Stereological Techniques for Solid Textures

Robert Jagnow* MIT
Julie Dorsey† Yale University
Holly Rushmeier‡ IBM Research

Abstract

We describe the use of traditional stereological methods to synthesize 3D solid textures from 2D images of existing materials. We first illustrate our approach for aggregate materials of spherical particles, and then extend the technique to apply to particles of arbitrary shapes. We demonstrate the effectiveness of the approach with side-by-side comparisons of a real material and a synthetic model with its appearance parameters derived from its physical counterpart. Unlike ad hoc methods for texture synthesis, stereology provides a disciplined, systematic basis for predicting material structure with well-defined assumptions.

CR Categories: I.3.7 [Three-Dimensional Graphics and Realism]: Color, shading, shadowing, and texture— [I.3.3]: Picture/Image Generation—Viewing algorithms

Keywords: stereology, texture synthesis, solid textures, volumetric textures, procedural textures, spatial sampling theory

1 Introduction

Many real objects exhibit complex spatial variation in their surface color and finish. To generate synthetic objects with a comparable, realistic appearance, the area of texture synthesis has been extensively explored in computer graphics [Ebert et al. 1994]. Recently, a number of authors have directed their attention toward synthesizing textures on 3D object surfaces based on representative 2D images [Turk 2001; Gorla et al. 2001]. By using physically occurring input textures, these algorithms can often produce a rich, natural object appearance.

Two-dimensional textures [Blinn and Newell 1976] or small-scale geometric textures parameterized in 2D (e.g., bidirectional texture functions, bumps, displacements) are effective for applying a natural appearance to objects with inherently 2D coverings, such as paint, skin, fur, or mechanically roughened surfaces. Gardner [1984], Peachey [1985], and Perlin [1985] introduced the idea of 3D solid textures to represent objects with surface properties that result from being cut out of a 3D spatially varying material.

For many years procedural techniques have proved useful for the artistic generation of 3D solid textures. However, 3D procedural shaders are often highly parameterized with nonintuitive inputs that can make it difficult, even for a talented artist, to match the appearance of a physical sample. Similar to 2D texture synthesis, it is desirable to generate 3D solid textures directly from physical samples. However, obtaining a fully 3D solid texture sample is far more difficult than obtaining a 2D texture sample.

*e-mail: rjagnow@alum.mit.edu
†e-mail: dorsey@cs.yale.edu
‡e-mail: holly@acm.org

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or direct commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 1515 Broadway, New York, NY 10036 USA, fax +1 (212) 869-0481, or permissions@acm.org.
© 2004 ACM 0730-0301/04/0800-0329 $5.00

Figure 1: A synthetic image, rendered with the solid textures generated by our algorithm.

In this paper we demonstrate the use of techniques from traditional stereology developed in the fields of biology and material sciences [Hagwood 1990; Underwood 1970] to generate 3D solid textures from physically captured 2D images. The result is a collection of methods applicable to the class of solid textures composed of particles distributed in a binding medium. This class includes man-made building materials such as concrete aggregates, asphalt, and terrazzo, naturally occurring materials such as igneous rock, and materials that exhibit discrete volumetric voids, such as sponges and foams.

While the class of solid textures we consider in this paper is restricted, by drawing on stereology as developed in other fields we add to the existing array of tools for extracting 3D information for computer graphics applications. Furthermore, since stereology has been developed as a tool for quantitative analysis, it has well-defined assumptions and a rigorous mathematical basis. This allows for the generation of reliable, precise solid textures for computer graphics applications.

2 Previous Work

A wide variety of 3D procedural texturing methods have been developed over the years, but relatively few are based on physical data [Dischler and Ghazanfarpour 2001; Wei 2001]. One approach to using physical data is Heeger and Bergen's pyramid-based texture analysis and synthesis [1995]. An initial 3D noise distribution is modified so that the histogram of each frequency band matches the histogram of the corresponding frequency band in a 2D image. In the same spirit, Dischler et al. [1998] use a spectral analysis of orthogonal images of a physical 3D volume and iteratively alter the 3D noise distribution to match the statistics of the original images. This allows their method to capture aspects of anisotropic solids

such as wood and marble. Both methods work well for a subclass of common natural textures, but are unable to capture material structure composed of discrete particles.

Recently Markov Random Field (MRF) algorithms have received a great deal of attention for generating 2D textures [Wei and Levoy 2001]. If a fully 3D solid texture sample is available, using a 3D extension to these algorithms is natural. Generating 3D textures from 2D samples with MRFs is not as straightforward. Wei [2001; 2003] describes an MRF approach using multiple 2D images to synthesize a 3D texture. This approach is successful for some texture classes, but it fails to accurately characterize the 3D distribution of macroscopic particles.

Lefebvre and Poulin [2000] successfully generate 3D wood textures from 2D images by analyzing an input image to obtain specific parameters for a procedural shader. However, this approach does not generalize for other classes of solid textures.

Dischler and Ghazanfarpour [1999] discuss the problem of generating solid textures with macroscopic structure. They describe a technique to synthesize natural particle shapes to be embedded in a 3D volume. The design of the particle shape begins with a scanned cross-section of a physical particle. However, the proposed approach does not describe how to capture the full structure of an existing material by estimating the particle size distribution.

Solid textures are also of interest in the material and biological sciences. A precise quantitative characterization of heterogeneous materials is needed to study structures that are built or grown from these materials [Underwood 1970; Howard and Reed 1998]. Since obtaining full three-dimensional samples of such solids is an expensive and time-consuming process, the discipline of stereology was developed to infer 3D distributions from 2D samples. With the advent of digital imagery, image analysis and stereology are frequently used in conjunction for a variety of applications [Wojnar 2002]. In this paper we demonstrate the application of some of the fundamental techniques of stereology to computer graphics solid texture synthesis.

3 Estimating 3D Distributions

An important observation in stereology is that the macroscopic statistics of a 2D image are related to, but not equal to the statistics of a 3D volume. In this work, we present a disciplined approach to recovering 3D volume parameters using methods motivated by spatial sampling theory. We begin by demonstrating the approach with a distribution of spheres, and then extend the approach to work with arbitrary particle types.

3.1 Distributions of Spheres

To illustrate the process, we first consider a 3D distribution of spherical particles having a maximum diameter of d_{max}. A 2D slice through the volume results in circular profiles, also having a maximum diameter of d_{max}. Our objective is to establish a relationship between the size distribution of 2D circles, expressed as the number of circles per unit area, and the size distribution of 3D spheres, expressed as the number of spheres per unit volume. This process is known as "unfolding". Our approach is most similar to that proposed by Saltikov [1967].

For any distribution of identical convex particles, particle density, N_V, is related to the profile density, N_A, by the fundamental relationship of stereology [Underwood 1970]:

$$N_A = \bar{H} N_V \tag{1}$$

where $\bar{H}$ is the mean caliper diameter of the particle, i.e., the distance between tangent planes averaged over all orientations of the particle. For spheres, $\bar{H}$ is simply the diameter.

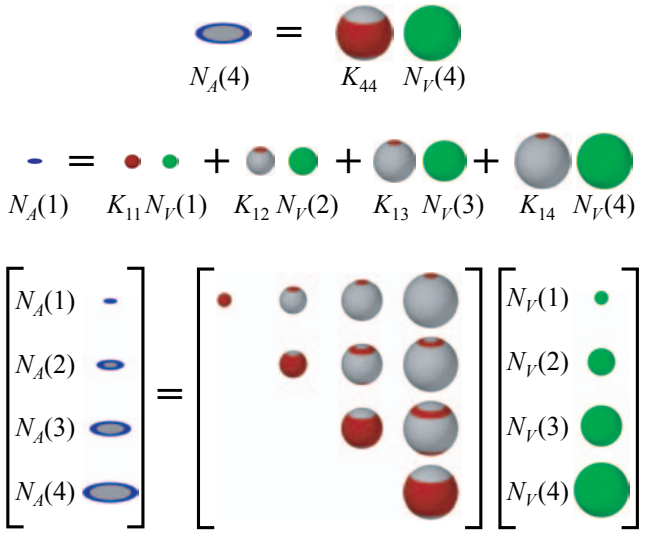


Figure 2: In this set of three equations, the blue disks represent profile densities, $N_A(i)$, the green spheres represent particle densities, $N_V(i)$, and the red and white spheres represent the probabilities that a sphere of a given size appears with a particular profile size. The first two expressions are used to calculate the densities of the largest and smallest profile sizes respectively. These and the remaining density computations are expressed in the matrix equation.

For most aggregate volumes, it is unlikely that the particles will all be of the same size, so we instead use a histogram approach that is common to a number of stereological algorithms. We group both particles and profiles according to their diameter into n evenly sized bins. Spherical particles are clustered according to their diameter to yield particle densities $N_V(i), \{1 \le i \le n\}$. In a random 2D slice through the volume, circular profiles are similarly clustered according to their diameter to yield profile densities $N_A(i), \{1 \le i \le n\}$.

The densities N_V and N_A are related by the values K_{ij}, which express the relative probabilities that a sphere in the jth histogram bin with diameter j/n, exhibits a profile in the ith histogram bin with diameter $(i-1)/n < d \le i/n$. Profiles of the largest size, $N_A(n)$ can only result from slices near the equator of the largest spheres, $N_V(n)$. This relationship is visually represented at the top of Figure 2 for $n = 4$ and $d_{max} = 1$. In contrast, profiles of the smallest size, $N_A(1)$, can result from a slice near the poles of a sphere of any diameter, as expressed in the second equation in Figure 2. The complete density vectors N_V and N_A are related by the expression

$$N_A = d_{max} K N_V \tag{2}$$

The corresponding visual representation is shown at the bottom of Figure 2. Spheres can only exhibit profiles of equal or smaller diameter, so K is an upper-triangular matrix where

$$K_{ij} = \begin{cases} \frac{1}{n}\left(\sqrt{j^2-(i-1)^2} - \sqrt{j^2-i^2}\right) \text{ for } j \ge i \\ 0 \text{ otherwise} \end{cases} \tag{3}$$

Given this relationship, if we know the profile density distribution N_A, we can solve for the particle densities N_V as

$$N_V = \frac{1}{d_{max}} K^{-1} N_A \tag{4}$$

Since K is an upper-triangular matrix, its determinant is the product of the diagonal elements—all of which are nonzero. Thus, $|K|$ is nonzero, and K is guaranteed invertible.

3.2 Distributions for Other Particles

For a nonspherical particle P, we cannot easily classify the profile size according to its diameter, so we need a different metric. We have chosen to use $\sqrt{A/A_{max}}$, where A is the area of the profile and A_{max} is the largest encountered area of any profile. The profile area can be easily and reliably measured in digital images simply by counting pixels. Taking the square root results in values that tend to be more evenly distributed among equally sized histogram bins—a property that is important for minimizing numerical error. This also establishes a linear relationship between the profile measure and the particle scale. In contrast, prior authors have instead opted to categorize profiles according to A/A_{max}, but used a nonlinear scale for histogram bins [Saltikov 1967; Underwood 1970]. Note that classifying profiles by $\sqrt{A/A_{max}}$ is equivalent to classifying spherical particles by d/d_{max} as was done in Section 3.1.

As with spherical particles, we must compute a matrix K to relate particle size to profile size. This relationship can be expressed as

$$N_A = \bar{H} K N_V \tag{5}$$

where $\bar{H}$ is the mean caliper diameter of particle P. Each matrix entry K_{ij} represents the *normalized* relative probability that a particle in column j exhibits a profile in row i. More explicitly, particles in column j are scaled uniformly by j/n, and profiles in row i have a classification value $\sqrt{A/A_{max}}$ between $(i-1)/n$ and i/n. We refer to these probabilities as normalized in the sense that the probabilities in the final column of K sum to 1, and for each column j,

$$\sum_{i=1}^{n} K_{ij} = j/n \tag{6}$$

Note that if only one histogram bin is used ($n = 1$), then Equation 5 reduces to the fundamental relationship of stereology (Equation 1).

For an arbitrary particle P, represented as a watertight polygon mesh, it may be difficult to compute the K matrix analytically. To compute these statistics, we use a Monte Carlo routine that takes advantage of the speed of modern graphics hardware.

To compute a cross-sectional area of particle P, the polygon mesh is assigned a random orientation and rendered such that the near clipping plane of an orthographic camera cuts through the particle at a random depth. As the mesh is rendered, the stencil buffer counts how many times each pixel is touched during rasterization. Odd values indicate that a pixel is inside the cross-section; even values denote pixels outside the cross-section. Thus, calculating the area is a simple matter of summing the odd-valued pixels in the stencil buffer. The resulting area calculations are used to populate the histogram for each column of the K matrix. This process must keep track of the maximum encountered profile area, A_{Pmax}, and can also be used to compute the mean caliper diameter of P, $\bar{H}_P$.

If a slice through a non-convex particle results in two or more disjoint profiles, as shown in the top slice in Figure 3, then each disjoint region should be considered separately, and each will contribute to the histogram construction.

For each particle type we tested, this process converged to a residual of $< 0.5\%$ for each histogram bin within 100,000 iterations. Computation time was less than four minutes. Some example statistics for simple particles are shown in Figure 4.

Before this data can be used in our stereological calculations, we must compute a scale factor s to relate the size of particle P to the size of the particles seen in our input image. Suppose the image exhibits profiles with maximum area A_{img}. This is equal to the maximum profile of P, if scaled uniformly by

$$s = \sqrt{A_{img}/A_{Pmax}} \tag{7}$$

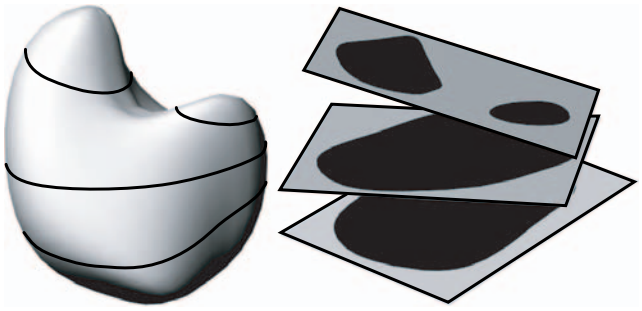


Figure 3: The K matrix for an arbitrary particle is constructed by calculating the cross-sectional area of random slices through the volume.

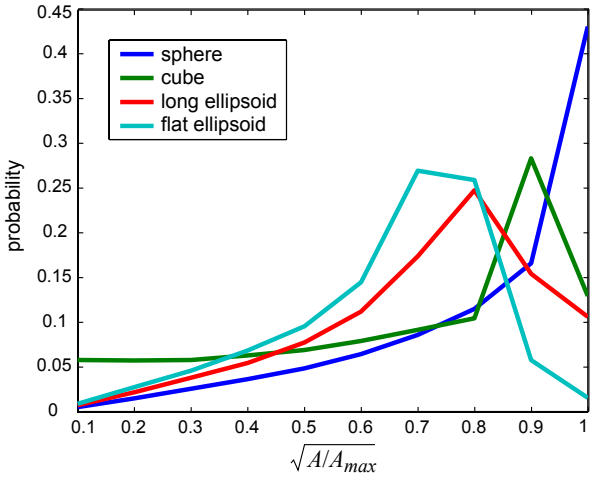


Figure 4: Likelihood of cross-sectional area for simple particle types.

This scale factor is used to calculate the mean caliper diameter $\bar{H} = s\bar{H}_P$, which is used in Equation 5.

Finally, if we compute the profile densities N_A from the input image, we can solve for the particle densities N_V as before:

$$N_V = \frac{1}{\bar{H}} K^{-1} N_A \tag{8}$$

3.3 Managing Multiple Particle Types

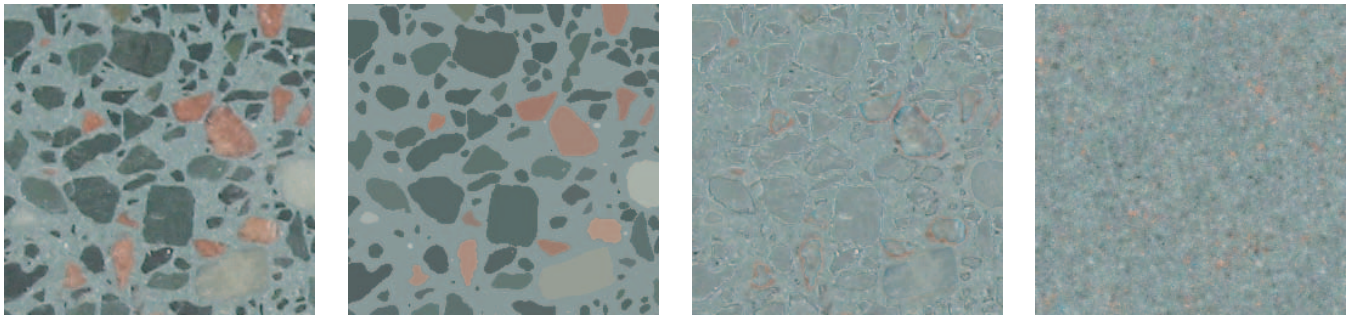
In many instances, a volume may exhibit more than one type of particle. In this case, each particle type i will have its own mean caliper diameter $\bar{H}_i$, representative matrix K_i, and distribution N_{Vi}:

$$N_A = \sum_i \left(\bar{H}_i K_i N_{Vi}\right) \tag{9}$$

If we assume that each particle type exhibits the same distribution—i.e., particle type and size distribution are uncorrelated—then this can be reexpressed as follows:

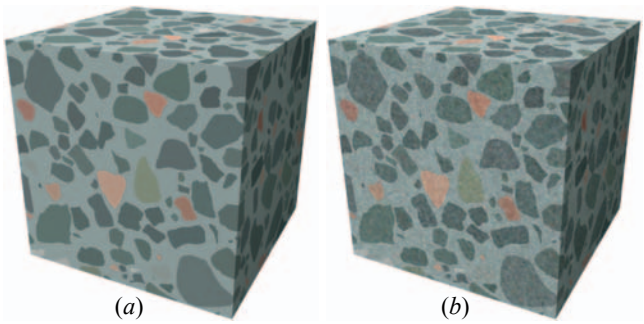
$$N_A = \sum_i \left(\bar{H}_i K_i P(i) N_V\right) \tag{10}$$

$$= \sum_i \left(\bar{H}_i K_i P(i)\right) N_V \tag{11}$$



(*a*) (*b*) (*c*) (*d*)

Figure 6: The mean color value for each profile (*b*) can be subtracted from the input image (*a*) to yield a residual (*c*). Here, the residual has been recentered around the color of the binding material for clarity. This residual lacks the macroscopic structure of the input, and can be resynthesized as a 3D volume [Heeger and Bergen 1995] (*d*).



(*a*) (*b*)

Figure 5: A cube of synthesized material, colored using mean profile colors (*a*) and by adding a 3D noise function (*b*).

where $N_V = \sum N_{Vi}$ is the total particle density, and $P(i)$ is the probability that a particle is of type i. This allows us to solve for the particle densities N_V as

$$N_V = \left[\sum_i \left(\bar{H}_i K_i P(i)\right)\right]^{-1} N_A \tag{12}$$

4 Reconstructing the Volume

Once the particle densities N_V have been recovered, a volume can be constructed to match the appearance of the input image. The reconstruction process establishes particle positions and colors, as well as a residual noise function to add fine details characteristic of the input.

4.1 Annealing

The synthetic volume is populated according to the density distribution N_V such that the largest particle P in the aggregate is scaled uniformly by s from Equation 7. The naive approach for populating the volume is to add one particle at a time, randomly testing orientations and translations until sufficient vacant space is found. Unfortunately, this method fills space inefficiently and works only for loosely packed volumes. Instead, we populate the volume with all of the particles, ignoring overlap, and then perform simulated annealing to resolve collisions. This method repeatedly searches for all collisions and then relaxes particle positions to reduce interpenetration.

The annealing process considers the volume to repeat in the x, y, and z directions so that the resulting volume can tile seamlessly in space. If the annealing process pushes the center of a particle outside of the volume in one direction, then that particle is moved to the opposite side of the volume; thus, the global density of the particles cannot be altered. In practice, visual repetition is only noticeable in rendered images if the texture volume is exactly aligned with a large planar face. This can be avoided with a simple rotation of the texture volume.

4.2 Color

If particle size and color are uncorrelated, then each particle can be assigned the mean color of a randomly chosen profile from the input image. Similarly, the binding material can be assigned the mean color of all non-profile pixels in the input image. An example result is shown in Figure 5(*a*).

If particles of different sizes exhibit different colors, then distinguishable color categories can be automatically identified by applying the k-means clustering algorithm to the set of mean profile colors. The stereological analysis process can then be applied to the profiles in each color category, and the combined results can be used to populate a synthetic volume.

4.3 Adding Fine Details

As can be seen in Figure 5(*a*), using the mean color for each particle produces an unsatisfying result as it fails to capture color variations at sub-particle scale. To replicate the input appearance, we start by subtracting the mean color values of each profile—Figure 6(*b*)—from the original input (*a*) to obtain a residual (*c*). Residual values for each pixel can range from -1 to 1 in each color channel. The images shown here have been recentered around the color of the binding material for clarity. The residual lacks the structure of the original input and responds well to the application of Heeger and Bergen's synthesis algorithm [1995] in three dimensions (*d*). This newly synthesized volume of texture can then be added to the mean color values to obtain the result shown in Figure 5(*b*), which exhibits both the structure and the characteristic noise frequencies of the input.

The residual volume should be synthesized to match the pixel scale of the input image. Like the particle volume, the residual volume can be synthesized to allow for seamless repetition in the x, y, and z directions. Thus, the dimensions of the residual volume do not need to match the dimensions of the particle volume.

Attempts to estimate noise distributions for individual particles were largely unsuccessful due to the insufficient sample size of the

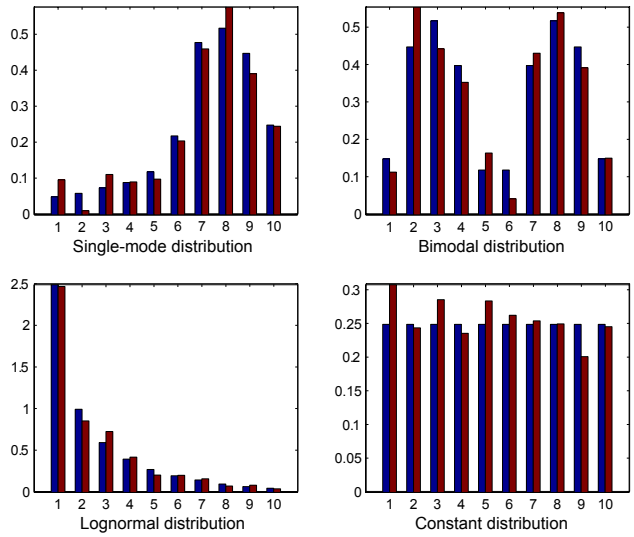


Figure 7: Performance of the density recovery algorithm on a variety of input distributions. Each graph shows 3D particle densities grouped into ten histogram bins. Actual distributions are shown in blue and estimated distributions in red.

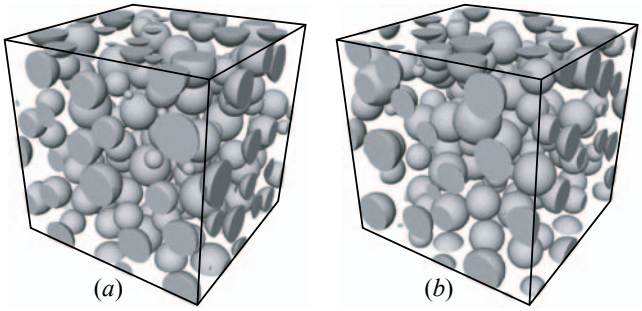


Figure 8: Comparison of the single-mode volume of spherical particles (*a*) and a comparable volume obtained via the density recovery algorithm (*b*).

input profiles. Furthermore, we discovered that applying different noise functions to individual synthetic particles resulted in sharper visible boundaries than appear in the input images.

5 Results

To evaluate the accuracy of the algorithm, we tested the process on a series of synthetic volumes, a physical volume with known parameters, and several physical datasets with unknown parameters. These results and corresponding analysis are discussed in the remainder of this section.

5.1 Synthetic Volumes

To test the robustness of the algorithm under various conditions, we analyzed a number of different synthetic distributions. In each case, a synthetic volume was populated with spherical particles, and an analysis was performed by counting the visible profiles in ten equally spaced slices through the volume. Figure 7 shows the results of our algorithm applied to single-mode, bimodal, lognormal, and constant distributions. These results were based on between 1050 and 1400 profile observations, grouped into 10 evenly sized

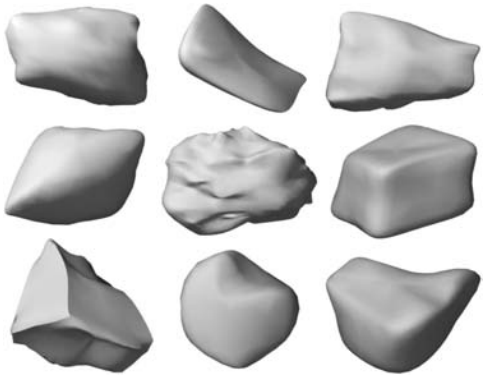


Figure 9: A collection of particle shapes used by the solid texture algorithm.

histogram bins. Figure 8 illustrates a side-by-side comparison of a small subregion of the single-mode volume and a comparable region in a volume generated with the recovered density values.

5.2 Working with Physical Data

For the synthetic volumes described above, we benefit from being able to obtain an exact profile count and from knowing the exact particle shapes *a priori*. In contrast, when working with physical data, we cannot predict exact particle shapes, we are often unable to count small profiles, and we are often limited to fewer profile observations. Each of these introduces potential sources of error into our calculations.

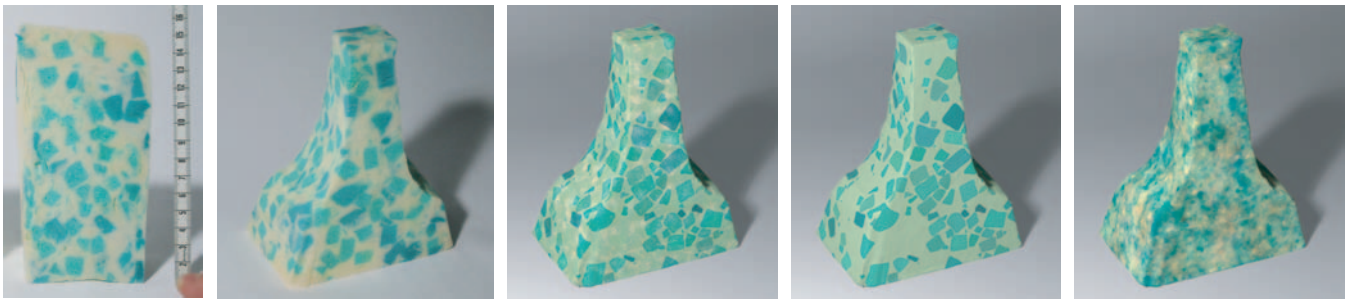
The problem of reconstructing a particle shape from a representative 2D slice is insoluble without additional information. Unless a full particle can be extracted from the volume, particle shapes need to be estimated. For the results shown here, particles were created by manually editing the control vertices of a NURBS sphere until the desired shape was achieved. Example particle models are shown in Figure 9. Only one or two particle shapes were used in each data set.

Errors in the volume density recovery process are typically manifest as either dramatically different densities in adjacent histogram bins or negatively populated histogram bins. If only a few profiles have been observed in one or more of the profile histogram bins, then numerical errors should be expected. This problem can be reduced simply by decreasing the number of histogram bins that are used for the calculations.

Negative estimates in the recovered volume histogram are particularly likely for the bins representing the smallest particles. It should be expected that small profiles may be obscured by noise or may be removed completely from the volume by the sample preparation process. This problem of "missing fines" is addressed in prior publications [Keiding and Jensen 1972; Maerz 1996]. These underrepresented profiles may, in turn, result in negative estimates for small particle densities. It should not necessarily be considered an undisciplined approach to clamp these values to zero.

5.3 Test Volume

In order to test the algorithm on physical data under controlled conditions, we constructed a volume with known particle shape and distribution. Part of the volume was sliced into planar regions, as shown in Figure 10(*a*), and the profiles were counted to estimate the profile density distribution. The remainder of the volume was carved into an abstract shape and scanned with a 3D turntable scanner. Finally, a synthetic volume was rendered using the density



(*a*) (*b*) (*c*) (*d*) (*e*)

Figure 10: To test the algorithm under controlled conditions, we constructed and measured a volume with known particle shape and distribution (*a*). An abstract shape carved from the physical volume (*b*), is synthetically replicated using the color, noise, and distribution parameters recovered by our algorithm (*c*). For comparison, we also show the synthetic volume with mean color values only (*d*) and a synthetic replica of the volume rendered using Heeger and Bergen's algorithm [1995] (*e*).

values recovered by the stereological analysis process. Figure 10 shows a side-by-side comparison of the actual volume (*b*) and the synthetic rendition (*c*). Also shown for comparison are the synthetic volume rendered with mean color values only (*d*), and a synthetic volume textured by applying Heeger and Bergen's algorithm [1995] to the initial input texture (*e*). This final image effectively captures much of the color and frequency information, but fails to capture the discrete structure of the input.

5.4 Physical Inputs

Figure 12 shows some results of the algorithm. In each image pair, the input texture is shown on the left and the corresponding synthetic result on the right. In each case, the residual volume is synthesized at a resolution of 128x128x128 voxels in less than three minutes. The most time-consuming part of the process is the volume annealing, which can require anywhere from a few minutes to an hour, depending on the particle density, size, and complexity. To reduce the number of potential collisions that must be checked, we hash the particles into a regular grid at each iteration of the annealing process. Annealing times can also be improved by substituting simpler proxy meshes for complex particle shapes.

Figures 1 and 11 show synthetic scenes rendered using solid textures recovered by our algorithm.

6 Conclusions

The methods described in this paper expand the class of 3D solid textures that can be synthesized from 2D photographs. More specifically, we introduce methods that operate on textures with a discrete macroscopic structure. We draw techniques from a class of existing literature that offers a number of additional synthesis tools for graphics. The statistical approach provides a sound basis for synthesizing material distributions with well-defined assumptions to perform accurate predictive rendering.

Future work includes automated estimation of 3D particle shapes and extension of these techniques for a greater variety of input textures.

Acknowledgements

Thanks to Maxwell Planck for his assistance on the project and to Virginia Bernhard for the curled cat sculpture.

Figure 11: A synthetic image, rendered with the solid textures generated by our algorithm.

References

BLINN, J. F., AND NEWELL, M. E. 1976. Texture and reflection in computer generated images. *Communications of the ACM 19*, 10, 542–547.

DISCHLER, J., AND GHAZANFARPOUR, D. 1999. Interactive image-based modeling of macrostructured textures. *IEEE Computer Graphics and Application 19*, 1, 66–74.

DISCHLER, J., AND GHAZANFARPOUR, D. 2001. A survey of 3D texturing. *Computers and Graphics 25*, 10.

DISCHLER, J., GHAZANFARPOUR, D., AND FREYDIER, R. 1998. Anisotropic solid texture synthesis using orthogonal 2D views. *Computer Graphics Forum, Proceedings of Eurographics 1998 17*, 3.

EBERT, D. S., MUSGRAVE, F. K., PEACHEY, D., PERLIN, K., AND WORLEY, S. 1994. *Texturing and modeling: a procedural approach*. Academic Press Professional, Inc.

GARDNER, G. Y. 1984. Simulation of natural scenes using textured quadric surfaces. In *Computer Graphics (Proceedings of ACM SIGGRAPH 84)*, ACM, 11–20.

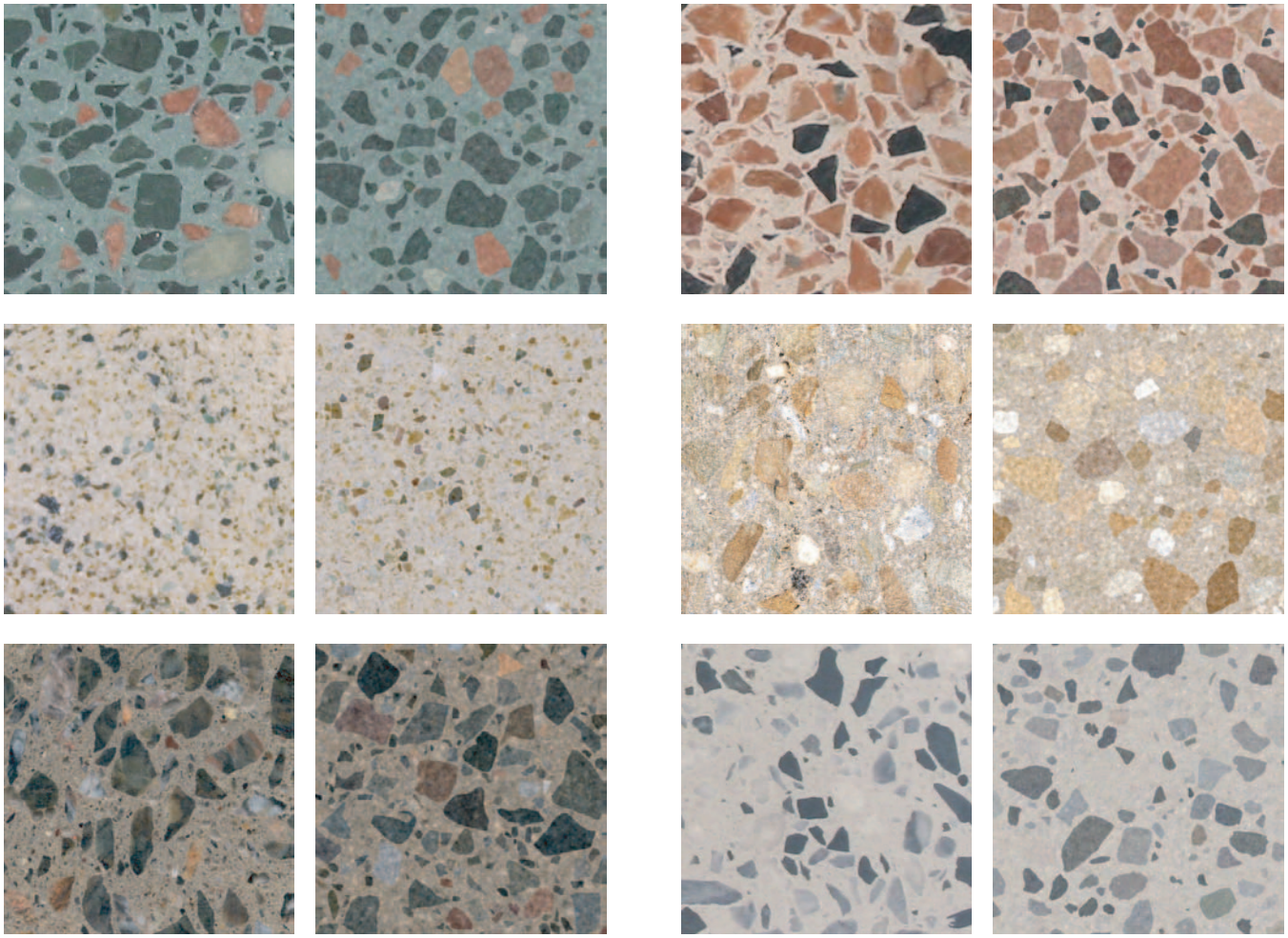


Figure 12: In each image pair, physical inputs to the solid texture algorithm are shown on the left and synthetic results are shown on the right.

GORLA, G., INTERRANTE, V., AND SAPIRO, G., 2001. Growing fitted textures. ACM SIGGRAPH 2001 Sketches and Applications, August.

HAGWOOD, C. 1990. A mathematical treatment of the spherical stereology. *NISTIR 4370* (July), 1–17.

HEEGER, D. J., AND BERGEN, J. R. 1995. Pyramid-based texture analysis/synthesis. In *Proceedings of ACM SIGGRAPH 2001*, Computer Graphics Proceedings, Annual Conference Series, ACM, 229–238.

HOWARD, C., AND REED, M. 1998. *Unbiased Stereology*. Springer-Verlag.

KEIDING, N., AND JENSEN, S. T. 1972. Maximum likelihood estimation of the size distribution of liver cell nuclei from the observed distribution in a plane section. *Biometrics 28*, 3 (September), 813–829.

LEFEBVRE, L., AND POULIN, P. 2000. Analysis and synthesis of structural textures. In *Graphics Interface 2000*, 77–86.

MAERZ, N. H. 1996. Reconstructing 3-D block size distributions from 2-D measurements on sections. In *Proceedings of the FRAGBLAST 5 Workshop on Measurement of Blast Fragmentation*, 39–43.

PEACHEY, D. R. 1985. Solid texturing of complex surfaces. In *Computer Graphics (Proceedings of ACM SIGGRAPH 85)*, ACM, 279–286.

PERLIN, K. 1985. An image synthesizer. In *Computer Graphics (Proceedings of ACM SIGGRAPH 85)*, ACM, 287–296.

SALTIKOV, S. A. 1967. The determination of the size distribution of particles in an opaque material from a measurement of the size distribution of their sections. In *Proceedings of the Second International Congress on Stereology*, 163–173.

TURK, G. 2001. Texture synthesis on surfaces. In *Proceedings of ACM SIGGRAPH 2001*, Computer Graphics Proceedings, Annual Conference Series, ACM, 347–354.

UNDERWOOD, E. E. 1970. *Quantitative Stereology*. Addison-Wesley.

WEI, L., AND LEVOY, M. 2001. Texture synthesis over arbitrary manifold surfaces. In *Proceedings of ACM SIGGRAPH 2001*, Computer Graphics Proceedings, Annual Conference Series, ACM, 355–360.

WEI, L. 2001. *Texture Synthesis by Fixed Neighborhood Searching*. PhD thesis, Stanford University.

WEI, L., 2003. Texture synthesis from multiple sources. ACM SIGGRAPH 2003 Sketches & Applications, July.

WOJNAR, L. 2002. Stereology from one of all the possible angles. *Image Analysis and Stereology 21*, Supplement 1, S1–S11.