texture sample with a white noise. Thus, the texture sample assumes the role of the filter impulse response or point spread function, and the expected spectrum of the texture field is that of the sample.

In practice excellent results are obtained when the white noise is replaced by a “sparse” (white) noise or point process containing approximately ten non-zero values per point spread area. The out-of-order convolution with a sparse noise results in a significant computational advantage relative to direct convolution. Further savings may be realized if the noise is quantized to integer values.

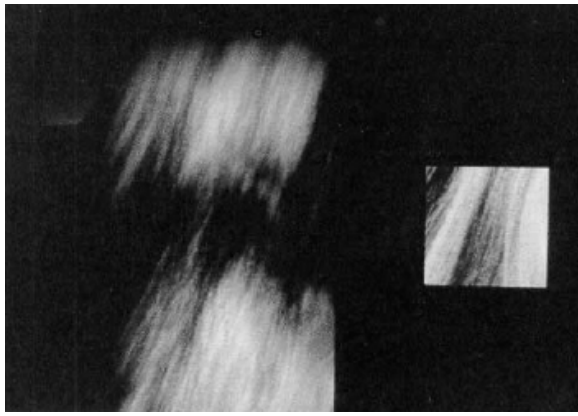


Figure 6: Sampled impulse response (right, from [Bryant, 1978]) and synthetic texture (left).

### 5.3 Sampled Impulse Response Filtering

When it is desired to generate a random texture field which emulates a prototype texture and a sample of that texture is available, we use the result that, if a stochastic process is modelled as a finite impulse response filter excited by white noise, then a properly windowed sample of the process is expected to resemble the impulse response of the filter. The transform of the window should be a narrowband approximation to the delta function since the windowing multiplication (modulation) is equivalent to the convolution of the sample spectrum with the window spectrum. Modest spectral averaging is appropriate since it reduces the local variance of the sample spectrum with respect to "true" (prototype) spectrum; see the discussion of modified periodogram spectrum estimation in a text such as [Beauchamp and Yuen, 1979]. The transform of a Gaussian window is also Gaussian and the bandwidth of the transform is inversely proportional to the size of the window.

Considered in the spatial domain, the window should be large enough to include one or more periods of the lowest frequency components which characterize the texture. For the purposes of painting or image synthesis the window size can be determined by inspection and adjusted if necessary. "As big as possible" is not a good size because spatial convolution is an  $O(\text{size}^2)$  procedure; the unused source texture is put to a better use by averaging several samples in the frequency domain and inverse transforming, thereby reducing the variance of the sample as an estimate of the prototype impulse response.

The sparse convolution procedure is used to produce a random texture field from the windowed sample. While the resulting textures roughly resemble their prototypes (Figure 6), the sampled impulse response procedure is attractive in that it does not require estimation of texture parameters. As such it should be noted that sampled impulse response filtering is a synthesis procedure rather than a texture model.

## 6 Applications

### 6.1 Digital painting

Several variations on these texture synthesis procedures form the basis of a digital paint program [Figures 7, 8]. Sparse convolution allows large, textured areas to be generated rapidly. The painter de-

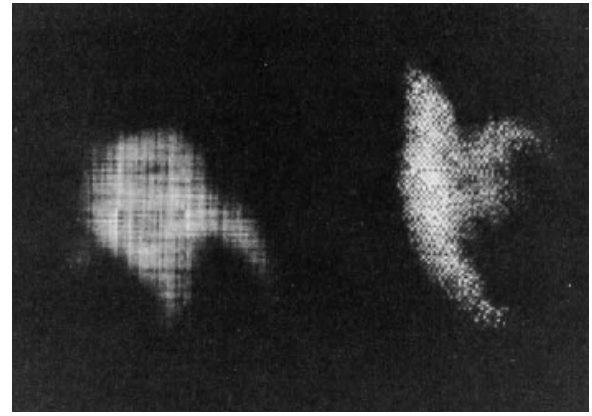


Figure 9: Painting on a virtual surface. Illustration board (left) and sandpaper (right) surfaces are shown.

signs a texture sample using one of the methods described above and specifies the extent, mean, spread, and "sparseness" of the sparse noise. Noise points lying outside the designated extent are discarded. In the current implementation on a nine-bit frame buffer with a twenty-four-bit color lookup table, the resulting texture field is copied into the frame buffer and the lookup table is adjusted to contain a color wash in the portion of the table indexed by the texture field. The texture field merges smoothly with its surroundings because the edges of the texture field inherit the window applied to the texture sample. Many variations on the copying operation are possible, ranging from the linear interpolation

```
pic[x,y] <- pic[x,y] + d * (tex[row, col] - pic[x,y])
```

[Whitted, 1983; Smith, 1982] to a "dry-brush" in which a second texture modulates the density of application of the principal texture. The effect of the texture application then depends on where it is applied (in relation to the second texture) so a virtual surface results. The virtual surface can serve analogously to the textured surfaces of traditional media. The periodic textures resulting from the spectrum painting procedure may be repeated in a mosaic to form an extended, virtual surface. This is implemented simply by indexing the virtual surface texture with the current screen address, modulo the size of the texture.

In the most versatile texture application configurations the texture is used as a weight to interpolate between the background and some other color source, rather than interpolating between the texture and the background. These configurations are described in the "painters raster-op":

```
pic[x,y] <- pic[x,y] + d * tex[row,col] * f(pic[x,y], ...)
```

(the ellipses indicate the presence of additional parameters). For example, if  $f$  is set to  $c - \text{pic}[x,y]$  where  $c$  is a scalar color mixed by the painter, then the texture describes the density or spread function of a "textured airbrush". In an eight-bit or nine-bit frame buffer it is practical to precompute the function  $f$  and store it in a table indexed by  $\text{pic}[x,y]$ . The table is recomputed when the auxiliary parameters change (this does not happen more than once per texture application). The effect of the painters raster-op depends on the contents of the target area as well as on the applied texture, so each texture application builds upon the existing texture rather than replacing it.

In addition it is often useful to reference the texture through a shaping table  $v[t]$ . The default value of the table is  $v[t] = t$ , making the

table invisible. By altering the slope of the table function as  $v[t] = a*t + b$  linear compression and negation are achieved. Rectification, clipping, power modification, and non-linear filtering are obtained without additional programming by using appropriate shaping functions.

This is a cheap and powerful filtering method. It is similar to color lookup table techniques but it may be applied locally and its effect depends on the current texture application configuration. The spectral shaping effects and design procedures for this method are developed in [LeBrun, 1979]. In keeping with the intuitive approach used in the texture synthesis procedure, the user draws the shaping function using a digitizing tablet and evaluates it interactively. A painting is created by using the texture synthesis procedure as a brush with a definable shape and textural properties. The size of the synthesis region is reduced in order to paint details. If the specified size is smaller than the size of the texture sample then the texture sample is applied in place of the synthesized texture. In this case the "brushstroke" (applied texture) is no longer random and the repeated application of the texture sample may be evident.

This is usually not a serious problem, since in most painting styles the semantic detail (detail explicitly specified by the artist) occurs at a much larger scale than the process texture. Several approaches to the problem of creating small, random textures (on the order of ten square pixels) have been attempted, however.

One approach is to synthesize the texture with an all-pole filter driven by noise. The all-pole filter can predict a small area of texture provided that an adjacent area of support or filter memory has been obtained. In our application the filter memory is set to the neighboring areas of the painting (if not zero) or to the texture prototype or sample. The filter is designed by a linear prediction fit to the sample. The linear prediction method is very attractive in that it can monitor the filter stability (a common problem in the design of all-pole filters) and in that it accomplishes the filter design task [Makhoul]. This would be an ideal method if the filter were implemented in hardware, but it is rather slow for interactive painting when implemented in software.

A second approach is to precompute some of the desired texture and obtain a "brushfull" of it by positioning a window corresponding to the size of the brush. If the window is moved with each texture application then each "brushstroke" is distinct but characteristic of the desired texture.

The success of texture synthesis in painting is in part indicated by the statistic that Figures 7,8 were created using between 2000 and 10,000 points entered from a digitizing tablet, while each picture contains approximately 100,000 separate color regions. The input streams from the digitizing tablet and keyboard may be recorded to produce a relatively compact representation of a painting. A more controversial indication of success is that these paintings do not suggest their computer origin, and they have been mistaken for digitized images.

## 6.2 Terrain synthesis and texture mapping

Synthetic textures may be interpreted as height fields to create a variety of terrain types. Figure 10 is reminiscent of a terrain shaped by erosion, while Figure 11 resembles a filtered fractal surface with the qualification that it has a modest directional trend. It is interesting to note that if the texture sample is a step function then the out-of-order convolution procedure reduces to a variant of the Levy faulting process which was used to implement the first Brownian terrain simulations [Mandelbrot, 1977]. The preservation of the sample spectrum in the texture field is evident as both have a 1/f

(amplitude) spectrum.<sup>1</sup> Recently Haruyama and Barsky have similarly considered the fractal 1/f type spectra as an instance of a more general spectral approach to texture [Haruyama & Barsky, 1984]. Their paper adapts the attractive recursive-subdivision fractal construction to produce non-fractal textures with a variety of possible spectra.

The periodic textures resulting from inverse Fourier transformation are useful in texture mapping computer-rendered objects since they may be repeated to cover large areas without producing the border discontinuities which are visible in the non-periodic texture map used in Figure 8.

## 7 Conclusions

The frequency domain interpretation of texture motivates the development of a spectral texture synthesis method which can generate synthetic textures having the broad bandwidth and irregular spectra characteristic of natural textures. An interactive procedure for generating textures with arbitrary spectra was presented. This procedure has been used to realize some of the potential of digital painting as a software medium possessing few inherent limitations or characteristics. The textural complexity of the resulting images approaches that obtainable in traditional artistic media.

## References

1. Beachamp, K. and Yuen, C. Digital Methods for Signal Analysis. Allen and Unwin, London, 1979.
2. Brodatz, P. Texture. Dover, New York, 1966.
3. Bryant, P. in Playboy 25, 4 (April 1978), 118-127.
4. Cross, G. and Jain, A. Markov random field texture models. IEEE Transactions on Pattern Analysis and Machine Intelligence 5, 1 (1983), 25-38.
5. Fournier, A., Fussell, D., and Carpenter, L. Computer rendering of stochastic models. Communications of the ACM 25, 6 (1982), 371-384.
6. Haruyama, S. and Barsky, B. Using stochastic modelling for texture generation. IEEE Computer Graphics and Applications March, 1984 7-19.
7. Hassner, M. and Sklansky, J. The use of Markov random fields as models of texture. Computer Graphics and Image Processing 12, 4 (1980) 357-370.
8. LeBrun, M. Digital waveshaping synthesis. J. Audio Engineering Society 27, 4 (April 1979), 250-265.
9. Lu, S. and Fu, K. Computer generation of texture using a syntactic approach. Computer Graphics 12, 13 (1978), 147-152.
10. Makhoul, J. Linear prediction: a tutorial review. Proc. IEEE 63, (1975) 561-580.
11. Mandelbrot, B. Fractals-Form, Chance, and Dimension. Freeman, San Francisco, 1977.
12. Mandelbrot, B. The Fractal Geometry of Nature. Freeman, San Francisco, 1982.
13. Mezei, L., Puzin, M., and Conroy, P. Simulation of patterns of nature by computer graphics. Information Processing 74, 52-56.
14. Oldham, K. and Spanier, J. The Fractional Calculus. Academic Press, New York, 1974.
15. Reeves, W. Particle systems - a technique for modelling a class of fuzzy objects. Computer Graphics 17, 3 (July 1983), 359-376.
16. Schachter, B. and Ahuja, N. Random pattern generation processes. Computer Graphics and Image Processing 10, 2 (1979), 95-114.
17. Smith, A. Paint, in Tutorial: Computer Graphics, J.C. Beatty and K.S. Booth eds. (IEEE 1982), 501-515.
18. Voss, R., Fourier synthesis of Gaussian fractals: 1/f noises, landscapes, and flakes. Presented at ACM Siggraph Conference, 1983.
19. Whitted, T. Anti-aliased line drawing using brush extrusion. Computer Graphics 17, 3 (July 1983), 151-156.

<sup>1</sup>In the two-dimensional case the step is randomly rotated before each weighted addition to create an isotropic 1/f texture. In general the rotation of the texture sample is avoided in order to achieve textures with directional characteristics.



Figure 8: The background in this picture is painted while the stone figures are a texture-mapped computer rendered object. The edge of the (non-periodic) texture map is visible as a horizontal seam on the left figure; this was retained for an esthetic reason.

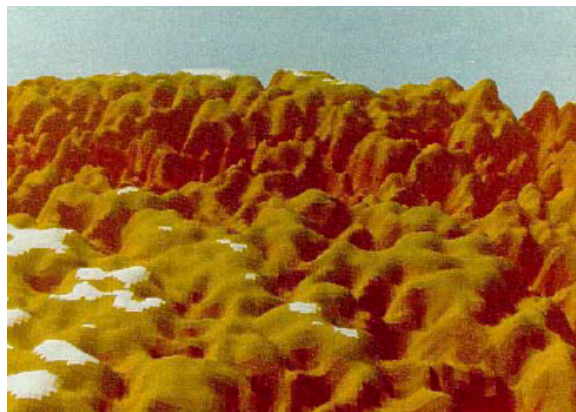


Figure 10: A synthetic terrain generated from a lumpy texture sample.

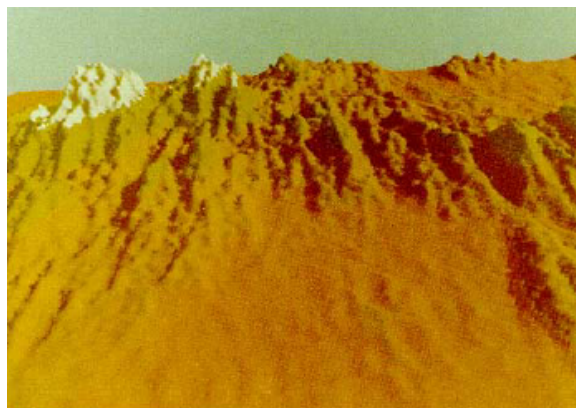


Figure 11: A synthetic terrain resembling a fractal surface. A directional characteristic ("north-south ridges") is evident.

Permission to copy without fee all or part of this material is granted provided that the copies are not made for or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission. (c) 1984 ACM 0-89791-138-5/84/007/0245 \$00.75

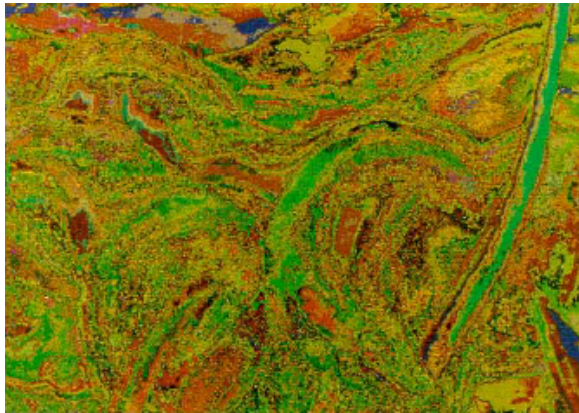


Figure 1: Random false-coloration of a digitized painting. The number of distinct color regions is a large fraction of the total number of pixels.

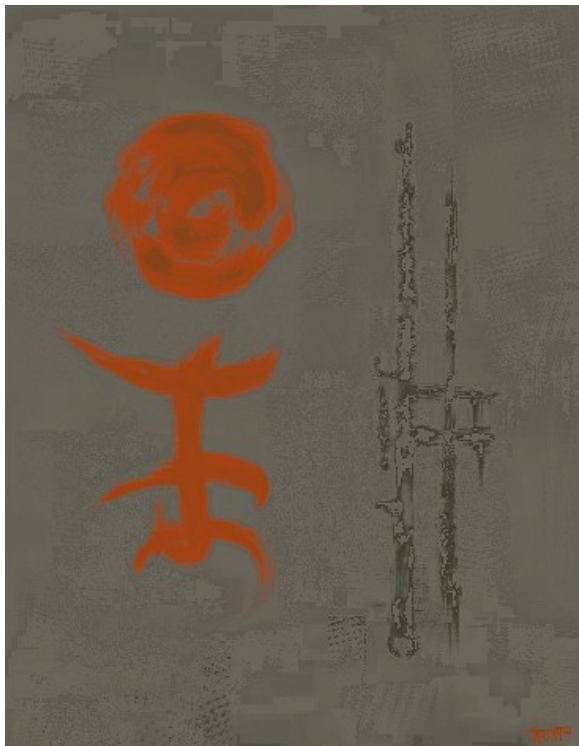


Figure 7: A digital painting incorporating synthetic textures.