Stereological Techniques for Synthesizing Solid Textures from Images of Aggregate Materials

by

Robert Carl Jagnow

Submitted to the Department of
Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

[February 2005]

December 2004

Author ....
Department of Electrical Engineering and Computer Science
December 31, 2004

Certified by ....
Julie Dorsey
Professor of Computer Science, Yale University
Thesis Supervisor

Accepted by ....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Stereological Techniques for Synthesizing Solid Textures from Images of Aggregate Materials

by

Robert Carl Jagnow

Submitted to the Department of Electrical Engineering and Computer Science
on December 31, 2004, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Computer Science and Engineering

Abstract

When creating photorealistic digital scenes, textures are commonly used to depict complex variation in surface appearance. For materials that have spatial variation in three dimensions, such as wood or marble, solid textures offer a natural representation. Unlike 2D textures, which can be easily captured with a photograph, it can be difficult to obtain a 3D material volume. This thesis addresses the challenge of extrapolating tileable 3D solid textures from images of aggregate materials, such as concrete, asphalt, terrazzo or granite.

The approach introduced here is inspired by and builds on prior work in *stereology*—the study of 3D properties of a material based on 2D observations. Unlike ad hoc methods for texture synthesis, this approach has rigorous mathematical foundations that allow for reliable, accurate material synthesis with well-defined assumptions. The algorithm is also driven by psychophysical constraints to insure that slices through the synthesized volume have a perceptually similar appearance to the input image.

The texture synthesis algorithm uses a variety of techniques to independently solve for the shape, distribution, and color of the embedded particles, as well as the residual noise. To approximate particle shape, I consider four methods—including two algorithms of my own contribution. I compare these methods under a variety of input conditions using automated, perceptually-motivated metrics as well as a carefully controlled psychophysical experiment. In addition to assessing the relative performance of the four algorithms, I also evaluate the reliability of the automated metrics in predicting the results of the user study.

To solve for the particle distribution, I apply traditional stereological methods. I first illustrate this approach for aggregate materials of spherical particles and then extend the technique to apply to particles of arbitrary shapes.

The particle shape and distribution are used in conjunction to create an explicit 3D material volume using simulated annealing. Particle colors are assigned using a stochastic method, and high-frequency noise is replicated with the assistance of existing algorithms. The data representation is suitable for high-fidelity rendering and physical simulation. I demonstrate the effectiveness of the approach with side-by-side comparisons of real materials and their synthetic counterparts derived from the application of these techniques.

Thesis Supervisor: Julie Dorsey
Title: Professor of Computer Science, Yale University

Acknowledgments

Thanks to my family for their understanding.

Thanks to Bob, Ray, and the lab for their knowledge.

Thanks to Fredo and Holly for their advice.

Thanks to my thesis committee for their guidance.

Thanks to Max for his assistance.

Thanks to Dave and Hector for their friendship.

Thanks to Jeff for his encouragement.

Thanks to MITOC for keeping me sane.

Thanks to Sloan Crew for keeping me pumped.

Thanks to Tomas for his patience, kindness, and infinite support.

Contents

List of Figures

Chapter 1

Introduction

Within the field of computer graphics, one of the most common objectives is to replicate the appearance of real-world objects in a simulated digital environment. Video games attempt to simulate reality to create immersive, compelling environments. Architects need to accurately depict the appearance of real materials in order to create convincing, physically-correct renderings of what a building will look like, even before the first brick is laid. And as digital models become increasingly common in Hollywood entertainment, movie and television producers are concerned with replicating natural appearances so that digital objects can be seamlessly composited with live footage. In all of these areas, there is a pressing need for accurate depiction of real-world materials.

1.1 Reality in a Box

When using a computer to generate photorealistic scenes, part of the challenge is representing a continuous, infinitely detailed world within the confines of a digital, finite medium. Artists and researchers rely on simplifying assumptions and heuristics to reduce reality into a manageable form.

At the most coarse level, real-world objects can be defined in terms of their shape and color. But even these attributes are not as simple as they might first seem. An accurate representation of physical shape includes not only the coarse geometry, but also the fine-scale features that contribute to the overall appearance. Likewise, an accurate physical representation of color requires knowledge about how much light is reflected in each outgoing direction on the surface as a result of incident light from each incoming direction. A number of data representations have been developed to make the complexity of the real world manageable for purposes of realistic image depiction.

1.2 Geometric Representation

Inside a computer, the 3D geometry of a digital scene can be described in a variety of ways. The most common representation is the polygon mesh, which consist of a collection of vertices and connected polygonal faces. For real-time computer rendering, polygon meshes are used almost exclusively, as specialized hardware can process this type of data extremely quickly. Other common data structures include subdivision surfaces and non-uniform rational B-splines (NURBS), each of which is defined by a collection of control points and parameters to govern surface smoothness. There are a variety of other 3D surface representations, both implicit and explicit.

Each representation has its own advantages. Triangle meshes are simple and fast to render, making them appropriate for video games or other real-time visualization. Subdivision surfaces and NURBS offer smooth variation along surface geometry for natural, high-fidelity appearances. These latter representations are used more frequently in architectural renderings or movies, where images can be generated over a longer period of time, and physical correctness is more important than rapid synthesis.

Of course, the appearance of real-world objects is affected by more than just their shape. Color and other material properties must also be accurately represented in order to render a synthetic scene with a convincing look. This type of additional information could be associated with each vertex, polygon, or control point of a geometric model, but this approach has severe limitations when attempting to represent detailed appearances.

1.3 The Role of Texture

Most real-world objects exhibit complex spatial variation in their surface color and finish. In theory, this type of appearance might be achieved in a digital scene by creating fine-scale geometry that represents every tiny bump and material feature. However, in practice, creating geometry with this level of detail is prohibitive from a standpoint of authorship, digital storage, and computer rendering.

Instead, for simulating complex appearances, the computer graphics community relies heavily on the use of *texture*—color and material data that can be mapped to the surface of a geometric model [Catmull 1974; Blinn and Newell 1976]. In its simplest form, a texture can be a photograph of a real-world material. When projected onto a geometric surface, textures can convey the appearance of material variation, as well as an illusion of small-scale geometric features.

Textures can contain more than just color information. Small aberrations in surface geometry can be parameterized in 2D with a variety of representations that include bump maps [Blinn and Newell 1976], displacement maps [Cook 1984], or bidirectional texture functions [Dana et al. 1999]. Furthermore, textures can encode other variations in material parameters such as specularity, opacity, or surface normal.

There are a number of challenges associated with authoring textures and mapping them to a 3D surface. To achieve a desired appearance, an artist may either paint a texture by hand or take a photograph of a physical structure, if such an object exists. This swatch of texture can then be tiled over the surface of a digital model. However, the human eye is very perceptive of the repetitive artifacts that can be caused by texture tiling; the result may be visually distracting and can detract from the overall appearance.

Of course, most objects in digital scenes are not purely planar. Mapping an inherently 2D texture to the surface of a 3D model can be a challenging task. The texture parameterization may result in distortion and visible seams. Even for a simple object like a sphere, there is no distortion-free mapping from a 2D texture onto the 3D surface, and singular points and seams are unavoidable. Furthermore, at sharp corners like those on a cube, it is often impossible to maintain a consistent appearance across geometric edges. Examples of

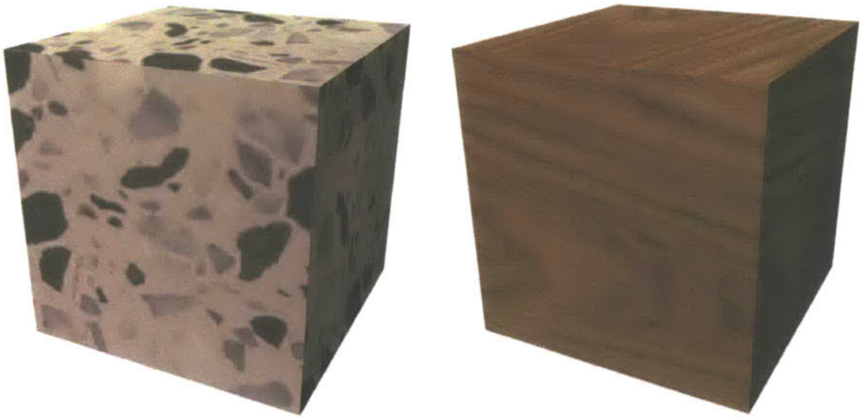


Figure 1-1: Problems of texture continuity are often apparent at geometric seams when using 2D textures. These boxes appear as though they are wallpapered as opposed to being cut out of solid materials.

texture inconsistencies at geometric boundaries can be seen in Figure 1-1.

The problem of texture continuity is often especially apparent for *anisotropic* textures, which exhibit varying appearance characteristics in different directions. For instance, on the surface of a wooden table, the grain of the wood has a clear orientation. As demonstrated by the wooden cube in Figure 1-1, more than one 2D texture would be needed to accurately portray this appearance consistently on all six faces.

In many cases, the problem of texture consistency stems from the fact that the appearance of a real object may result from being cut out of a 3D spatially varying material. 2D textures may be appropriate for conveying the appearance of fundamentally 2D features such as paint, skin, fur, or mechanically roughened surfaces. However, two-dimensional representations are often inappropriate for inherently volumetric materials such as wood, concrete, or granite. Gardner [1984], Peachey [1985], and Perlin [1985] introduced the idea of 3D *solid textures* as a representation for such spatially varying materials.

Just as a surface texture can be expressed as a 2D array of *texels*, a solid texture can be represented as a 3D array of *voxels*—discrete volumetric elements that contain color or material parameters. Solid texture can be mapped to the surface of a geometric model by evaluating where each point on the surface of the model falls within the voxel array.

Solid textures effectively address the problems of surface parameterization and consistency, but introduce new challenges as well. Simply obtaining a solid texture can be a difficult task. While 2D textures can be captured with a simple photograph, it is typically impractical to scan a 3D object by cutting numerous thin slices through the material. For hard materials such as concrete or granite, this type of destructive process may be infeasible. A few nondestructive scanning techniques exist, including magnetic resonance imaging, tomography, and confocal microscopy; however, these techniques require costly measuring instruments, and are only practical within certain domains of materials.

Even if a solid texture can be obtained from a physical object, the result is limited in space and resolution. In order to texture a geometric model that is larger than the texture volume, the 3D texture may be tiled in space, which can result in unacceptable visible artifacts along texture seams.

One alternative to the voxel representation is *procedural textures*—3D functions that can be evaluated at any point in space to return material parameters. For many years, procedural techniques have proved useful for the artistic generation of 3D solid textures. However, engineering a realistic procedural texture can be a challenging task that involves deriving a mathematical expression to quantify appearance characteristics. Furthermore, 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. Ideally, it is preferable to extrapolate such parameters automatically from a 2D image with a desired appearance.

1.4 Texture Synthesis

The problems associated with finite textures are addressed by research in the area of *texture synthesis*—techniques that aim to replicate the qualitative appearance of an input image without the repetition that results from simple tiling [Ebert et al. 1994]. Much of this research relies on assumptions about how humans perceive textures. A number of texture synthesis algorithms have been proposed, each of which tends to work well within a specific domain of appearances.

In addition to addressing the problem of tiling artifacts, research in texture synthesis offers solutions to the problems of continuity and distortion caused by surface parameterization. A number of authors have proposed algorithms for synthesizing textures directly on 3D objects based on representative 2D images [Wei and Levoy 2001; Turk 2001; Gorla et al. 2001; Zhang et al. 2003]. These synthesis algorithms operate within a local texture domain, which allows the techniques to guarantee continuity across geometric edges while minimizing spatial distortion. By using physically occurring input textures, these algorithms can often produce rich, natural appearances.

The vast majority of texture synthesis research has focused on synthesizing 2D textures from 2D image inputs. Only a few viable solutions have been proposed for the synthesis of 3D textures from 2D images [Heeger and Bergen 1995; Dischler et al. 1998; Lefebvre and Poulin 2000; Dischler and Ghazanfarpour 2001; Wei 2001; Wei 2003]. This prior work will be discussed in greater detail in Chapter 2.

1.5 3D Textures of Aggregate Materials

In this thesis, I address the challenge of synthesizing 3D solid textures from 2D input images for the class of textures composed of discrete 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. Some examples of these types of materials are shown in Figure 1-2.

The algorithms described in the following chapters extrapolate information about the 3D volume based on observed *profiles*—2D curves that denote the boundaries between discrete particles and the binding material that surrounds them. I introduce a number of algorithms that analyze this input to recover material parameters that quantify the texture appearance.

To address these challenges, I build on techniques from the disciplines of computer vision and computer graphics, as well as established methods in *stereology*—a spatial version of sampling theory developed in the fields of biology and material science [Hagwood 1990; Underwood 1970]. Specifically, much of the work in stereology describes the relationship between the 3D parameters of a material and the 2D features that can be seen in a single

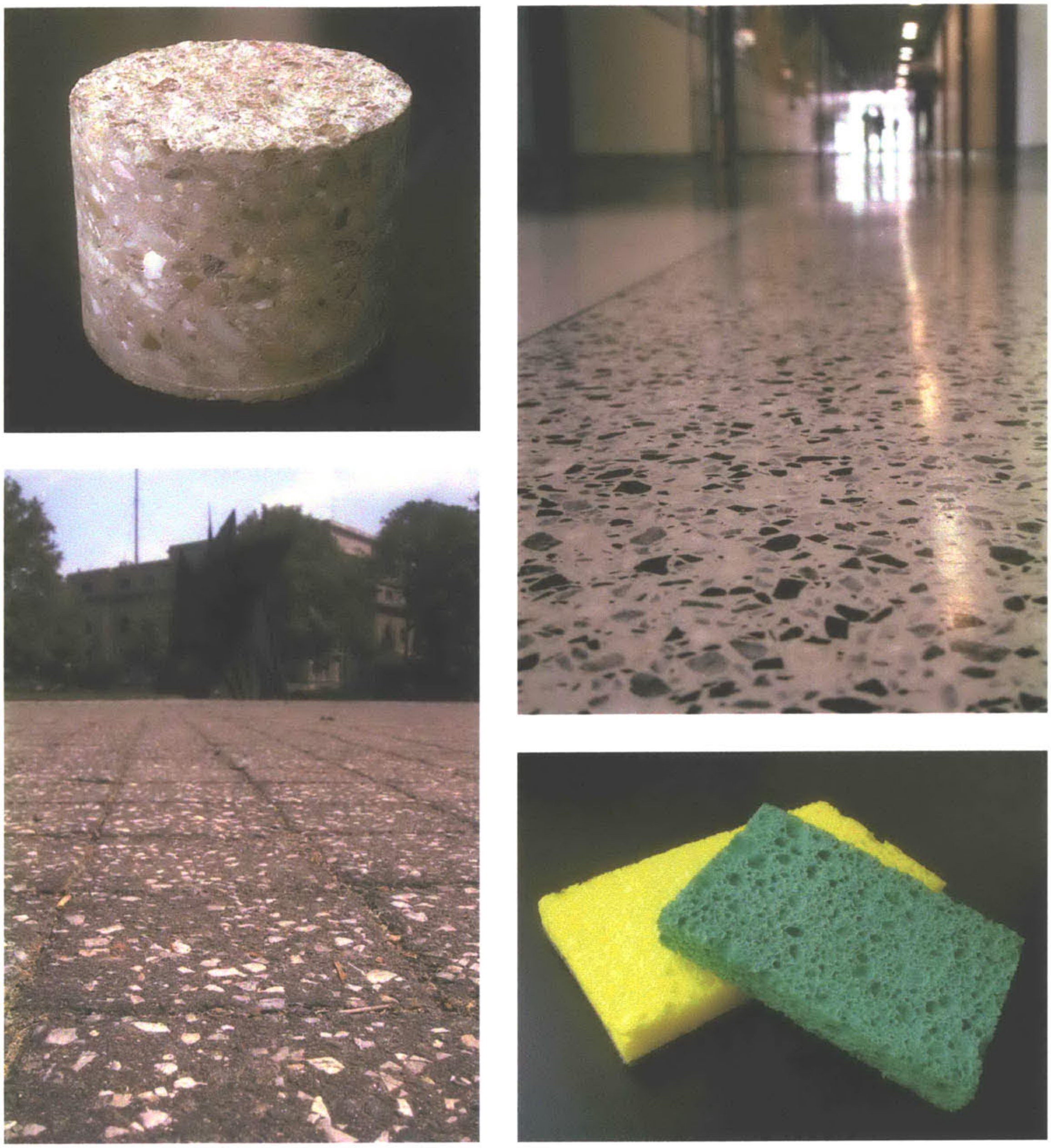


Figure 1-2: Each of the materials shown here can be represented as particle aggregates. Clockwise from upper-left are concrete, terrazzo, sponge, and asphalt. For the sponge, the material voids can be given the same treatment as the particles in the other examples.

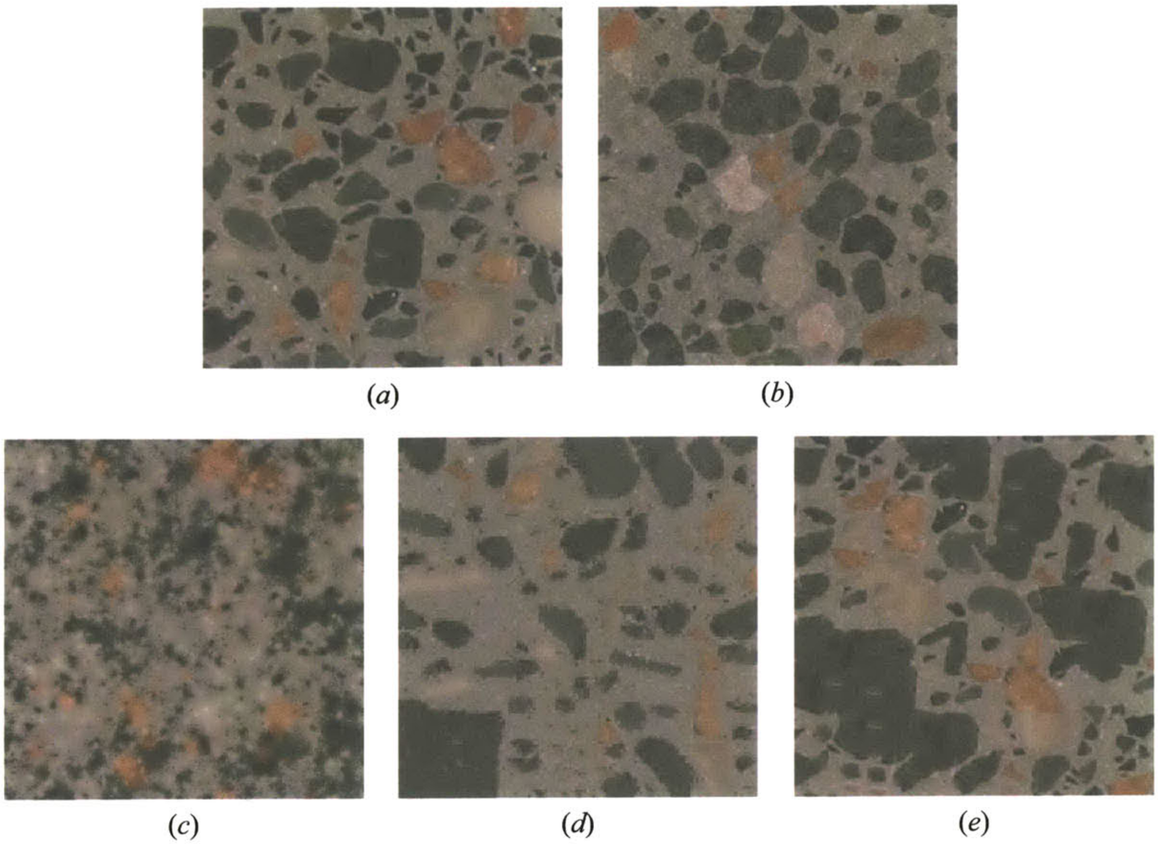


Figure 1-3: An image of an aggregate material (*a*) can be approximated using the techniques described in this thesis (*b*). Shown for comparison are the results produced by Heeger and Bergen's algorithm [1995] (*c*), Efros and Leung's method [1999] (*d*), and Efros and Freeman's Image Quilting [2001] (*e*).

slice through that material. In biology, these relationships allow an observer to extrapolate properties of an organic tissue sample based only on 2D observations of the cells. In material science, these relationships are important for predicting the structural characteristics of a physical object.

The texture synthesis approach introduced here independently recovers material parameters for the particle shape, particle distribution, color and noise. Once all the parameters of a material have been obtained, I demonstrate how to construct a synthetic texture volume with a comparable appearance to the input example. A sample comparison of original and synthetic textures is shown in Figure 1-3. The input image (*a*) is analyzed to recover the texture parameters. A single slice through this synthetic volume (*b*) is perceptually similar to the input. The solid texture can be tiled seamlessly in three dimensions and is suitable for high-fidelity rendering.

1.6 Stochastic vs. Structured

Generally speaking, textures can be classified according to their appearance on a scale from stochastic to structured. On one extremum are entirely random textures, such as white noise. On the other end are highly structured textures, such as a checkerboard. Aggregate materials form a particularly interesting class of textures because their appearance contains elements at both ends of this classification spectrum. The textures are structured in the sense that profile appear as finite, disjoint regions with sharp boundaries, characteristic shapes, and precise distributions. However, particle shapes and distributions also exhibit stochastic attributes, and the overall texture appearance is affected by random noise and color variation.

This mix of stochastic and structured attributes makes aggregate materials particularly challenging from a standpoint of both analysis and synthesis. A number of existing algorithms in texture synthesis assume stochastic inputs. When these methods are applied to images of aggregate materials, they tend to blur particle boundaries and eliminate structure. An example of this can be seen in the synthetic image in Figure 1-3(*c*), which was produced using Heeger and Bergen's algorithm [1995].

Other research has considered the use of nonparametric models that aim to enforce local texture coherence. These methods are able to capture more of the local texture structure, but they often fail to effectively capture the individual particle shapes and overall distribution. Figure 1-3(*d*) was generated by Efros and Leung's texture synthesis algorithm [1999] on an input texture of size 128×128—half the resolution of the image shown in Figure 1-3(*a*). Texels were selected from the five best matches obtained with a window size of 13 pixels.

Results from Efros and Freeman's Image Quilting algorithm [2001] are shown in Figure 1-3(*e*). These results were obtained with an input image of size 256×256 using a patch size of 30×30 and an overlap region of five texels. Patches in the synthesized texture were randomly selected from the best three matches.

The results produced by Efros and Leung's algorithm and by Efros and Freeman's Image Quilting are two-dimensional and have a significant dependence on the selected input parameters. These and other existing algorithms are discussed in greater detail in Chapter 2.

In order to synthesize textures that are perceptually similar to the input, both the stochastic and structured attributes must be replicated. In this thesis, by independently considering the parameters of a texture—particle shape, particle distribution, color, and noise—I am able to evaluate each distinct characteristic in the most appropriate domain.

1.6.1 Structured Attributes—Shape and Distribution

Particle shape and distribution are considered from a structural perspective. Each image is first segmented into discrete profiles, reducing the texture to collection of well-defined shapes. 3D Particle shapes are estimated by extrapolating information from the 2D profiles. I consider four different methods for particle synthesis, three of which incorporate characteristic profile shapes directly into synthetic 3D particles.

In order to evaluate the performance of these algorithms, the resulting 3D particles are, in turn, evaluated relative to the profile shapes that they exhibit. Profiles from the original texture are directly compared to profile slices through the synthetic 3D particle via automated metrics and psychophysical evaluation. This approach is described in greater detail in Chapter 3.

To approximate the particle distribution, simple counting measures are computed on the segmented image profiles. The 3D distribution is then derived by applying stereological techniques, as described in Chapter 4.

1.6.2 Stochastic Attributes—Color and Noise

In contrast to the structured attributes of shape and distribution, color and noise parameters are considered from a stochastic perspective. For most input textures, it is safe to assume that there is no spatial relationship between the colors of distinct particles. As such, colors from segmented 2D profiles can be randomly assigned to 3D particles.

In addition to the particle colors, I assume the presence of a residual noise function that contributes high-frequency information to a texture. This noise can be captured with the assistance of existing techniques that assume stochastic inputs. Color and noise are discussed in detail in Chapter 6.

1.7 Contributions

This thesis addresses unsolved problems in the areas of texture synthesis, stereology, and shape perception. The primary contributions are as follows:

- A comprehensive algorithm for synthesizing 3D textures of aggregate particles from physical 2D samples. Slices through the resulting 3D textures are perceptually similar to the input.

- Introduction of two novel methods for extrapolating 3D particle shapes from 2D profiles.

- Analysis of four 3D shape approximation methods compared via automated metrics and a carefully implemented user study.

- Assessment of two automated metrics as predictors for the psychophysical evaluation of 2D shape similarity.

- A computational method for recovering the size distribution of one or more arbitrarily shaped particles.

- Methods for reconstructing a synthetic volume based on recovered parameters for particle shape, particle distribution, color, and noise.

- Validation of the solid texture synthesis algorithm via direct comparison of a physical model and its synthetic counterpart.

Although this thesis considers a restricted class of solid textures, by drawing on stereology and the understanding of human perception as developed in other fields, this research adds to the existing array of tools for extracting 3D information for computer graphics applications. Furthermore, since stereology and the study of human perception have been developed as a tools for quantitative analysis, they have well-defined assumptions and a rigorous mathematical basis. This allows for the generation of reliable, precise solid textures for computer graphics applications.

In contrast to the voxel-based data structure described previously, the data representation that I have chosen to use maintains exact particle shapes, yet is more compact than a discretely sampled volume. Furthermore, by maintaining a notion of explicit, precise particle shapes, the volumetric data is appropriate for physical simulation and direct 3D visualization.

1.8 Overview

In the remainder of this thesis, I begin by describing some of the prior research that serves as the foundation for this work, as described in Chapter 2.

Chapter 3 describes four different techniques for extrapolating 3D particle shapes from 2D profiles, including two novel methods that are introduced here. These four methods are then compared using automated metrics, as well as a complete user study. In addition to assessing the relative performance of the proposed algorithms, Chapter 3 evaluates the utility of the automated metrics as reliable predictors of algorithm performance from a perceptual standpoint.

Stereological techniques for recovering particle distribution are described in Chapter 4. This is followed in Chapter 5 by a discussion about sources of error and interdependence between particle shape and distribution.

Methods for approximating particle color and high frequency noise are described in Chapter 6. This chapter also describes how the recovered material parameters are used together to populate and anneal a synthetic texture volume. I provide a detailed description of the data representation, including considerations for rendering high-fidelity images.

In Chapter 7, I demonstrate the results of the algorithm as applied to a number of input texture examples. Finally, in Chapter 8, I provide conclusions and discuss potential avenues for future research.

Chapter 2

Previous Work

The work presented in this thesis draws on prior contributions in the areas of computer vision, human perception, computer graphics, and stereology. The most relevant research is described below.

2.1 Texture Authoring and Dimensionality

Digital textures were first used for computer-generated images in the mid-1970s [Catmull 1974; Blinn and Newell 1976]. Since then, textures have been widely recognized for their value in creating compelling synthetic images by conveying complex material appearances without substantially altering the underlying geometry.

2.1.1 2D Texture

Textures are routinely categorized according to the dimension of the data used. In their simplest form, two-dimensional textures map color data from an image to a surface. In this case, the textures are typically stored as 2D arrays of *texels*—color samples that may be augmented by additional material parameters such as specularity, transparency, or surface normal. Textures may be obtained by using procedural techniques, physical simulation, artistic authoring, or direct acquisition from real-world objects.

The simplest way to obtain a 2D texture is to photograph an object that depicts a desired appearance. As with any finite representation, if a photograph is to be applied to a geometric surface that is larger than the texture region, then tiling artifacts may result.

To give the illusion of small-scale geometric bumps or wrinkles, *bump maps* can be used to depict small displacements in the surface geometry [Blinn 1978]. A 2D array containing small offsets from the underlying surface can be filtered to obtain a *normal map*, which indicate how the surface normal is perturbed when performing shading computations. This representation can give a compelling illusion of surface detail, but does not actually alter the underlying geometry. Bump maps can be procedurally generated [Blinn 1978] or estimated from existing surfaces [Rushmeier et al. 1997].

Procedural methods for authoring 2D textures include the use of reaction-diffusion, which is effective for creating striped and spotted appearances on surfaces [Witkin and Kass 1991; Turk 1991]. Other authors have used physical simulation to replicate real-world phenomena such as metallic patinas [Dorsey and Hanrahan 1996], weathering patterns created by the flow of water [Dorsey et al. 1996], corrosion [Merillou et al. 2001], and biological growth [Sumner 2001].

At the far end of the 2D texture representation are *Bidirectional Texture Functions (BTFs)*, which use a collection of input images to simulate remarkably complex appearance characteristics without altering the underlying geometry [Dana et al. 1999; Vasilescu and Terzopoulos 2004]. Unlike bump maps, BTFs can convey the appearance of self-occlusion, self-shadowing, and interreflection.

2.1.2 2.5D Texture

In contrast to 2D bump maps or BTFs, *displacement maps* modify the actual geometry of an object in a region near its surface [Cook 1984]. Displacement maps fall into the category of textures that are often referred to as being 2.5-dimensional, as they create 3D geometry, but their influence is typically limited to a constrained region near the surface of an underlying model.

For simulating certain types of real-world appearances, 2.5D textures offer an effective, concise representation. For instance, Dorsey et al. noted that a variety of weathering effects take place only in a thin region near the surface of real-world objects [Dorsey et al. 1999]. Their research represented models as surface-aligned volumetric *slabs*—an effective foundation for physical simulation of weathering phenomena. Some simulations for cracking and peeling produce what might be considered 2.5D textures [Hirota et al. 1998; Paquette et al. 2002], as do the cellular texture methods of Fleischer et al. [1995].

Geometry images fall at the extreme end of 2.5D textures [Gu et al. 2002; no et al. 2003]. With this representation, a single 2D array of texels can be used to depict a complete 3D model. Points in the array are transformed into 3D space based on their color, and model connectivity is implicit in the adjacency relationships of the texels.

2.1.3 3D Texture

For representing materials that exhibit spatial variation and internal structure, 3D textures offer a natural solution. Common materials that fall into this category include wood, concrete, asphalt, granite, and marble. In applications where 3D modifications are made to rigid objects, solid textures are vital for maintaining a consistent, realistic appearance [Cutler et al. 2002].

Similar to solid textures, a number of authors have also used *volumetric textures* to procedurally represent finely scaled geometric features enclosed within a volumetric region. This representation is useful for rendering detailed appearances such as hair [Kajiya and Kay 1989], clouds [Schpok et al. 2003], foliage [Neyret 1996], or other complex, repetitive geometry [Neyret 1995].

The overwhelming majority of solid textures that are used in computer graphics are procedurally generated [Gardner 1984; Peachey 1985; Perlin 1985; Perlin and Hoffert 1989; Ebert et al. 1994]. For simulating materials with discrete particle shapes, Worley introduced a cellular texture basis function [1996]. Unfortunately, none of these representations are well-suited for automated extrapolation of appearance parameters from example images. Furthermore, existing procedural shaders often do not have a sufficiently large parameter space to accurately portray the appearance of specific aggregate materials.

The dearth of physical 3D textures stems largely from the fact that real-world 3D datasets tend be be exceedingly difficult to obtain. Non-destructive methods such as tomography or magnetic resonance imaging can measure internal material structures, but these processes require costly equipment, and the acquired data are often not appropriate for representing appearance parameters.

2.1.4 Surface Parameterization

One drawback of 2D and 2.5D textures is that an explicit parameterization must be defined to map points from 2D texture space onto the 3D geometry in digital scenes with nonplanar surfaces. For some applications, straightforward parameterizations such as orthographic or perspective projection can be used to define this relationship [Heckbert 1986]; however, these simple techniques yields stretching artifacts on surfaces that are nearly tangent to the direction of projection.

A number of automated parameterization methods have been proposed [Bier and Sloan 1986; Ma and Lin 1988; Bennis et al. 1991; Maillot et al. 1993; Lee et al. 1998; Lévy 2001; Lévy et al. 2002], each based on metrics for minimizing texture distortion, which can be defined according to a variety of different criteria. Metrics have been introduced to minimize change in distance [Zigelman et al. 2002], preserve right angles [Lévy and Mallet 1998; Haker et al. 2000], reduce the visibility of texture seams [Tarini et al. 2004], or adapt to texture details [Carr and Hart 2004]. Parameterizations derived from these methods are model-specific, and must be recomputed for each new geometric shape.

In contrast to 2D and 2.5D textures, no explicit parameterization is necessary for 3D textures. Assuming that the texture volume fills the entire space surrounding a geometric model, surface points can be colored by indexing directly into the 3D array or procedural shader.

2.2 Texture Synthesis

To address the challenge of authoring large quantities of textures from small input examples, a number of researchers have recently turned their attention to texture synthesis.

Existing methods in texture synthesis can be classified according to the dimensionality of the input data and the dimensionality of the synthesized texture result.

2.2.1 2D to 2D

Most algorithms in texture synthesis start with a 2D source texture and yield a 2D target. This category of methods can be further classified according to the assumptions made regarding how humans perceive texture, and how this knowledge can be used to effectively replicate a desired appearance.

Frequency Analysis

One category of research is founded on the assumption that texture frequency at various scales has a strong influence on human perception of appearance. Heeger and Bergen rely on this assumption to perform pyramid-based texture analysis and synthesis, which coerces an initial noise distribution such that the energy histogram of each frequency band matches the histogram of the corresponding frequency band in a source image [1995]. This approach tends to be effective for replicating stochastic textures, but does not perform well on structured images.

When operating on color images, Heeger and Bergen assume a strong interdependence between the red, green, and blue color channels. Their technique first decorrelates the color bands in a manner that captures the majority of the color variation in a single dimension. The algorithm then operates on each decorrelated channel independently and finally recombines the results. For images in which the color variation cannot be effectively captured in a single dimension, this method tends to yield color artifacts.

Portilla and Simoncelli allow for synthesis of more structured textures by utilizing a model that captures and reproduces the joint statistics of wavelet coefficients [1998; 2000]; however, the approach is only described for greyscale images, and is still poorly suited for use with highly structured textures. In particular, for textures that exhibit discrete particles, the individual particle shapes tend to be merged together or broken irregularly.

Statistical Sampling

In the texture synthesis work of De Bonet, novel textures are created in a two-stage process [1997]. In the analysis phase, the algorithm computes the joint occurrence of visual features across multiple frequencies in the source image. In the synthesis phase, a target image is created by successively adding details in bands of increasing frequency in a manner that respects the joint occurrence of features observed during the analysis process.

This approach works remarkably well for reproducing the appearance of stochastic textures, but fails to effectively replicate structured textures.

Local, Nonparametric Methods

Other research in 2D texture synthesis is based on the assumption that texture appearance depends primarily on local coherence in a small neighborhood surrounding each texel. This assumption has led to the development of methods that synthesize textures sequentially, one small region at a time, often in a manner that copies data directly from a source image to a target image [Popat and Picard 1993].

In particular, a number of authors have adopted the use of *Markov Random Fields (MRFs)*—graph-based representations that can be used to model probabilistic dependencies between graph nodes [Zhu et al. 1998; Paget and Longstaff 1997; Efros and Leung 1999]. Dependencies are limited to the local neighborhood of a graph node, making the approach well-suited for perceptual models that emphasize the role of local texture coherence.

Efros and Leung utilized an MRF framework to synthesize one texel at a time by searching for texels in the source image that have a similar neighborhood to those already synthesized in the target image [1999]. An example result produced by their method can be seen in Figure 1-3(*d*). Due to the size of the search space, this approach tends to be very slow. Wei and Levoy accelerated this method using tree-structured vector quantization and multiresolution synthesis [2000]. Their use of a hierarchical approach improved the performance of the algorithm, but the method still yields artifacts when applied to structured source images.

Efros and Freeman introduced the use of image quilting to address the problems of speed and coherence in local, nonparametric texture synthesis models [2001]. Their algorithm stitches together small patches of texture from a source image. As can be seen in Figure 1-3(*e*), this tends to result in particle shapes and distributions that do not effectively reflect those see in an input image. This approach was later refined to use irregular image patches, which can be stitched together with graph cuts [Kwatra et al. 2003].

Other noteworthy contributions in this area come from Ashikhmin, who includes a mechanism for limited user control of the texture synthesis process [2001], and Hertzmann et al., who introduced a general image analogies framework for transferring appearance characteristics between images [2001].

Because they operate in a local texture neighborhood, these nonparametric methods tend to be well-suited for synthesis directly on the surface of 3D geometry [Wei and Levoy 2001; Turk 2001; Gorla et al. 2001; Zhang et al. 2003]. However, it should be remembered that such parameterizations are model-specific and must be recomputed for each geometric model where surface texture is desired. In contrast, solid textures can be conveniently mapped to the surface of any 3D model without having to perform an explicit parameterization.

2.2.2 3D to 3D

A handful of the algorithms discussed in the previous section can also be adapted for synthesizing 3D textures if a fully 3D solid texture sample is available [Wei and Levoy 2001; Efros and Leung 1999; Kwatra et al. 2003]. However, this class of algorithms tends to be of limited use, as it can be difficult to obtain an fully three-dimensional input sample. Furthermore, increasing the dimensionality of the input tends to amplify the problems that local nonparametric methods have with speed and coherence.

Falling into a category by itself is work by Bhat et al. [2004] that uses the image analogies framework introduced by Hertzmann et al. [2001] to synthesize 3D geometry based on example surface deformations. This geometric texture synthesis process can be best characterized as yielding a 2.5D result, as the geometric deformations are limited to a region near the surface of the original model.

2.2.3 2D to 3D

In this thesis, I am interested primarily in the class of texture synthesis algorithms that take a 2D physical sample as input and extrapolate a 3D volume with a comparable appearance. This is an enormously useful class of algorithms, as 2D input samples can be easily obtained, and the resulting 3D textures offer advantages with parameterization and visual consistency. However, only a few such methods exist.

Heeger and Bergen's pyramid-based texture analysis and synthesis, which was described above for 2D textures synthesis, can also be used to synthesize a solid texture from a 2D input [1995]. The process can be divided into an analysis stage and a synthesis stage. During analysis, a 2D Laplacian pyramid is derived from an input texture such that each level of the pyramid corresponds to a particular frequency band in the image. In the synthesis phase, a 3D Laplacian pyramid is used. The energy histogram at each subband in the 3D pyramid is coerced to match the histogram at the corresponding level of the 2D pyramid. The 3D pyramid is then collapsed to yield a solid texture with a comparable appearance to the input.

In the same spirit, Dischler et al. [1998; 2001] use a spectral analysis of orthogonal images of a physical 3D volume and iteratively alter a 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. These algorithms work well for a subclass of common natural textures, but are unable to reliably reproduce the appearance of materials composed of discrete particles.

Lefebvre and Poulin [2000] successfully generate 3D wood textures from 2D images by analyzing an input image to obtain parameters for a procedural shader. This specific algorithm does not generalize for other classes of solid textures. However, the concept of automatically recovering parameters for a procedural model shares some similarities with the methods used in this thesis.

Markov Random Fields have been used with limited success in generating 3D solid textures from 2D images. A method introduced by Wei [2001; 2003] attempts to generate texels one at a time by simultaneously matching texel neighborhoods from three orthogonally oriented input textures. This approach is successful for some textures classes, but tends to amplify texture artifacts that are seen when the methods are used in 2D. Because the method computes only one voxel at a time, it may be unreasonably slow for calculating a large synthetic volume. When applied to images of aggregate materials, the resulting 3D textures fail to accurately characterize the size or distributions of macroscopic particles, and may also poorly represent the overall color histogram.

2.3 Recovering Particle Distributions

When dealing with discrete macroscopic particles, part of the challenge in synthesizing a 3D texture is accurately recovering the distribution of particles based only on a 2D input. This problem is also of interest in the material and biological sciences, where 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; Hagwood 1990].

Since obtaining full three-dimensional samples of solids is an expensive and time-consuming process, the discipline of stereology was developed to infer 3D shape and distribution from 2D samples. The distribution recovery technique that is used in this thesis is most closely related to that of Saltikov [1967], who uses histogram binning to establish a correlation between the distributions of 2D profiles and 3D particles of various sizes. This approach will be discussed in greater detail in Chapter 4.

Comparable techniques have been described by a number of other authors [Wicksell 1925; Keiding and Jensen 1972; Schopper 1975; Taylor 1983]. DeHoff attempts to fit a simple two-parameter curve to the particle sizes by assuming that they exhibit a lognormal distribution, which is commonly the case for physical materials [1965]. However, this approach overly constrains the space of possible solutions.

With the advent of digital imagery, image analysis and stereology are frequently used in conjunction for a variety of applications [Wojnar 2002]. Stereological techniques were first used for texture synthesis by Jagnow et al. [2004], who used the methods for estimating particle distributions when assuming that the individual particle shapes are known *a* *priori*. This thesis relaxes these early assumptions regarding the particle shapes and further demonstrates the application of some of the fundamental techniques of stereology for solid texture synthesis.

2.4 Particle Shape Estimation

Before we can recover particle distributions, we need a suitable method for extrapolating individual 3D particle shapes based only on observed 2D profiles. This problem has been previously addressed within computer graphics, computer vision, and stereology. A number of specialized solutions have been presented, each targeted to particular assumptions and data constraints.

2.4.1 Surface Inflation

When only a single profile is available, Igarashi et al. constrain the reconstruction problem by assuming that the 3D shape exhibits a circular profile along its chordal axis [1999]. This approach yields smoothly curved polygonal models. Other authors have adapted this method to operate on implicit surfaces[Karpenko et al. 2002; De Araujo and Jorge 2003]. However, particle inflation is suitable only for producing smooth, rounded shapes, and is poorly suited for synthesizing particles with sharp corners.

2.4.2 Spherical Harmonics

Methods for extrapolating individual cell shapes from 2D biopsies have been studied by a number of authors within the stereological community. In *histology*—the study of microscopic structures of plants or animals—the analysis of particle distribution, shape, and variability has been used for detecting the presence of cancerous or precancerous cells [Gardner et al. 2004] and other diseases [Keiding and Jensen 1972].

For estimating 3D shapes, the spherical deformation model is one of the most commonly used methods within the stereological community. Using spherical harmonics as basis functions, Hobolth and Jensen demonstrate how to establish a relationship between the observed frequencies in a 2D profile and those of a 3D particle [2002; 2003]. A number of other authors have also adopted the use of spherical harmonic basis functions for 3D shape approximation, as the representation is compact and mathematically rigorous [Edvardson and Smedby 2002; Weistrand 2001].

While spherical harmonics offer an appropriate shape representation within certain domains, these methods are often unsuitable for representing particles with sharp corners or distinct features. This will be discussed in greater detail in Chapter 3.

2.4.3 Volume Carving

Within computer vision, a large collection of research has been devoted to estimating shape from object silhouettes. In the case where camera locations are known, Martin and Aggarwal developed a computational method for approximating the bounding volume of an object by intersecting extruded silhouettes of the object as seen from various viewpoints [1983]. In this pioneering work, it is assumed that the silhouettes are obtained via orthographic projection. Accuracy can be improved by using conical projections that originate from image viewpoints [Laurentini 1994; Matusik et al. 2000; Moezzi et al. 1996; Szeliski 1993; Fang et al. 2003; Saito and Kanade 1999] and by incorporating photometric measurements into the volume estimation process [Seitz and Dyer 1999; Kutulakos and Seitz 2000; Kutulakos 2000]. As the number of viewpoints approaches infinity, the shape of the result approaches that of the object's *visual hull*. Unfortunately, this representation is unable to capture certain types of shape concavities [Laurentini 1995].

Volume carving may result in models with a faceted appearance due to a limited number of viewpoints. Some authors have attempted to smooth the surfaces by using splines for the shape representation [Sullivan and Ponce 1998; Lazebnik et al. 2001].

For the images considered in this thesis, the observed 2D profiles represent object slices rather than projected silhouettes. Furthermore, each profile is generated by a different particle, so there is no known correlation between distinct input shapes. Regardless, by applying simplifying assumptions to the input data, I demonstrate the applicability of volume carving methods for particle shape approximation.

2.4.4 Generalized Cylinders

Within the computer vision community, the generalized cone and generalized cylinder have long been recognized as effective primitives for shape representation due to their ability to express a wide variety of shapes with few parameters [Binford 1971; Marr and Nishihara 1978; Brooks 1981]. In the graphics community, Dischler and Ghazanfarpour [1999] used generalized cylinders to model discrete particle shapes embedded in macrostructured textures. However, the resulting shapes tend to exhibit unnatural symmetry that is often not present in the input data.

This thesis considers the use of generalized cylinders for shape approximation and also introduces a novel method derived from this representation.

2.4.5 Multiple Slices

For some data collection techniques, a dense series of 2D slices can be acquired from adjacent planes within a 3D dataset. Methods that fall into this category include magnetic resonance imaging (MRI), tomography, and confocal microscopy. In instances where multiple parallel slices are available, several techniques have been proposed for volumetric reconstruction [Boissonnat 1988; Geiger 1993; Bouteiller and Baudry 2002; Barequet and Sharir 1994; Meyers et al. 1992; Fix and Ladner 1998; Barequet et al. 1996; Barequet et al. 2003].

Unfortunately, costly resources are needed for most data acquisition technologies that acquire densely sampled slices of volumetric data. Since 3D datasets are not readily available, they will not be further considered in this thesis. Instead, the focus will be on algorithms for extrapolating information from easily obtainable 2D images.

2.5 2D Shape Perception

For the majority of the existing literature in shape approximation, the objective is to build an accurate geometric reconstruction of an existing 3D object. In contrast, for a single input image with discrete, uncorrelated profiles, there is insufficient information for extrapolating an accurate 3D model—the problem is, by nature, heavily unconstrained. A more appropriate objective is to approximate a particle shape that exhibits perceptually similar profiles to those seen in the input. Since perceptual similarity can be a subjective measure, we need some way of quantifying the results of various shape approximation algorithms.

In order to compare the similarity of profiles, a wide variety of metrics are available. For an extensive overview of such techniques, I refer the reader to the summary by Loncaric [1998]. Unfortunately, few existing 2D shape comparisons metrics are perceptually motivated, due in part to the complexity of the human visual system, which is not entirely well understood. The shape metrics that will I consider in this thesis are motivated by prior research that has identified curvature as an important characteristic discerned by low-level human perceptual mechanisms [Dobbins et al. 1987; Or and Zucker 1989; Ben-Shahar et al. 2003].

Specifically, I compare profiles using metrics of total curvature magnitude [Gardner et al. 2004; Loncaric 1998] and area per square perimeter [Li et al. 2003], which is sometimes referred to as *circularity* due to the fact that the metric yields a maximal value for circular shapes.

These simple metrics cannot be expected to reliably capture all psychophysical aspects of profile similarity, so direct user evaluation also plays a valuable role in shape comparison. A number of authors have used direct psychophysical evaluation to assess visual fidelity [Meyer et al. 1986; Rushmeier et al. 2000; Pellacini et al. 2000; McNamara et al. 2000]. Of particular interest in this area is work by Watson et al., who rate the performance of automated metrics for predicting experimental results [2000; 2001]. The user experiments described in their research measure *naming time*—the elapsed time from when an object is shown to a subject until it is named. However, this metric is inappropriate for gauging the similarity of abstract shapes.

Martens and Myszkowski use a comparable methodology for evaluating the performance of perceptually-motivated appearance metrics [1998]. In their psychophysical experiments, users were asked to numerically rank the visual similarity of image pairs. This thesis applies comparable methods for assessing the effectiveness of automated techniques for predicting the perceptual similarity of 2D shapes.

2.6 Summary

This chapter has described an extensive collection of prior research in the areas of computer vision, computer graphics, stereology, and human perception. In the following chapters, I introduce novel algorithms that build upon this foundation of prior work in order to estimate particle shapes, recover particle distributions, and synthesize solid textures that appear perceptually similar to example 2D texture inputs.

Chapter 3

Particle Shape Estimation

A single slice through an aggregate material can reveal any number of 2D profiles. Since each profile results from a slice through a different particle, there is insufficient information to reconstruct any single particle exactly. The problem is, by nature, heavily unconstrained. Without any *a priori* knowledge about the particles, we need to rely on limited assumptions about their shape and distribution.

This chapter considers four different methods for approximating 3D shapes from 2D profiles. The *spherical deformation* and *generalized cylinder* models described below were introduced in prior publications; the *constructive solid geometry* and *morphed generalized cylinder* methods are my own contribution to the problem.

Section 3.6 presents a comparison of these four techniques using automated, perceptually-motivated metrics, as well as direct user evaluation. This analysis is similar to that of prior authors who assess the performance of automated statistics as predictors of visual fidelity [Martens and Myszkowski 1998; Watson et al. 2000; Watson et al. 2001].

The objectives in this chapter are twofold. The first goal is to determine which algorithms are most effective for generating 3D particle shapes that exhibit 2D profiles with a comparable appearance to the input data. The second objective is to assess how reliably the automated metrics serve as predictors of the psychophysical evaluation.

Method	**Spherical Harmonics**	**Constructive Solid Geometry**	**Generalized Cylinder**	**Morphed Generalized Cylinder**
Number of input profiles	1 or more	3	2	3 or more
Treatment of input profiles	Frequency data only	orthogonal silhouettes	orthogonal slices	orthogonal slices

Figure 3-1: Summary of input parameters and data assumptions for shape approximation methods

3.1 Overview of Shape Estimation Methods

For all of the methods considered here, I assume that the particle shapes are genus zero—i.e., the surfaces are topologically equivalent to a sphere. Furthermore, it is reasonable to assume that the largest visible profiles result from slices near the center of the largest particles within the volume. Depending on the algorithm, one or more characteristic profiles are selected from an image and used as input. The profiles are chosen by hand and are expressed using a polygonal representation.

The methods differ in the number of profiles that are used as input and how the profiles are incorporated into a 3D shape, as summarized in Table 3-1. For the spherical harmonics method, only the frequency information of the 2D curve is considered, and the original input profiles are not preserved in slices through the synthesized particle shape. Any number of profiles can be used as input. For the results shown in this thesis, three curves are always used.

The constructive solid geometry algorithm takes exactly three profiles as input. In this case, I assume that the the profiles represent the silhouette of the object as seen from three orthogonal views. A geometric model is constructed in a manner that preserves the input profiles in the silhouette. However, the algorithm does not necessarily preserve the input curves in slices through the novel particle.

For the generalized cylinder approach, two input profiles are used. The first input curve and half of the second are incorporated directly into the geometry such that the they can be seen in slices through the resulting particle. The remaining half-profile is ignored.

Finally, for the morphed generalized cylinder method, three or more input profiles can be used. The examples shown in this thesis use exactly three inputs, which are preserved in the construction process such that the same three profiles can be seen in slices through the resulting geometric shape.

Each of these four methods will be discussed in greater detail in the following sections.

3.2 Spherical Deformation Models

A number of papers in both the graphics and stereological literature use spherical deformation models as a foundation for representing particle shapes [Edvardson and Smedby 2002; Weistrand 2001; Hobolth and Jensen 2002; Hobolth 2003]. With these methods, each particle is modeled as a sphere that is deformed inward or outward from its center. Particles created with this technique are restricted to being *star-shaped*—i.e., all points on the surface are visible from a single point within the particle.

In two dimensions, star-shaped profiles can be expressed as a radius-vector function $r(\theta)$, $0 \leq \theta < 2\pi$, which indicates the distance from the center of the profile to its boundary in each radial direction θ. This function can then be expressed as a Fourier series expansion,

$$r(\theta) = \frac{b_0}{\sqrt{2\pi}} + \sum_{m=1}^{\infty} b_m^c \frac{1}{\sqrt{\pi}} \cos(m\theta) + \sum_{m=1}^{\infty} b_m^s \frac{1}{\sqrt{\pi}} \sin(m\theta)$$

with the Fourier coefficients,

$$\begin{aligned} b_0 &= \int_0^{2\pi} r(\theta)\frac{1}{\sqrt{2\pi}} d\theta \\ b_m^c &= \int_0^{2\pi} r(\theta)\frac{1}{\sqrt{\pi}}\cos(m\theta) d\theta, \quad m \geq 1 \\ b_m^s &= \int_0^{2\pi} r(\theta)\frac{1}{\sqrt{\pi}}\sin(m\theta) d\theta, \quad m \geq 1 \end{aligned}$$

The radial basis functions corresponding to b_m^c and b_m^s are shown in Figure 3-2. Figure 3-3 shows three example profile shapes approximated with Fourier expansion series having five, ten, and fifteen terms. Note that the sharp corners become rounded—particularly when few terms are used—but the general shape of each profile is maintained.

Similar to the two-dimensional case, in three dimensions, a star-shaped particle can be expressed as a spatial radius-vector function $d(\theta,\phi)$, $0 \leq \theta < 2\pi$, $0 \leq \phi \leq \pi$. This function indicates the distance from the center of the particle to its boundary in each polar direction (θ,ϕ). Just as $r(\theta)$ can be reexpressed as a Fourier series expansion, $d(\theta,\phi)$ can be rewritten as a Fourier-Legendre series expansion

$$d(\theta,\phi) = 1 + \sum_{n=1}^{\infty} \sum_{m=-n}^{n} a_n^m Y_n^m(\theta,\phi)$$

where Y_n^m, $(n \geq 1, -n \leq m \leq n)$, are spherical harmonics, which can be though of as a basis for Fourier series expansion in three dimensions. The spherical harmonics can be expressed as follows:

$$Y_n^m(\theta,\phi) = \sqrt{\frac{2n+1}{4\pi}\frac{(n-m)!}{(n+m)!}} P_n^m(\cos\theta)e^{im\phi}$$

where P_n^m are the associated Legendre functions of the first kind [Weisstein 2003]. The spherical harmonic bases are shown in Figure 3-4

Hobolth and Jensen create 3D particles based on 2D observations by relating the coefficients b_0, b_m^s and b_m^c of the 2D Fourier expansion to the coefficients a_n^m of the 3D Fourier-Legendre expansion [2002]. This relationship is based on the assumption that the 2D Fourier coefficients are mutually independent and that they form a stationary Gaussian process on the underlying circular domain—i.e., the covariance between any two points depends only on the angle between them. The 2D profile can be normalized via isotropic scaling such that $b_0 = \sqrt{2\pi}$. We assume that the remaining Fourier coefficients are distributed according to a Gaussian function with mean 0 and variance κ_m:

$$\kappa_m = \sum_{n=1}^{N} \frac{b_m^s + b_m^c}{2N}$$

As with the Fourier terms in two dimensions, we assume that the 3D Fourier-Legendre coefficients, a_n^m, are mutually independent and form a stationary Gaussian process on the

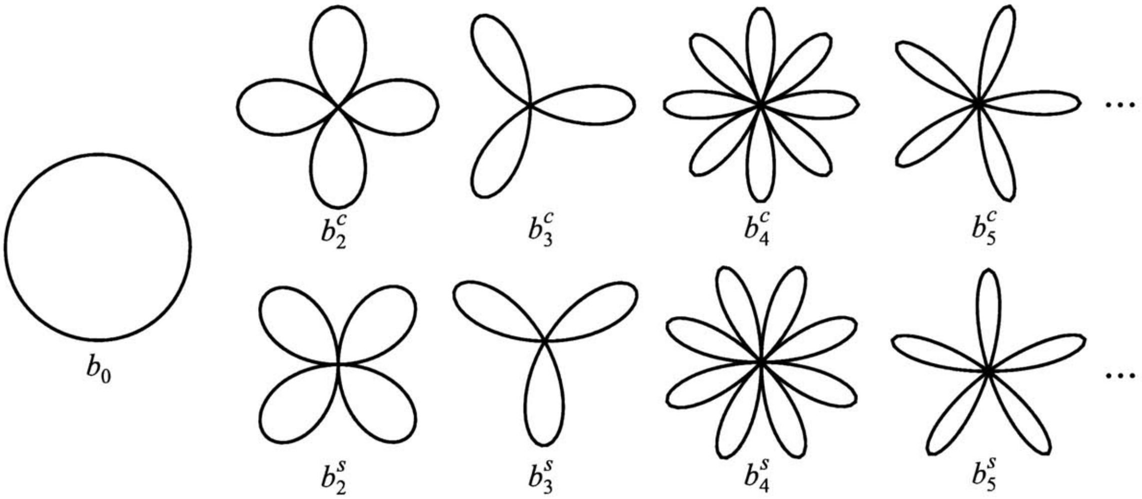


Figure 3-2: Radial basis functions for Fourier series expansion.

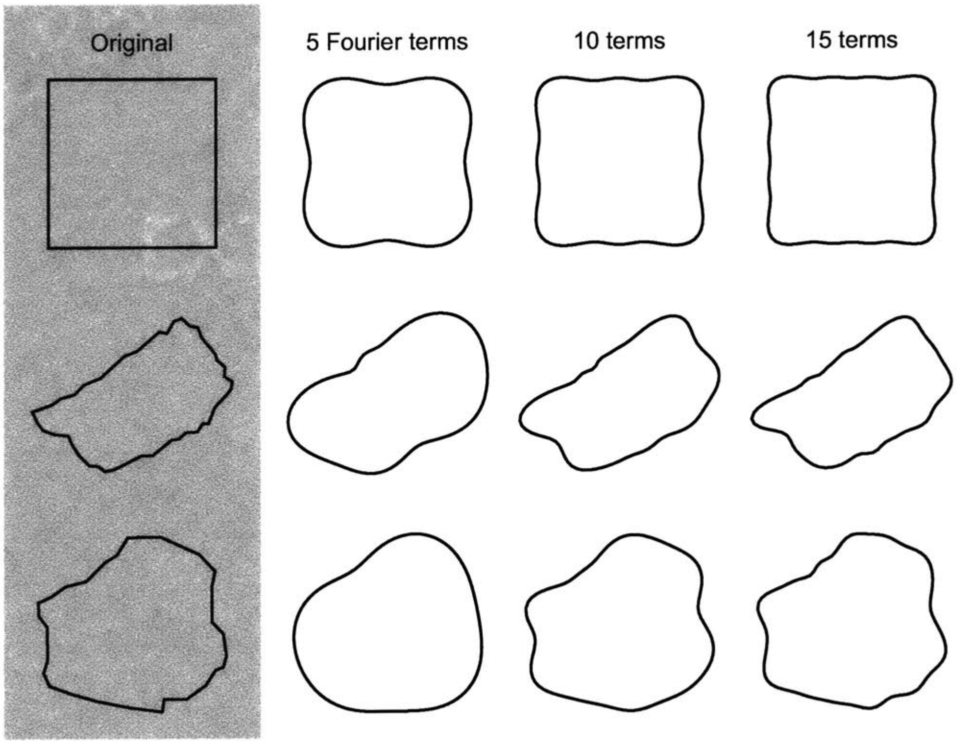


Figure 3-3: The original profiles on the left are approximated with Fourier series expansion with five, ten, and fifteen terms. General shape is maintained, but corners become rounded when few terms are used.

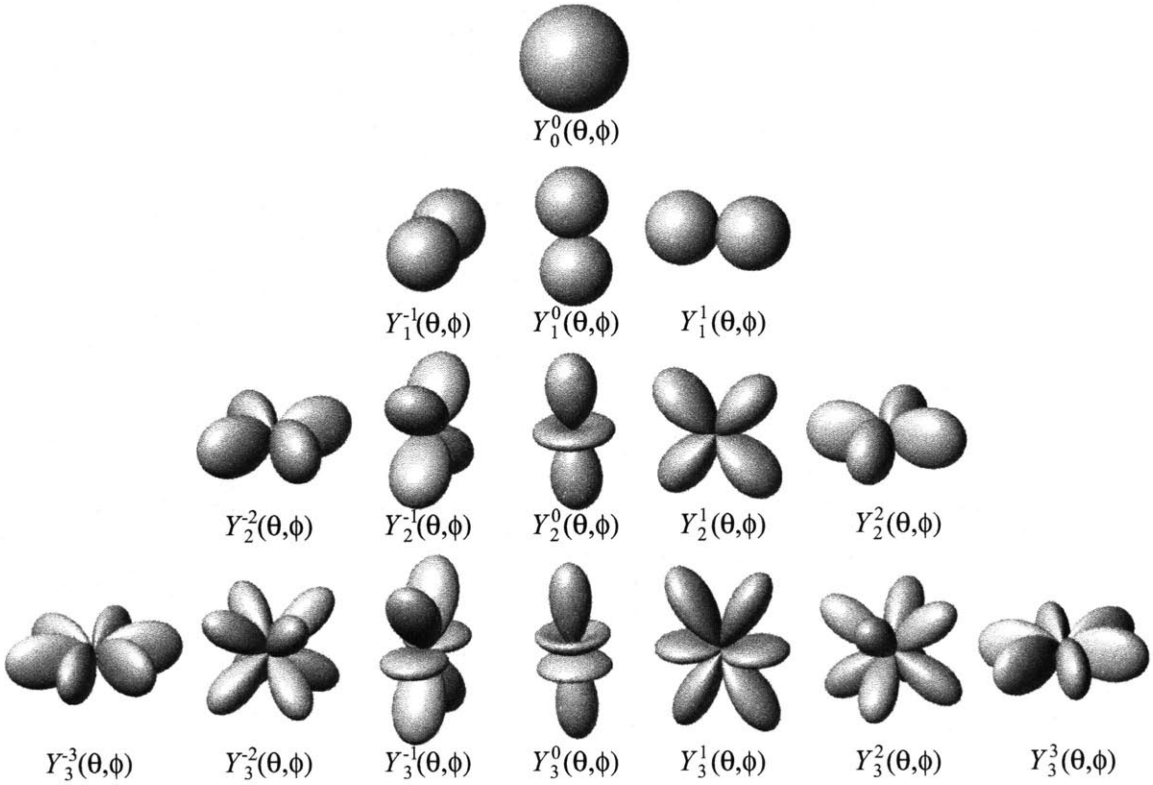


Figure 3-4: Spherical harmonic basis functions for Fourier-Legendre series expansion.

spherical domain with mean 0 and variance λ_n. Hobolth and Jensen demonstrate that the coefficients a_1^m can be ignored, and that the remaining variance values are related by the expression

$$\kappa_m = \sum_{n=m}^{\infty} \frac{2n+1}{2} \frac{(n-m)!}{(n+m)!} P_n^m(0)^2 \lambda_n, \quad n \geq 2.$$

This relationship relies on the assumption that the particle shape frequencies are isotropically distributed in three dimensions. After solving for the variances λ_n, Fourier-Legendre coefficients can be chosen according to the Gaussian distribution, resulting in particles that exhibit profiles with comparable frequencies to the input shape. It should be noted that any number of particles can be generated with the recovered variance values.

This approach for particle modeling is mathematically rigorous, but ultimately attempts to represent a complex shape with a small number of parameters by assuming isotropism, mutually independent frequency coefficients, and frequency stationarity on the sphere. As a result, the method is poorly suited for particles that exhibit sharp corners or distinct

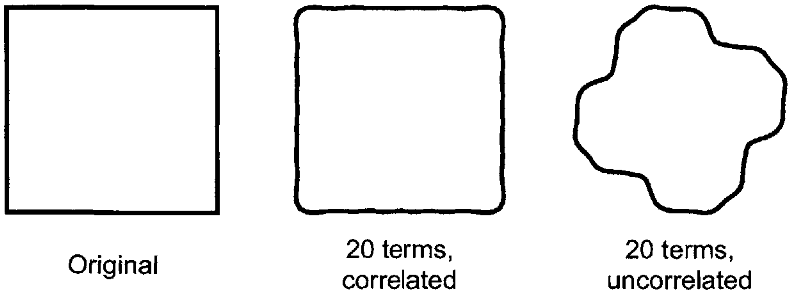


Figure 3-5: The decorrelation of frequency terms in a Fourier expansion series can cause deformations that impact the characteristic appearance of a 2D shape.

features.

The assumption that coefficients are mutually independent results in smoothing of sharp angles or other features that may characterize the input. Figure 3-5 illustrates how dramatic changes in appearance can result from the decorrelation of Fourier series expansion terms in two dimensions. The square shown on the left can be effectively approximated with 20 terms, as shown in the middle. If the frequencies are decorrelated, as shown on the right, the sharp corners disappear. The four-way symmetry is still apparent, but the shape loses many of the features that characterize its appearance. It should be noted that the Fourier coefficients in the middle and right images share identical variance values, κ_m. This same type of decorrelation results when shapes are approximated in three dimensions.

In the work by Hobolth, the author uses only ten expansion terms to approximate each shape, noting that the variance of the terms becomes excessively noisy after that point [2003]. For our results, we use 30 expansion terms, but still fail to capture the sharp features of the input profiles.

One inherent drawback of any spherical deformation method is the requirement that all resulting particles must be star-shaped. This constraint becomes increasingly problematic for long, slender particles with anisotropic shape characteristics.

3.3 Constructive Solid Geometry

As described in Chapter 2, if a static object is viewed from multiple calibrated viewpoints, then volume-carving methods can be used to approximate the geometry [Martin and Aggarwal 1983]. In contrast, in a 2D sample of an aggregate material, only one slice through any individual particle can be seen. In order to constrain the reconstruction problem, I select three characteristic profiles, which I assume are silhouettes of a single target particle shape, projected orthographically onto three orthogonally oriented viewing planes. I align the three profiles using a heuristic to determine a likely correlation, and then apply volume-carving techniques to yield an approximate 3D shape.

This approach makes extensive use of constructive solid geometry (CSG) modeling methods [Mortenson 1999]. The objective is to build a particle such that the orthographic projection of the 3D shape in the x, y, and z directions yields scaled versions of the three input profiles. The basic approach is to extrude each of the three input profiles and then arrange the three extruded volumes orthonormally. The CSG intersection of the volumes yields the particle. This process is shown in Figure 3-6.

In order to maintain the shape of the input profiles during the CSG intersection operation, the profiles must first be transformed such that their bounding boxes have the dimensions $a \times b$, $b \times c$, and $a \times c$ for some a, b, and c. In the first preparation step, each profile is rotated such that profile's second-order moment is aligned with the x-axis. The three profiles are then sorted according to the aspect ratio of their bounding boxes. This sorting yields profiles c_0, c_1, and c_2, with decreasing aspect ratio c_{ny}/c_{nx} where c_{nx} and c_{ny} are respectively the width and height of the bounding box of c_n. In order to minimize the change in aspect ratio of the input profiles, I select bounding box extents a, b, and c as follows:

$$\begin{aligned} a &= c_{0x} \\ b &= c_{0y} \\ c &= c_{0x}c_{2y}/c_{2x} \end{aligned}$$

Finally, I scale the profiles such that these dimension constraints are met, extrude each profile in the direction perpendicular to the image plane, and arrange the extruded volumes such that the bounding boxes are aligned as shown in Figure 3-6(b). The CSG intersec-

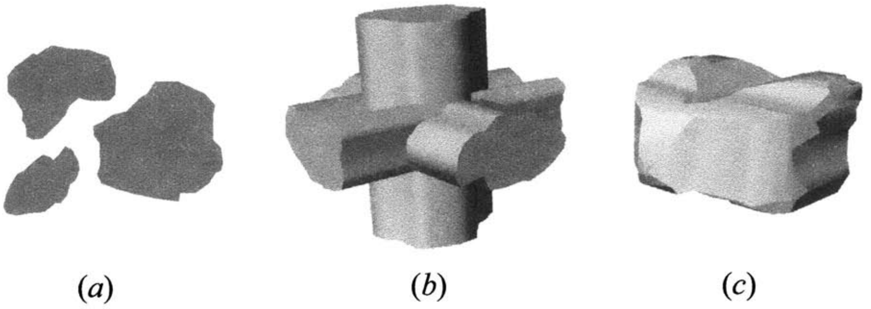


Figure 3-6: Constructive solid geometry (CSG) method for particle construction. Input profiles (*a*) are extruded and arranged orthonormally (*b*). The CSG intersection of the three extruded solids (*c*) yields a plausible 3D particle shape.

tion of the extruded volumes is guaranteed to exhibit the three input profiles when viewed orthographically, regardless of the convexity of the input shapes. However, this does not necessarily mean that the input profiles are preserved in slices through the novel particle.

The particles generated by this process tend to exhibit cross-sections that are perceptually similar to the input profiles, with the exception of a few long, slender shapes or hard corners that may be introduced by the process. It should be noted that at every point on the particle, the surface normal will be perpendicular to either the x-, y-, or z-axis. This results in a faceted look that may make this modeling approach unsuitable for applications where the complete 3D particles are viewed directly.

3.4 Generalized Cylinder Model

Dischler and Ghazanfarpour model macroscopic particles using a generalized cylinder method [1999]. The inputs to the algorithm are two representative particle profiles—a sweep curve s and a base curve b. The sweep curve is formed by cutting a profile in half along a vertical line that passes through its centroid. The other half of the profile is discarded.

Dischler and Ghazanfarpour choose to represent these two curves as radius-vector functions, $s(\theta)$ and $b(\theta)$, $0 \leq \theta < 2\pi$, which are expressed as interpolating cubic splines with 32 vertices at regularly spaced angular intervals. When using this representation, profile shapes are constrained to being star-shaped, as are novel particles generated by the method.

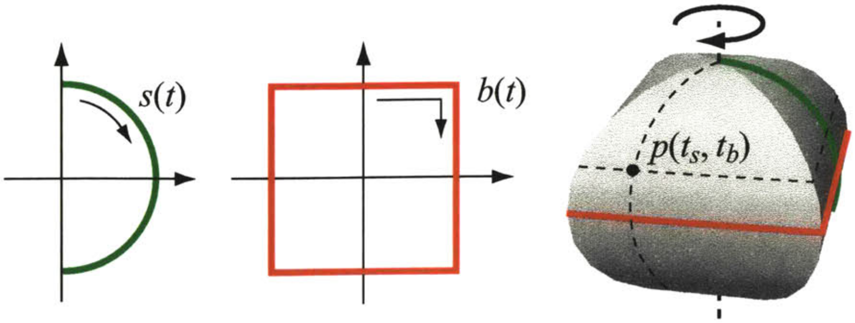


Figure 3-7: Generalized cylinder method for particle construction. The construction curves shown on the left are used to create the particle shape on the right.

To enable the algorithm to operate on a larger class of geometric inputs, I relax this constraint by instead expressing s and b parametrically as a function of t, where $0 \leq t \leq 1$:

$$s(t) = \begin{bmatrix} x_s(t) \\ y_s(t) \end{bmatrix} \text{ and } b(t) = \begin{bmatrix} x_b(t) \\ y_b(t) \end{bmatrix}$$

Using the curves s and b, a 3D particle can be constructed by sweeping the curve s around the base profile b to form a generalized cylinder as shown in Figure 3-7. The axis of rotation is defined to be perpendicular to b, passing through its centroid. Curve s is then oriented perpendicular to curve b such that its endpoints lie on the axis of rotation and s is tangent to $b(t)$ at $t = 0$. The particle shape is defined by the set of all points $p(t_b, t_s)$, $0 \leq t_b < 1$ and $0 \leq t_s \leq 1$, such that

$$p(t_b, t_s) = \begin{bmatrix} x_b(t_b) x_s(t_s) / \|b(0)\| \\ y_b(t_b) x_s(t_s) / \|b(0)\| \\ y_s(t_s) \end{bmatrix}$$

where the normalization term $\|b(0)\|$ is used to insure that curve s remains tangent to b.

Figure 3-8 shows some example particle shapes generated by the generalized cylinder algorithm. The process is intuitive and can be used to generate a variety of geometric shapes; however, the particles created by this method tend to have a synthetic appearance, as they exhibit clear symmetry around the axis of rotation.

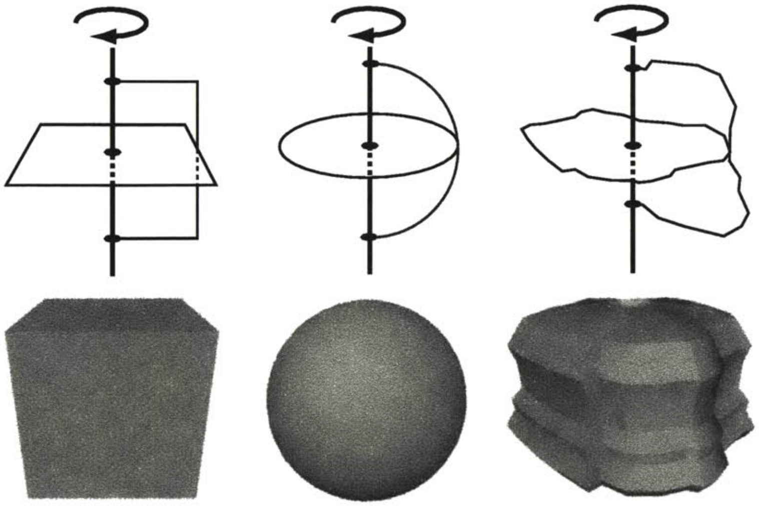


Figure 3-8: Example shapes created by the generalized cylinder method. The construction curves shown in the top row result in the particle shapes in the bottom row.

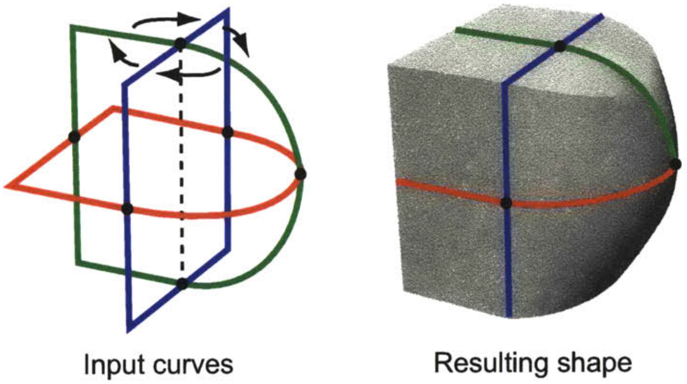


Figure 3-9: Morphed generalized cylinder method for particle construction. The base curve (red) and morph curves (green and blue) result in the shape shown at the right.

3.5 Morphed Generalized Cylinder Model

The final particle generation algorithm that we consider here is a novel approach introduced in this thesis. The method is motivated by the idea of establishing a set of orthogonally oriented wireframe constraints, and then creating a naturally shaped particle that smoothly interpolates between the constraints. The particle is created in a manner similar to that described for the Generalized Cylinder Method, with the exception that the generatrix curve is morphed from one constraint to the next as it is swept along the directrix, or base curve.

To begin, the input profiles are reoriented and sorted as indicated in Section 3.3. As described in the previous section, the base profile b—shown in red in Figure 3-9—is used as the directrix. The remaining curves, which we will refer to as morph curves, collectively take the place of the sweep curve s used in the generalized cylinder algorithm. If n total profiles are used as input, then morph profiles m_0 through m_{n-2} are oriented perpendicular to the base curve with equal angular spacing around the axis of rotation. For the examples shown in this thesis, three total curves are used, and the profile with the aspect ratio closest to one is selected as the base curve. However, it should be noted that different particle shapes can be generated either by using a different number of input curves or by considering alternate permutations of the input profiles.

The next step is to scale each profile to guarantee intersections at the black points shown in Figure 3-9. Meeting these constraints can be a difficult task if profiles are modified with uniform linear transformations. The algorithm does not restrict the profiles to be star-shaped, so the optimization space can be discontinuous and arbitrarily poorly behaved. Instead, I choose to meet the constraints by cutting the morph profiles at each point where I expect two profile curves to meet. I then scale each region of the profile independently.

To insure that each morph curve lies tangent to the base curve, morph profiles are scaled inward or outward from the axis of rotation. Next, I scale the curves along the axis of rotation to insure a common intersection at the poles.

In order to apply the morphed generalized cylinder algorithm, we need to establish a parameterization that allows for interpolation between successive morph curves. I have chosen to parameterize each profile according its normalized arc length [Verth and Bishop 2004].

The cumulative length along a curve $s(t) = [x(t), y(t)]^T$, $0 \leq t \leq 1$ can be expressed as follows:

$$l(t) = \int_0^t \sqrt{x'^2(t) + y'^2(t)}dt$$

To reparameterize the curve by its normalized arc length, we create a new expression $s_R(t)$ such that

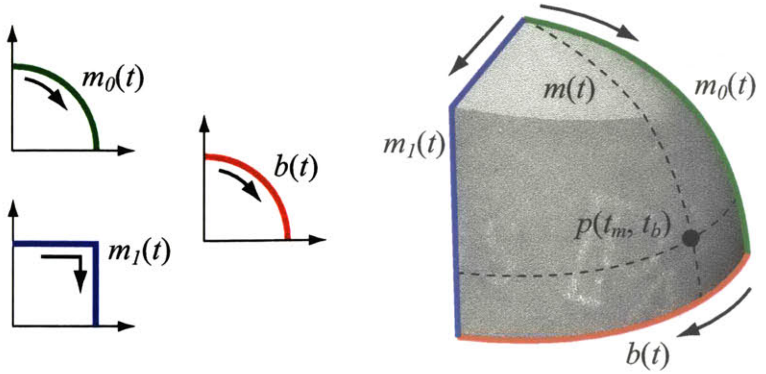


Figure 3-10: Construction process for the morphed generalized cylinder method. The construction curves shown on the left define the portion of the particle shape shown on the right.

$$s_R(t) = s(l^{-1}(l(1)t)),$$

noting that $l(1)$ is the total length of curve s. For the remainder of this section, we will assume that all curves have been reparameterized according to their arc length.

To demonstrate the construction process, we consider, without loss of generality, the region of the surface enclosed by three curves—morph curves $m_0(t)$ and $m_1(t)$ and a portion of the base curve $b(t)$, as shown in Figure 3-10. I express the three curves parametrically, parameterized according to their arc length, such that for each curve region, $0 \leq t \leq 1$:

$$m_0(t) = \begin{bmatrix} x_0(t) \\ y_0(t) \end{bmatrix}, m_1(t) = \begin{bmatrix} x_1(t) \\ y_1(t) \end{bmatrix} \text{ and } b(t) = \begin{bmatrix} x_b(t) \\ y_b(t) \end{bmatrix}.$$

The morphed generalized cylinder algorithm creates a 3D particle by interpolating the shape of the morph profiles. We use the morph curve, m, as the generatrix as it is swept along the directrix, or base curve, b.

At an intermediate position t_b between morph curves $m_0(t)$ and $m_1(t)$, we linearly interpolate to define the morph curve,

$$m(t_b, t) = \begin{bmatrix} x_m(t_b, t) \\ y_m(t_b, t) \end{bmatrix} = (1 - t_b)m_0(t) + t_b m_1(t).$$

With the help of this expression, we can define our desired surface as the collection of all points $p(t_b,t_m)$, $0 \leq t_b \leq 1$ and $0 \leq t_m \leq 1$, such that

$$p(t_b,t_m) = \begin{bmatrix} x_b(t_b)x_m(t_b,t_m)/\|m(t_b,1)\| \\ y_b(t_b)x_m(t_b,t_m)/\|m(t_b,1)\| \\ y_m(t_b,t_m) \end{bmatrix}. \tag{3.1}$$

By interpolating the profiles individually over each region enclosed by three curves, we are able to guarantee that all of the initial constraints are met precisely. The normalization term $\|m(t_b,1)\|$ in Equation 3.1 is necessary for enforcing this condition. An example shape generated by this process is shown on the right side of Figure 3-9.

3.6 Algorithm Comparison

To evaluate the relative performance of each of the four proposed algorithms, I consider the use of automated comparison metrics based on perceptually motivated criteria, as well as a user study in which participants attempt to discriminate between original and synthetic shape data.

The particle shapes used as a basis for comparison were obtained from 3D scans of four different rocks that were deliberately selected to span a wide variety of input appearances. The *smooth* particle is a rounded, convex surface. The *elongated* shape is more elliptical with a few sharp edges. The*faceted* model is nearly convex and exhibits a number of sharp edges. Finally, the *complex* model is highly non-convex and is the only one of the four meshes that is not star-shaped.

Each particle was approximated with each of the four reconstruction methods, resulting in a total of 20 particle shapes, including the originals. The input to the algorithms came from three orthogonally oriented profiles taken from the original particles. The orthonormal basis was selected to correspond approximately with each particle's first and second moments of inertia. Each of the 20 polygon meshes was scaled uniformly to insure that all particles have the same volume. Figure 3-11 shows the original particle shapes and the decoy particles generated by each algorithm. Randomly selected central slices through each particle are shown in Figure 3-12.

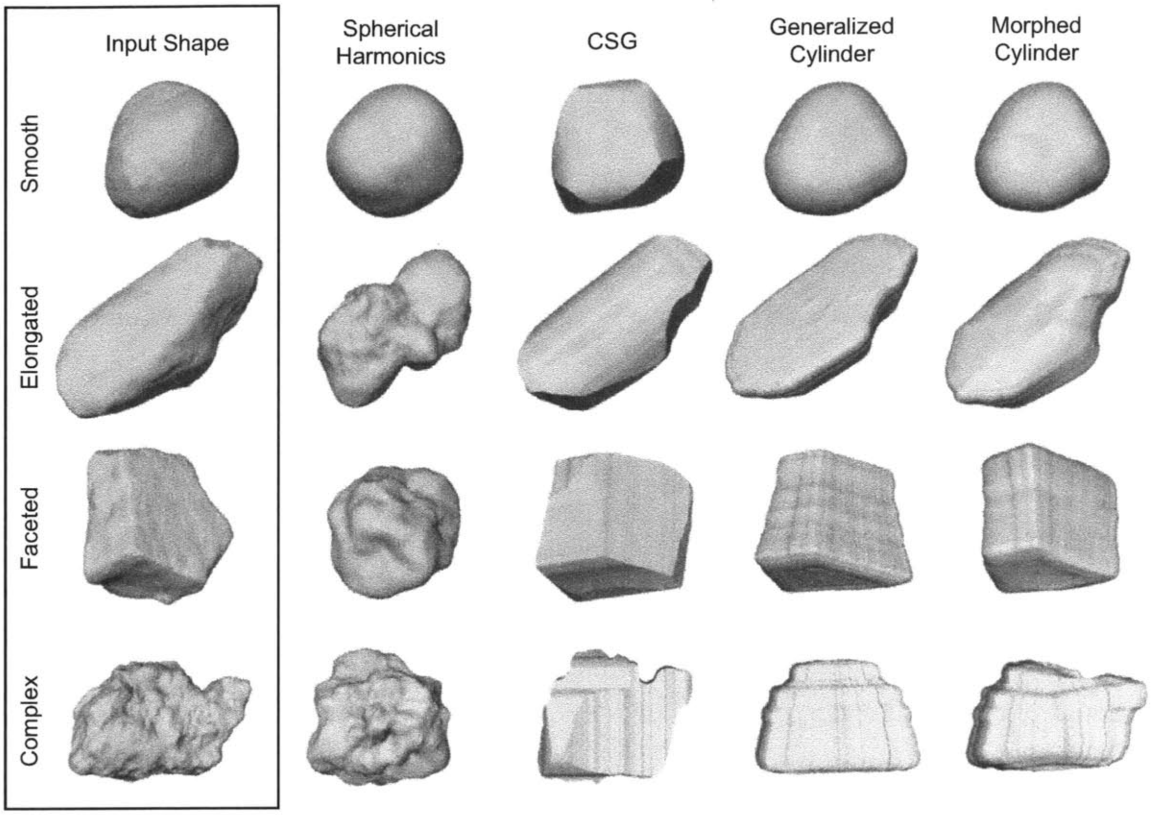


Figure 3-11: In the left column are the four scanned particle shapes used as the inputs to the shape approximation algorithms. To the right are the decoy particle shapes created by each of the four particle approximation methods.

3.6.1 Automated Performance Evaluation

For the purposes of this research, it is important that the synthetic particle shapes have a plausible appearance and that the particles yield profiles that are perceptually similar to the example data. Thus, I choose to evaluate the resulting particle shapes based on metrics that are motivated by human perception. Curvature in two dimensions is theorized to be an important characteristic identified by low-level human perceptual mechanisms [Dobbins et al. 1987; Or and Zucker 1989; Ben-Shahar et al. 2003]. The two metrics that I consider here are the total curvature magnitude of a profile and the measure of area per perimeter squared, also known as circularity [Gardner et al. 2004; Loncaric 1998; Li et al. 2003]. Each is an indicator of the visual complexity of a curve.

The total curvature magnitude is computed by summing the absolute value of the change in angle at each point along the curve. If the profile is convex, then this value sums to 2π.

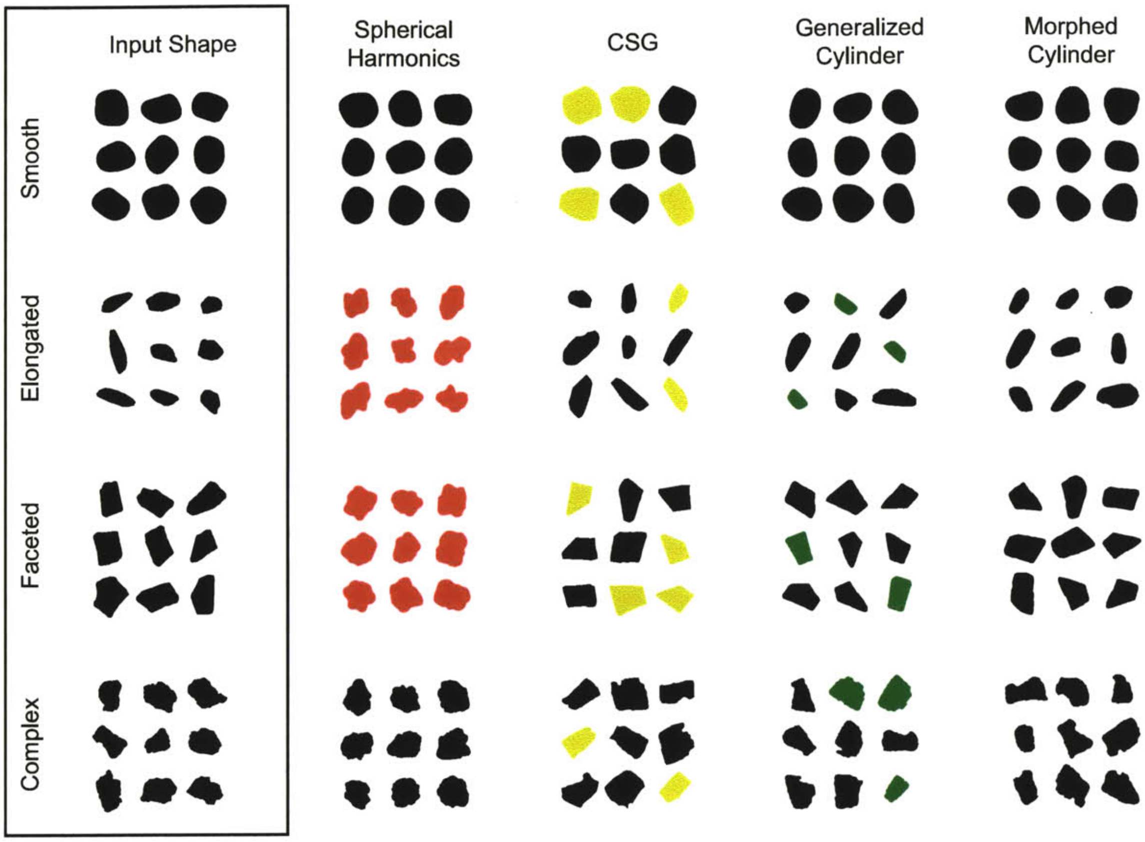


Figure 3-12: This table shows randomly selected central profiles for each original and synthesized particle shape. Profiles that are poorly matched to the original data are highlighted in color. Red profiles are overly curved or blobby, yellow profiles exhibit excessively sharp corners, and green profiles show symmetry that is not present in the original data.

Higher values indicate a higher degree of shape concavity. The circularity metric can range from a value of zero for degenerate shapes with zero area to a maximum of $1/4\pi$, which can only be achieved by a perfect circle.

To produce the values shown in Figure 3-13, I created a utility that sliced through each particle 10,000 times at uniformly distributed random orientations. I applied the automated metrics to the profiles and averaged the resulting values.

For the two graphs in the top row, the particles were always sliced through their center of mass. This yields profiles with relatively uniform statistics but fails to capture anomalous appearance characteristics that may occur at slices that are more distant from the particle center.

In the second row, the values were obtained by slicing particles at both a random ori-

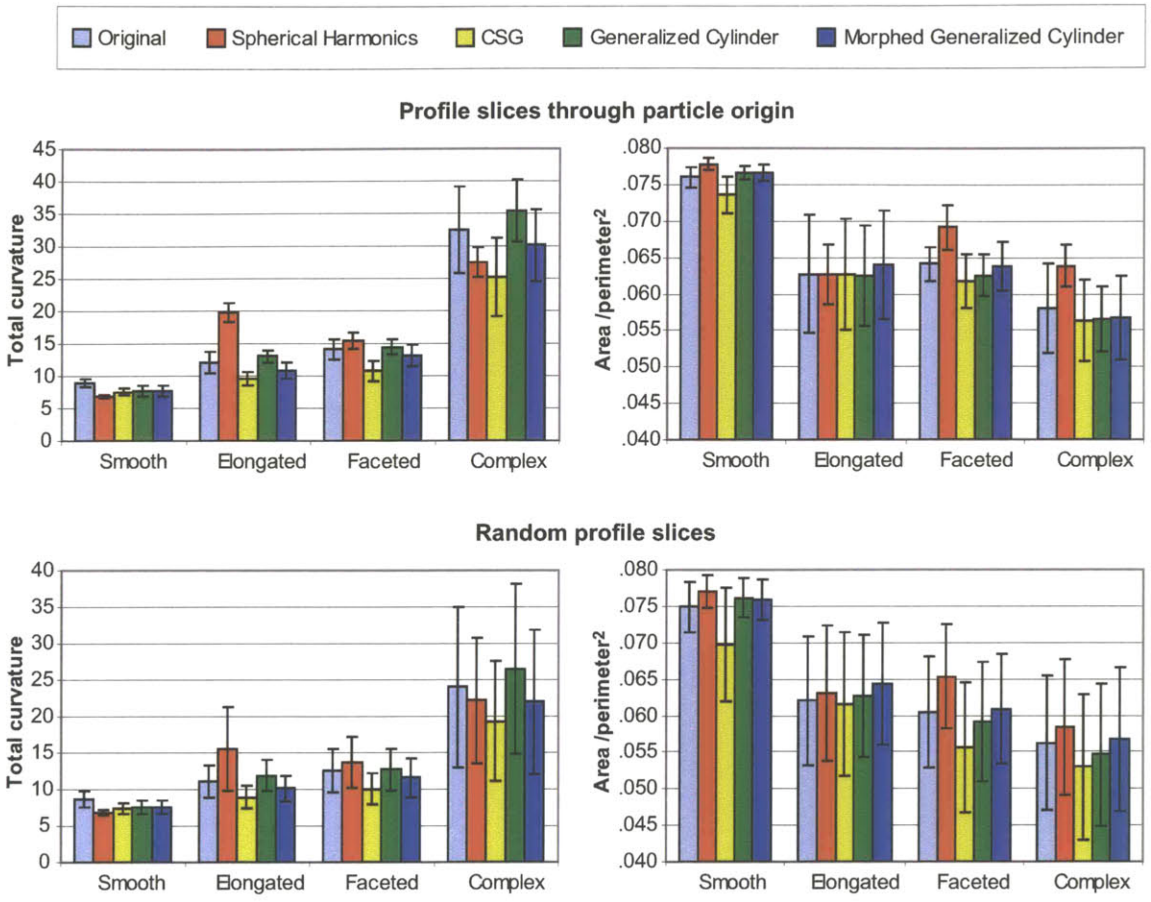


Figure 3-13: Original and synthetic particle shapes are compared via two measures of 2D visual complexity: total curvature and area/perimeter^2. These charts show the mean and standard deviation values obtained from 10,000 random slices through each particle shape. The top row shows values obtained from slices passing through the center of the particles. In the second row, slices are randomly selected.

Metric	**Profile Slice**	**Algorithm**			
		Spherical Harmonics	CSG	Generalized Cylinder	Morphed Generalized Cylinder
Area/Perimeter^2	**Center of Mass**	5.0%	2.5%	1.6%	1.4%
	Random	4.0%	5.4%	1.9%	1.6%
Total Curvature	**Center of Mass**	27.8%	20.6%	8.2%	9.5%
	Random	19.5%	18.6%	7.8%	9.3%

Figure 3-14: Average deviation from original data values for each algorithm, as measured for each of the two metrics and profile slicing techniques.

entation and a random offset from the particle's center of mass. The values obtained in this manner statistically represent the profiles that are visible in a slice through a volume of particles embedded in an opaque medium, much like the image shown in Figure 1-3(a). As expected, these profiles exhibit a greater diversity of appearances, resulting in higher standard deviation values.

To analyze the significance of these results, I apply analysis of variance (ANOVA), which measures the probability, p, that the results are a product of the *null hypothesis*—i.e., the probability that there is no difference between the algorithm performance, and the variability shown in the results is the product of chance [Cobb 1998]. For all of the datasets considered in this chapter, I set a threshold of significance at $p \leq 0.01$. For each of the four tables in Figure 3-13, a two-way analysis of variance (ANOVA) reveals a significant dependence on shape and choice of algorithm ($p < 0.001$), and also indicates significant interaction between the two factors ($p < 0.001$).

Figure 3-14 summarizes the shape statistics, showing the average deviation from the original data for each of the shape approximation algorithms. Based on these values, we see that the generalized cylinder and morphed generalized cylinder methods perform better than the spherical harmonics or CSG methods for matching the given shape statistics. However, there is no consensus on a total ordering of the four algorithms.

These methods attempts to capture perceptually meaningful criteria with a simple metric, but there are some drawbacks to the approach. The total curvature metric is sensitive to shape tessellation. Highly tessellated profiles may include high-frequency information that increases the total curvature magnitude without greatly contributing to the overall appearance of the shape. To a lesser extent, the perimeter calculation is also susceptible to this problem. Furthermore, as shown in Figure 3-12, some of the characteristics that distinguish the decoy profiles from the original profiles include the presence or absence of sharp corners, as well as the presence of symmetry. These characteristics, while easily seen by a human observer, are not captured by the statistics considered here. To obtain more definitive results, I consider the input of human observers, as described in the following section.

3.6.2 User Study

To obtain a more disciplined evaluation of the four proposed algorithms for shape approximation, I authored a user study in which participants attempt to distinguish between original and synthetic texture data. Users were shown a series of images in a computer-based test in which they were asked to evaluate shape similarity.

Data

All of the data used in the study was synthetically produced in order to keep the images free of noise, which would otherwise lend a distinctive appearance to the physically captured images. The study used the same four particle shapes described previously and shown in Figure 3-11.

For each shape, a synthetic volume of 1000 cubic units was populated with particles of constant size. The volumes for the *smooth*, *elongated*, and *faceted* datasets contain 4000 particles, each of which has a size of 0.125 cubic units. Thus, 50% of the volume is occupied. The datasets for the *complex* particle shapes contain 3500 particles, yielding volumes that are 44% occupied.

Of course, most real-world volumes do not exhibit particles that are all of the same shape and size. For the purposes of the user study, I deliberately eliminate the presence of other variables to allow users to focus on shape. However, it should be recognized that different particle distributions may play a role in shape perception.

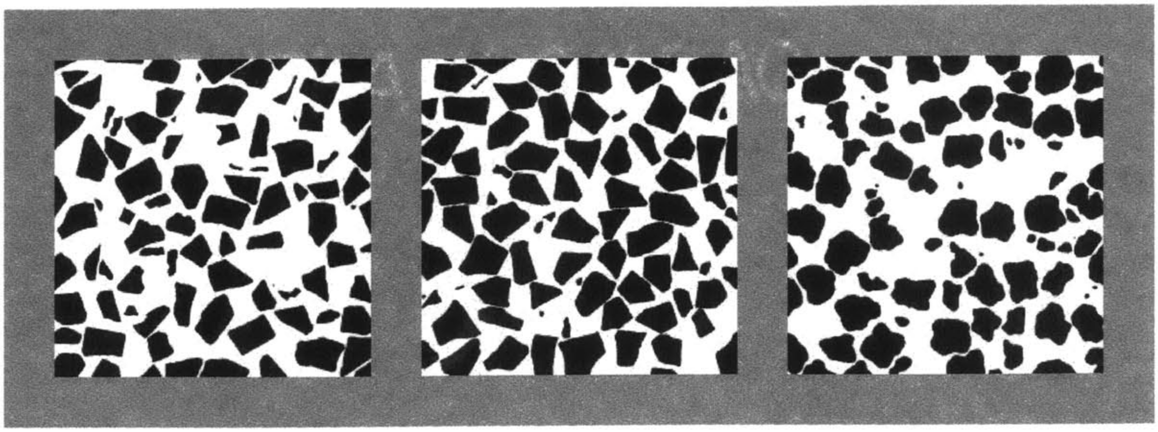


Figure 3-15: Example screen from user study. Users are asked to select the image that is least similar to the image in the center. Here, a decoy image is shown on the far right.

Procedure

To evaluate the effectiveness of the methods, study participants were given a computer-based test to compare image similarity. Users were shown a series of screens, each containing three images in a row, as depicted in Figure 3-15. In most instances, the center image and one of the adjacent images were generated by taking a slice through one of the four original volumes. The third image was generated by taking a slice through a decoy volume, generated using one of the four proposed methods. Users were asked to click on the image that appeared to be the least like the center image and were explicitly asked to evaluate the images based on the shape of the visible profiles rather than their size, density, or distribution. The test interface measured the accuracy and time required to make each selection.

To establish a baseline for the average selection time, 20% of the test screens contained three images that were all taken from the same volume. This is used to measure user response to an *ideal* particle—i.e., a theoretical decoy particle that has an identical appearance to the original shape.

The images used for the study depict black particles on a white background and are antialiased with a size of 250x250 pixels. Images were separated by 40 pixels of a neutral grey color. Users were seated comfortably in a dimly lit room at a distance of approximately 55cm from the computer monitor. Each image was shown at a size of 7.8cm, forming a

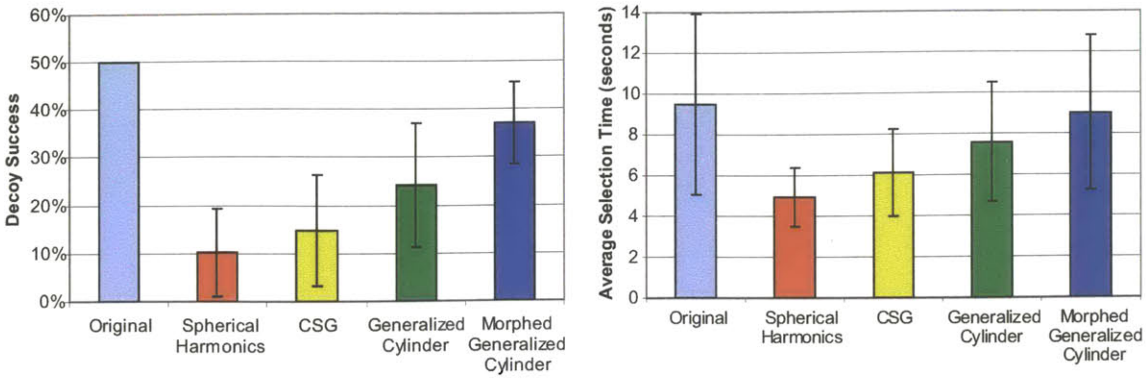


Figure 3-16: User study results showing the relative performance of four different algorithms for extrapolating particle shape based on limited shape information. The graph on the left shows the frequency with which users were "fooled" by a decoy model. Average selection times are shown on the right. Mean and standard deviation values are shown for both graphs.

visual angle 8.1 degrees.

In accordance with the guidelines established by the MIT Committee On the Use of Humans as Experimental Subjects (COUHES), each participant in the study was given a detailed written description of the study procedures, as well as a summary of their rights as a research subject. This consent form is included in Appendix A.

All study participants are members of the computer graphics and computer vision communities at MIT, none of whom had any considerable knowledge of the specific algorithms being tested. Participation was completely voluntary, and users were not compensated in any form.

Prior to the start of the test, users were given a brief training session, which included an example test with ten evaluations to familiarize them with the task and interface. The complete test contained 80 questions—four for each particle/algorithm combination, plus four questions for each particle shape in which no decoy was shown. The questions were presented in a different random order for each user, and images were randomly selected from a database for each screen. Between evaluations, users were shown a neutral grey screen for one second.

Results

Sixteen users participated in the study, yielding 1280 individual data points. The results of the study are summarized in Figure 3-16. In the first chart, we see the success rate at which the decoy particles were selected by the users. In the case of an ideal decoy particle, we would expect a success rate of 50%; i.e., we expect that a user would be equally likely to select the image from the original volume or the decoy volume. This baseline value is indicated by the light blue bar at the far left.

These values indicate a decisive ordering of algorithm effectiveness, ranking the morphed generalized cylinder method with the highest performance, followed by generalized cylinders, constructive solid geometry, and spherical harmonics. One-way analysis of variance (ANOVA) demonstrates that decoy success has a significant dependence on the algorithm used ($p < 0.001$).

These same results are reflected in the second table in Figure 3-16, which shows the average time the users spent making their selections. The baseline time is shown in light blue on the far left; this represents the case where all three images were derived from the same volume, and can be considered to be indicative of selection time for an ideal decoy particle. Again, ANOVA indicates a significant dependence on the algorithm used to create the decoy particles ($p < 0.001$).

Selection time should not, in and of itself, be considered a measure of algorithm performance. It does, however, provide some indication of how much user effort is involved in assessing shape similarity. In this study, I have chosen to use the metric to validate the results obtained in the ranking of decoy success. The relation between the measures of decoy success and selection time can be clearly seen from a comparison of the two charts in Figure 3-16.

As shown in Figure 3-17, the behavior of each algorithm can be considered more precisely by breaking apart the data and looking at how the performance of each method varies with different input parameters. Two-way analysis of variance applied to the decoy success rate indicates that performance has a significant dependence on the algorithm used ($p < 0.001$) as well as the shape that is being approximated ($p = 0.003$). Furthermore,

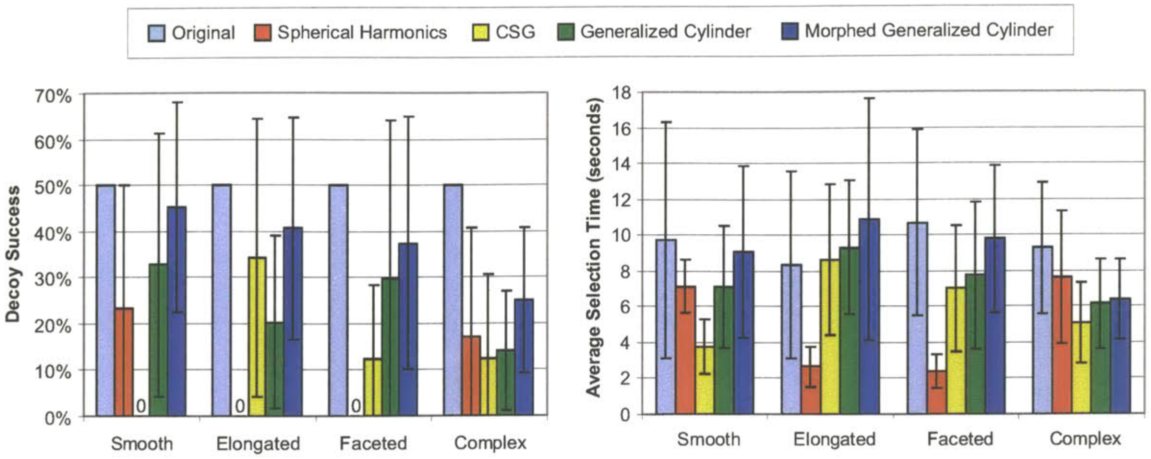


Figure 3-17: Detailed graph of user study results showing variation between input datasets. The graph on the left shows the frequency with which users were "fooled" by a decoy model. Average selection times are shown on the right. Mean and standard deviation values are shown for both graphs.

ANOVA indicates a significant interaction between the algorithm and shape ($p < 0.001$).

For each of the datasets, the morphed generalized cylinder algorithm generates more successful decoys than the other three alternatives. However, the relative performance of the other methods varies widely.

The spherical harmonics algorithm is acceptable for replicating the appearance of smooth surfaces that lack sharp features, but is poorly suited for representing elongated shapes or particles with sharp corners. In contrast, CSG provides a reasonable approximation for elongated particles or surfaces with sharp edges, but is poorly suited for representing rounded shapes.

As for the generalized cylinder method, I found that the algorithm tends to create profiles with clear symmetry across the axis of rotation. As a result, we see that the algorithm performs well for replicating particles that exhibit symmetry in the input dataset, but the method performs poorly on asymmetric shapes.

Once again, selection time should not be considered as a direct measure of algorithm performance, but serves to validate the ranking of decoy success. For the most part, the relative values in the left and right tables of Figure 3-17 are well-correlated. Two-way analysis of variance on the selection times indicates a significant dependence on the algorithm ($p < 0.001$). However, selection time does not exhibit a significant dependence on the particle shape ($p = 0.36$), nor does ANOVA indicate a significant interaction between the algorithm and shape ($p = 0.16$).

When looking at the standard deviation values in Figure 3-17, it should be remembered each user observed only four instances of each combination of shape and algorithm. For a rating of decoy success, this results in only two bits of information, yielding values of 0%, 25%, 50%, or 100%. High levels of variability in these values should be expected. Regardless, for both graphs, ANOVA demonstrates a significant dependence on the algorithm.

3.6.3 Performance Summary

Of the four algorithms considered here for replicating aggregate materials, this research points to the morphed generalized cylinder method as the most effective for approximating 3D particle shapes from 2D profiles. The three other methods—generalized cylinders, constructive solid geometry, and spherical harmonics—are each effective within certain domains, but do not generalize well to a wide variety of input shapes.

Automated metrics for predicting psychophysical response can be enormously useful for evaluating research results, as it is not always desirable or plausible to perform a complete user study. The automated techniques considered here are marginally successful for predicting algorithm performance from a psychophysical standpoint.

Only one of the metrics yields the same ordering as the user study, as seen from a comparison of Figures 3-14 and 3-16. The most accurate predictor is the measure of area/perimeter^2 applied to profiles through the particle's center of mass. Of the remaining metrics, each ranks the generalized cylinder and morphed generalized cylinder methods above the spherical harmonics and CSG methods, but there is no consensus on the total ordering.

One shortcoming of the automated metrics presented here is that they are unable to capture some of the appearance characteristics that were discerned by the user study. To improve automated prediction of perceptual shape similarity, additional metrics should be considered that account for symmetry, anisotropy and the presence of sharp corners.

Future researcher may consider the use of the metrics in this thesis as coarse predictors

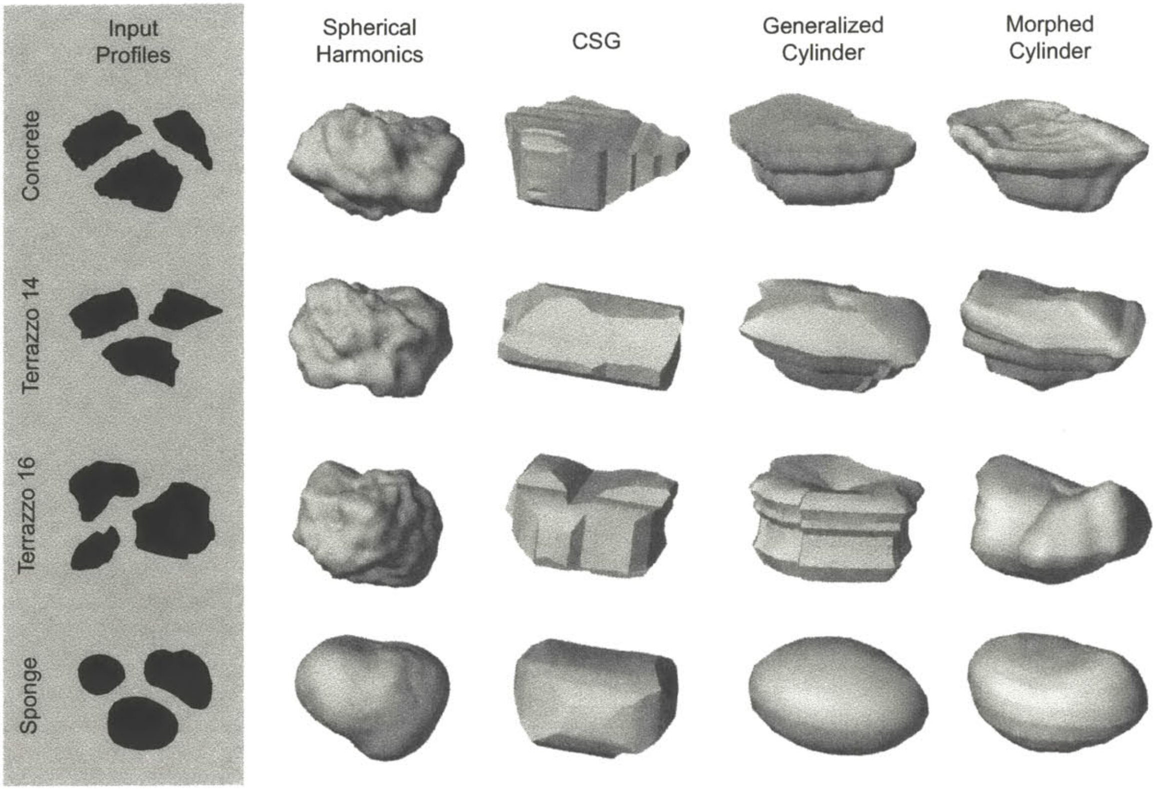


Figure 3-18: In the left column are characteristic profiles for each of four different real-world datasets. To the right are the 3D particle shapes created by each of the four particle approximation methods.

for shape similarity, but these automated techniques should not be treated as equivalent replacements for direct user evaluation.

3.7 Working with Real Data

When working with real data, there are a few caveats that must be considered. For the synthetic volumes that were described in the preceding sections, all particles have the same shape and size. For real volumes, it's not clear how many distinct particle shapes may be necessary to achieve the diversity of appearances that are present in the input sample.

While the temptation may be to use a large number of synthetic particles, it should be recognized that even simple particle shapes can exhibit a wide variety of profile appearances. For instance, a cube can exhibit profiles with anywhere from three to six faces. A cylindrical particle can exhibit profiles that appear as circles, ellipses, semicircles, rect-

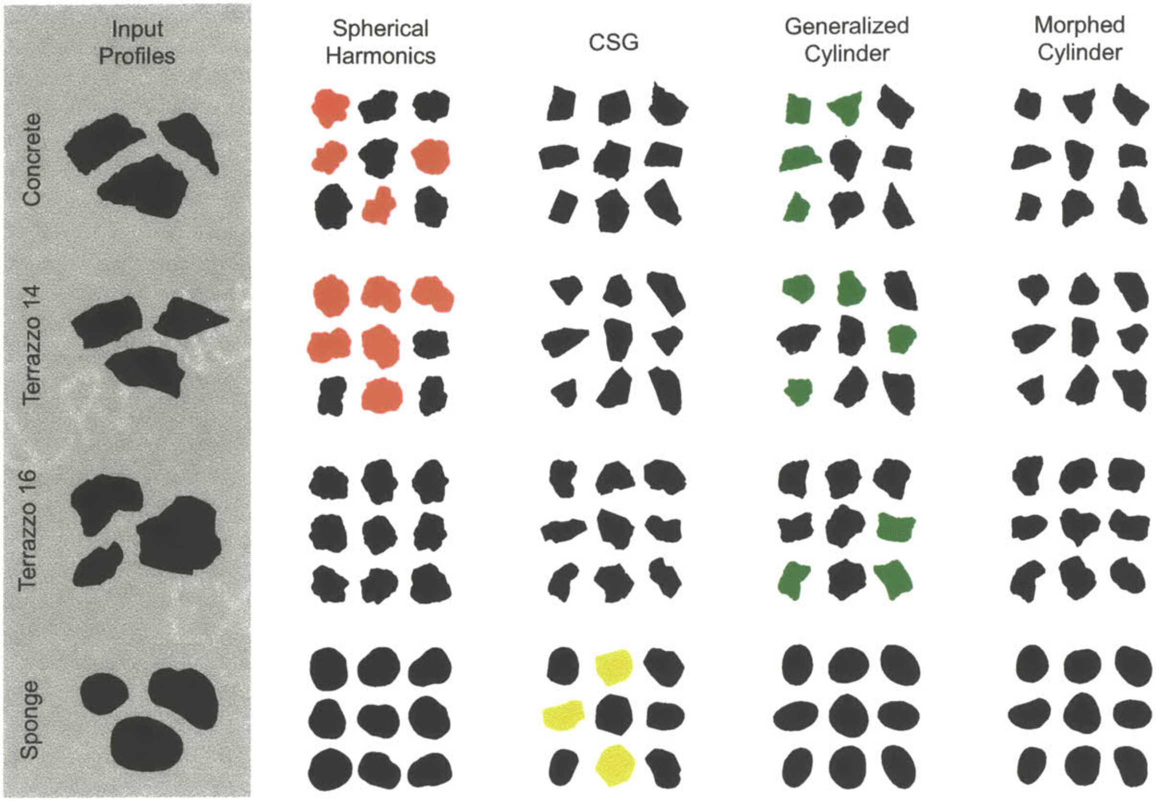


Figure 3-19: In the left column are characteristic profiles for each of four different real-world datasets. To the right are randomly selected central profiles for each synthesized particle shape. Profiles that are poorly matched to the original data are highlighted in color. Red profiles are overly smooth or curvy, yellow profiles exhibit excessively sharp corners, and green profiles show symmetry that is not present in the original data.

angles, parallelograms, closed parabolas, or a variety of other shapes. More complicated particles can exhibit an even greater array of profiles.

To test the four proposed methods on real world-data, I applied the algorithms to the input profiles shown in the left column of Figure 3-18, which were taken from segmented photographs of texture images. The corresponding synthetic particle approximations are shown on the right.

Central slices through these synthetic particles are shown in Figure 3-19. The appearance artifacts are comparable to those seen in Figure 3-12. As before, certain types of perceptual artifacts are associated with each particle estimation algorithm. Particles generated with the spherical harmonics algorithm tend to be overly smooth or blobby, CSG particles exhibit sharp corners, and generalized cylinder particles yield profiles with distinct symmetry.

Within the framework of the complete texture synthesis system, the performance of the shape approximation algorithms can be evaluated by looking at annealed texture volumes that include particles created with each method. In the left-hand column of Figure 3-20 are segmented regions from four input textures. The images on the right show slices through annealed 3D volumes that were created using the distribution recovery and annealing techniques described in the following chapters. Each synthetic image exhibits only one 3D particle shape varied under rotation, translation, and isotropic scaling.

In Figures 3-12, 3-19, and 3-20, I manually selected synthetic particle shapes that are poorly matched to those in the input texture. These shapes are highlighted in color to illustrate how they differ perceptually from the input. Red profiles are overly smooth or curvy, yellow profiles exhibit excessively sharp corners, and green profiles show symmetry that is not present in the input data.

Figures 3-19, and 3-20 corroborate our earlier findings that the morphed generalized cylinder is most effective for synthesizing particles over a wide variety of input appearances. Furthermore, they demonstrate that even a single synthetic particle shape can effectively convey the variety of 2D profile shapes seen a real-world texture.

3.8 Shape vs. Texture

Even though Figures 3-19 and 3-20 depict essentially the same data, there is a difference in how the images are perceived. The tabular format shown in Figure 3-19 emphasizes the structured attributes of the profile slices, which are seen as distinct shapes. In contrast, Figure 3-20 emphasizes the stochastic attributes of the profile slices, causing the images to be perceived as texture.

This difference in perception suggests the need for similarity metrics that operate both in the structured domain of profile shape and the stochastic domain of annealed textures. Future research may attempt to discern specific differences in how humans perceive images along this scale of appearances.

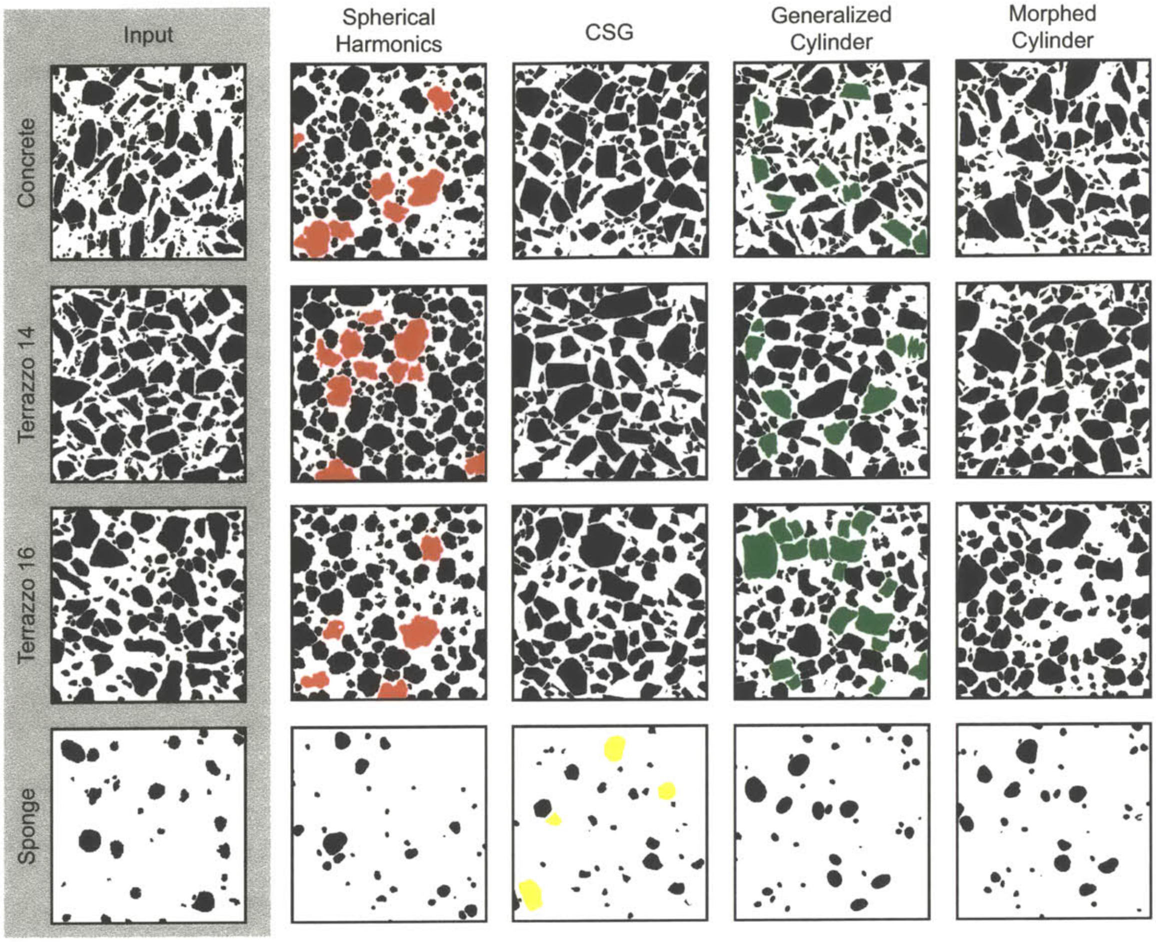


Figure 3-20: In the left column are segmented textures obtained from photographs of composite materials. The four columns on the right show synthetic textures, each of which exhibits a single 3D particle shape created by the indicated method. Profile shapes that do not appear similar to the input texture are highlighted in color.

3.9 Summary

This chapter has introduced four different methods for approximating particle shapes from characteristic 2D profiles. The constructive solid geometry and morphed generalized cylinder methods are novel contributions in this area of research. A rigorous user study indicated that the morphed generalized cylinder method outperforms the three other algorithms under a variety of input conditions. The other three methods—spherical harmonics, CSG, and generalized cylinders—each perform well under some circumstances, but are limited in the types of profile appearances they can effectively replicate.

Due to the unconstrained nature of the particle extrapolation problem, we rely heavily on assumptions about the input data in order to derive reasonable solutions. In the case of the spherical deformation method, one underlying assumption is that the frequencies seen in a profile are unrelated and can be modeled using a stationary Gaussian process. This results in the elimination of sharp corners that may be seen in the input data. Future research may consider the possibility of modifying the algorithm in a manner that respects the dependencies observed between the frequency bands in the input.

Although the spherical harmonics algorithm had the worst performance in terms of exhibiting similar 2D profiles, the particles that are created by the method have a satisfying appearance in 3D. This suggests that if the internal structure of a material is to be viewed directly, then the spherical deformation method may still offer a reasonable solution.

The four methods described here are presented in the context of synthesizing particles that will be viewed through 2D slices. Looking beyond this domain, the morphed generalized cylinder method may offer a solution for general shape modeling with simple, limited user input. Generalized cylinders have long been valued for their ability to represent a wide range of appearances with compact, intuitive parameters. Morphed generalized cylinders offer a wider array of 3D appearances, but can still be created with simple parameters. Very little computational effort is required to create a particle with this approach, which suggests the possibility of a user interface where an artist can edit three control curves simultaneously and immediately see the changes to a 3D model.

In addition to the user study, two automated metrics—total curvature and circularity—were considered for approximating perceptual shape similarity. These metrics were only marginally successful at predicting psychophysical response, and other metrics should be considered in future research. A number of other 2D shape metrics have been introduced within the computer vision community, but the vast majority are designed to identify object features—often points with high curvature—for object recognition. Unfortunately, these types of established metrics are of limited use when attempting to quantify the subjective similarity of abstract shapes.

Particle shapes recovered with the algorithms described here can be used as inputs to the next stage in the solid texture synthesis pipeline. The following chapter demonstrates a method for approximating particle size distributions based only on the particle shape and the observed profile distribution in a 2D slice through a real-world material.

Chapter 4

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 chapter, I present a disciplined approach to recovering 3D volume parameters using methods motivated by spatial sampling theory. I begin by demonstrating the approach with a distribution of spheres, and then extend the approach to work with arbitrary particle types.

4.1 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*. The approach demonstrated here 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],

$$N_A = \bar{H} N_V, \tag{4.1}$$

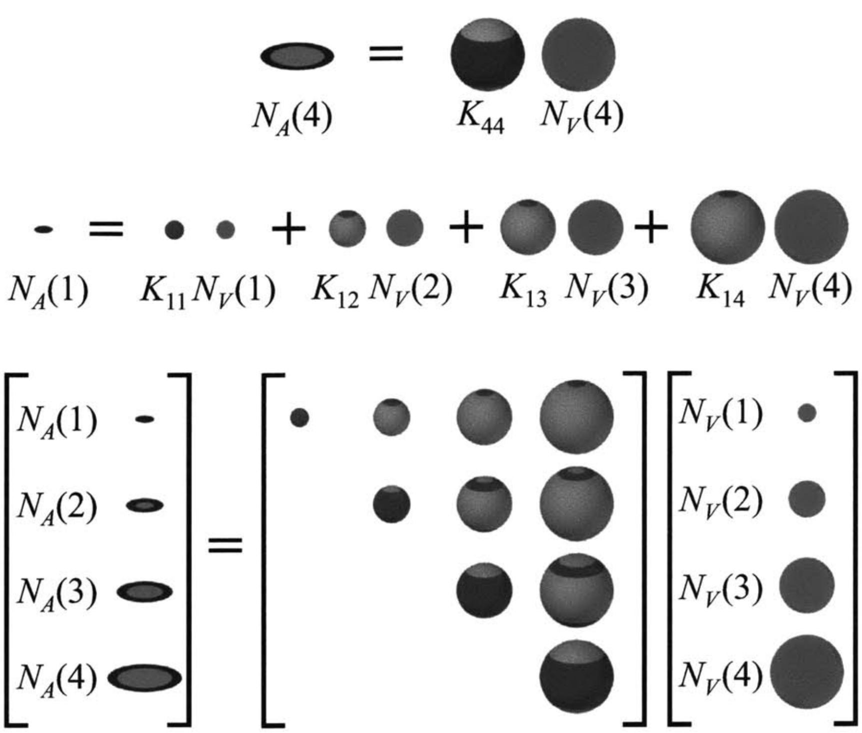


Figure 4-1: In this set of three equations, the blue disks represent profile densities, $N_A(i)$, the green spheres represent particle densities, $N_V(i)$, and the red and white spheres represent the probabilities that a sphere of a given size appears with a particular profile size. The first two expressions 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.

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.

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

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 $(i-1)/n < d \leq i/n$. 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 4-1 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 4-1. The complete density vectors N_V and N_A are related by the expression

$$N_A = d_{max} K N_V.$$

The corresponding visual representation is shown at the bottom of Figure 4-1. Spheres can only exhibit profiles of equal or smaller diameter, so K is an upper-triangular matrix where

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

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

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

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. In practice, the explicit matrix inverse never needs to be taken, as it is simple to solve for N_V using back-substitution.

4.2 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. I 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 4.1.

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

$$N_A = \bar{H} K N_V \tag{4.2}$$

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,

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

Note that if only one histogram bin is used ($n = 1$), then Equation 4.2 reduces to the fundamental relationship of stereology (Equation 4.1).

For an arbitrary particle P, represented as a watertight polygon mesh, it may be difficult to compute the K matrix analytically. To compute these values, I use a Monte Carlo approach that averages statistics observed by repeatedly slicing through the particle, as shown in Figure 4-2. At each step, the polygon mesh is assigned a random orientation and a cutting plane at a random depth. To compute the area of each observed profile, I implemented two different methods with trade-offs between speed and simplicity.

The first method can be easily implemented in a robust manner and takes advantage of the speed of modern graphics hardware and the *stencil buffer*, which can manage supplemental information about a rendered image [Woo et al. 1999]. At each iteration of the Monte Carlo process, the polygon mesh for particle P is rendered such that the near clip-

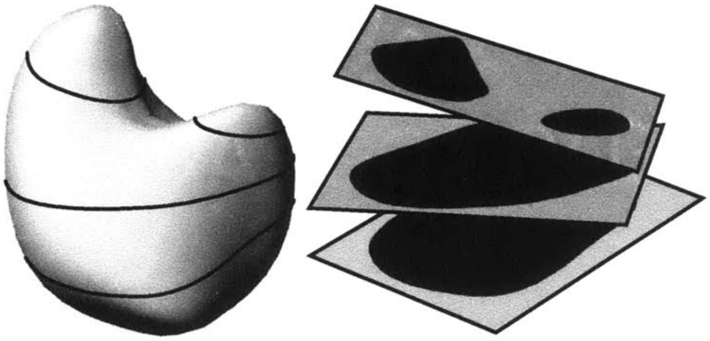


Figure 4-2: The K matrix for an arbitrary particle is constructed by calculating the cross-sectional area of random slices through the volume. If a single slice exhibits more than one profile, then each distinct region should contribute independently to the histogram.

ping plane of an orthographic camera is coincident with the random cutting plane. 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 second method is based on the *winged edge mesh* data structure [Baumgart 1972]. This approach is more difficult to implement robustly, but can calculate profile statistics more quickly and accurately. The polygon mesh for particle P is annotated such that each edge stores a pointer to the two adjacent triangles and each triangle stores pointers to the three adjacent edges. The algorithm begins by looking for an edge whose endpoints lie on opposite sides of the cutting plane. Once this is found, the polygon mesh can be rapidly traversed by taking advantage of the local connectivity information.

The calculations derived from either method can be used to populate the histogram for each column of the K matrix. The Monte Carlo 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, $\bar{H}_P$.

In some instances, a slice through a non-convex particle may result in two or more disjoint profiles, as shown in the top slice in Figure 4-2. If this were to occur in a real-world material sample, there would be no reliable way to discern if the adjacent profiles were the product of one or two distinct particles, and each would be considered separately

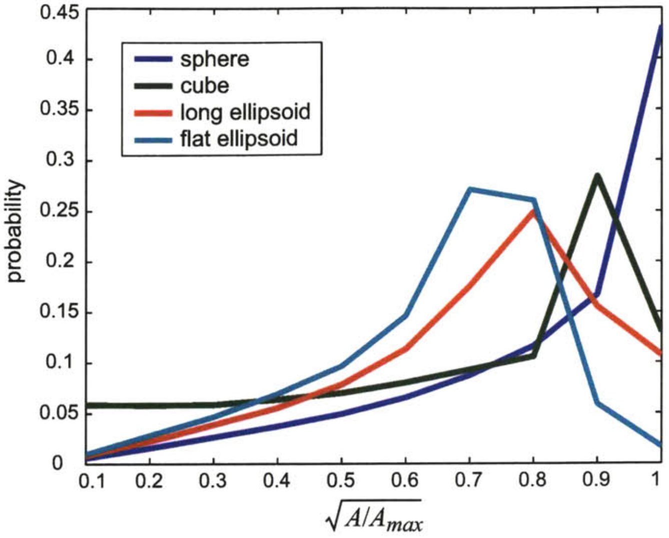


Figure 4-3: Likelihood of cross-sectional area for simple particle types.

when computing the profile density vector, N_A. Thus, when populating the K-matrix, each disjoint profile region in a particle slice should be considered separately, and each should contribute to the histogram construction.

For each particle type I tested, this process converged to a residual of $< 0.5\%$ for each histogram bin within 100,000 iterations. For the hardware rendering method, computation time was less than two minutes for input particles with 10,000 polygons. When using the winged edge mesh, this computation time can be reduced by more than 80%. Some example statistics for simple particles are shown in Figure 4-3.

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 the input image. Suppose the image exhibits profiles with maximum area A_{img}. This is equal to the maximum profile area of P if scaled uniformly by

$$s = \sqrt{A_{img}/A_{Pmax}}. \tag{4.3}$$

This scale factor is used to calculate the mean caliper diameter $\bar{H} = s\bar{H}_P$, which is used in Equation 4.2.

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

$$N_V = \frac{1}{\bar{H}} K^{-1} N_A \tag{4.4}$$

4.3 Managing Multiple Particle Types

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

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

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

$$\begin{aligned} N_A &= \sum_i \left(\bar{H}_i K_i P(i) N_V\right) \\ &= \sum_i \left(\bar{H}_i K_i P(i)\right) N_V \end{aligned}$$

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

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

When working with physical input data, these expressions may occasionally yield noisy histograms or negative values for some elements of N_V. A thorough analysis of these problems and other considerations when dealing with physical data are discussed in Chapter 5.

4.4 Summary

This chapter has described how a disciplined stereological algorithm can be used for recovering particle densities in a 3D volume based only on 2D profile observations and one or more estimated particle shapes. This is done by establishing a relationship between the size distribution of observed 2D profiles and related statistics of the embedded particles. This process was first derived for distributions of spheres and then extended to apply to particles of arbitrary shapes.

Statistics of an embedded particle shape are computed with a Monte Carlo method that takes repeated slices through a geometric model at random orientations and offsets. Considerations for disjoint profiles were discussed, and the resulting statistics for several example particles were shown.

One of the assumptions of the algorithm described here is that particles exhibited in the volume have uniformly distributed orientations. However, it should be noted that the method can be easily adapted for any known distribution of particle orientations. For instance, if a volume is populated with cylinders that are oriented perpendicular to the viewing plane, then the K-matrix can be computed accordingly. Cylinders under these constraints will always exhibit a constant profile size, so K will be zero everywhere except along its diagonal. By extrapolating from these observations, it may be possible to apply the density recovery algorithm to anisotropic materials.

In the following chapter, I consider sources of error in the texture synthesis pipeline, focusing on the interdependence between particle shape estimation and the distribution recovery process.

Chapter 5

Anticipating Error

The problem of synthesizing solid textures from images of aggregate materials is highly unconstrained. In order to obtain a solution, I rely on a number of assumptions, each of which may introduce error into the computation. In this chapter, I consider potential sources of error and concerns in obtaining physically plausible results.

As described in Chapter 3, the algorithms that we consider for extrapolating 3D particle shapes from 2D profiles are psychophysically motivated. The objective is to obtain particles whose profiles appear similar to those seen in an input image. However, there is no guaranteed that the resulting 3D models are physically correct. Poorly estimated particle shapes should be expected to introduce a certain amount of error into the distribution recovery algorithm. Additional errors may be introduced by quantization from histogram binning, as well as a finite number of profile observations in an input image.

5.1 Synthetic Volumes

To test the robustness of the distribution recovery algorithm under a variety of input conditions, I analyzed a number of different synthetic distributions. In each case, a synthetic volume was populated with spherical particles. Ten equally spaced slices were taken through the volume, and the observed profiles were binned according to their diameter to populate the profile density histogram, N_A. I then solved for the particle densities, N_V, using Equation 4.1.

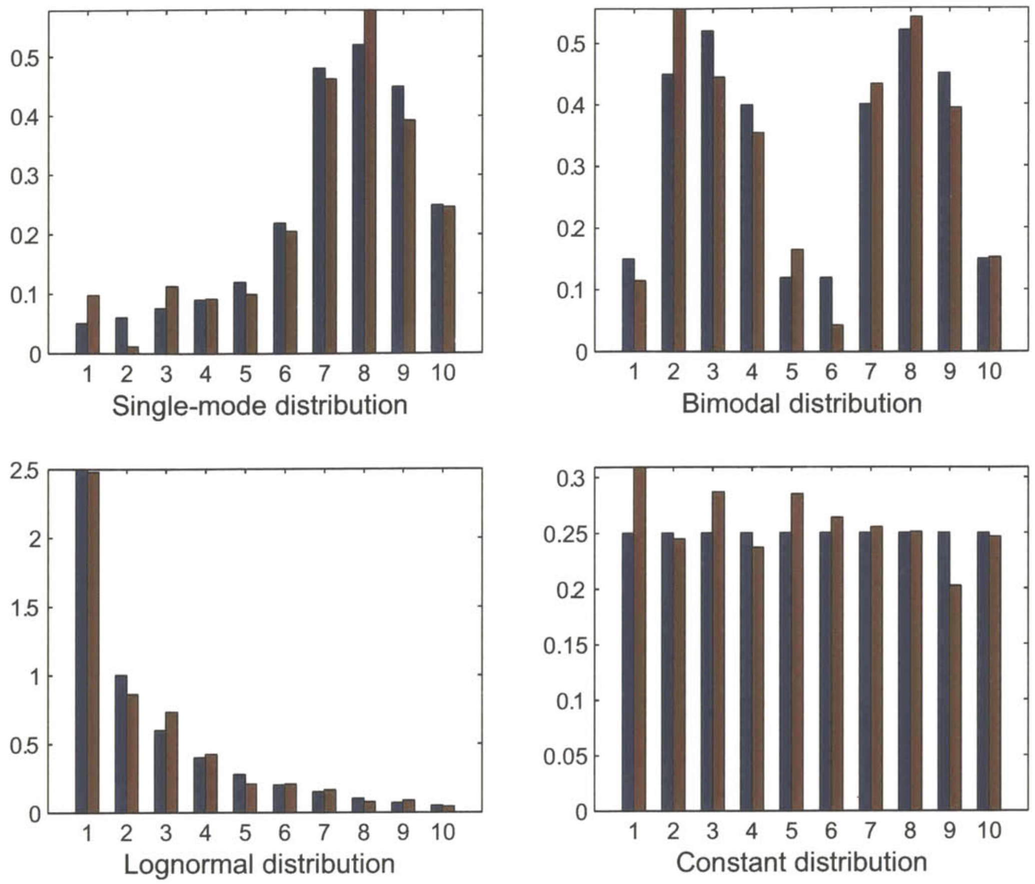


Figure 5-1: Performance of the density recovery algorithm on a variety of input distributions. Each graph shows 3D particle densities grouped into ten histogram bins. Actual distributions are shown in blue and estimated distributions in red.

Figure 5-1 shows the results of the algorithm applied to single-mode, bimodal, lognormal, and constant distributions. The actual particle distributions are shown in blue with the estimated distributions in red. These results were based on between 1050 and 1400 profile observations, grouped into ten evenly sized histogram bins. Figure 5-2 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.

All of the distributions are estimated with sufficient precision to obtain materials with a comparable appearance to the input. It should be noted that the distribution recovery algorithm obtains the most precise results when applied to a material with a lognormal particle distribution. This is noteworthy because lognormal distributions are commonly observed in both man-made and natural materials, as will be discussed in greater detail in

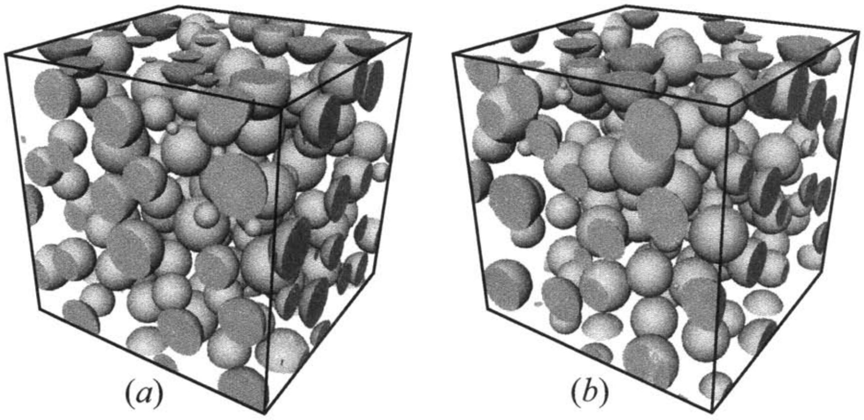


Figure 5-2: Comparison of the single-mode volume of spherical particles (*a*) and a comparable volume obtained via the density recovery algorithm (*b*).

Section 5.4.2.

5.2 Working with Physical Data

For the synthetic volumes described above, we benefit from being able to obtain an exact profile count, from having a large number of profile observations, and from knowing the exact particle shapes *a priori*—all of which make the calculations more statistically robust. In contrast, when working with physical data, we are often unable to count small profiles, we may be limited to fewer profile observations, and we must rely on particle shape approximations. Each of these introduces potential sources of error into our calculations.

Errors in the volume density recovery process are typically manifested as either dramatically different densities in adjacent histogram bins or negative estimates for particle densities N_V. If only a few profiles have been observed in one or more of the profile histogram bins, N_A, 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, which allows for greater statistical precision within each bin.

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. These underrepresented profiles may, in turn, result in negative estimates for small particle densities. Prior publications have addressed the phenomenon of underrepresented particle and profile observations in physical data, sometimes referred to as the problem of *missing fines* [Keiding and Jensen 1972; Maerz 1996]. It should not necessarily be considered an undisciplined approach to clamp negative estimates for particle densities to zero based on the assumption that the errors were introduced by underrepresented profile observations.

5.3 Accounting for Image Boundaries

When observing images of physical data, special care should be taken when counting profiles near the image boundary. To obtain statistically accurate results, a border should be established along the image boundaries that is equal to at least half the width of the bounding box of the largest visible profile. If the center of a profile lies outside of the boundary, then that profile should not contribute to the density histogram, N_A.

5.4 Interdependence of Particle Shape and Distribution

When dealing with physical volumes, the distribution recovery algorithm depends in part on the 3D geometry recovered by the shape estimation process. Considering this dependency, it would be useful to know the conditions under which an approximated particle shape can be used as a substitute for an actual particle shape.

In the following analysis, I first consider the restrictive condition in which a volume populated with particles of type P_{in} can always be approximated using an estimated particle shape of type P_{est} in such a manner that both volumes exhibit identical profile distributions, N_A. I then discuss how this is applicable for observed real-world distributions.

5.4.1 Guaranteed Particle Equivalence

As was shown in equation 4.2, if a particle P_{in} with a known shape is embedded in a volume with known distribution N_{Vin}, then it will exhibit a profile distribution N_A such that

$$N_A = \bar{H}_{in} K_{in} N_{Vin} \tag{5.1}$$

where $\bar{H}_{in}$ is the mean caliper diameter and K_{in} is the representative matrix of P_{in}.

If we instead assume that the particles have an estimated shape P_{est}, then we know from Equation 4.4 that we can solve for the particle distribution $N_{V_{est}}$ based on the observed profile distribution N_A.

$$N_{Vest} = \frac{1}{\bar{H}_{est}} K_{est}^{-1} N_A \tag{5.2}$$

Our objective is obtain a particle distribution $N_{V_{est}}$ that matches the observed profile distribution given in Equation 5.1. This can be done by substituting Equation 5.1 into Equation 5.2 as follows:

$$N_{Vest} = \frac{1}{\bar{H}_{est}} K_{est}^{-1} \bar{H}_{in} K_{in} N_{Vin}$$

This expression yields a physically plausible solution when all the entries in $N_{V_{est}}$ are nonnegative. Under further consideration, we see that a solution can be guaranteed only when the matrix product $K_{est}^{-1} K_{in}$ has all nonnegative entries.

To test guaranteed equivalence between simple particle shapes, I evaluated this matrix product for each combination of sphere, cube, elongated ellipsoid, and flattened ellipsoid with matrices of size 5 × 5. In all instances where we assume that the volume is populated with spherical particles—i.e., P_{est} is a sphere—this matrix product contains all nonnegative entries. In other words, for all of the considered particles, regardless of the actual distribution $N_{V_{in}}$, we are able to assume the that the shape is spherical and still obtain a physically plausible solution for $N_{V_{est}}$. However, of the four particle shapes that I considered, no two other particles exhibit this type of equivalency. Furthermore, when each of the approximated shapes shown in Figure 3-11 were considered as replacements for the original shape, none demonstrated guaranteed equivalency. In short, we find that guaranteed shape equivalency under arbitrary particle distributions is rare.

To understand why a sphere is a good substitute for other particle shapes, it helps to consider the physical interpretation of what takes place when solving Equation 4.2 for N_V

using back-substitution. The expanded matrix form of this equation appears as follows.

$$\begin{bmatrix} N_{A1} \\ N_{A2} \\ \vdots \\ N_{An-1} \\ N_{An} \end{bmatrix} = \bar{H} \begin{bmatrix} K_{11} & K_{12} & \cdots & K_{1,n-1} & K_{1,n} \\ 0 & K_{22} & \cdots & K_{2,n-1} & K_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & K_{n-1,n-1} & K_{n-1,n} \\ 0 & 0 & \cdots & 0 & K_{n,n} \end{bmatrix} \begin{bmatrix} N_{V1} \\ N_{V2} \\ \vdots \\ N_{Vn-1} \\ N_{Vn} \end{bmatrix}$$

In the first step, we solve for the density of the largest particles, giving us an expression that guarantees a positive result.

$$N_{Vn} = \frac{1}{\bar{H}K_{nn}} N_{An}$$

Having recovered a density estimate, N_{Vn}, we can anticipate that these particles will also contribute profiles of smaller sizes. Specifically, for profiles of size N_{An-1}, we expect the particles of the largest size to contribute a density of $\bar{H}K_{n-1,n}N_{Vn}$. This value is included in the second step in the back-substitution process:

$$N_{Vn-1} = \frac{1}{\bar{H}K_{n-1,n-1}} (N_{An-1} - \bar{H}K_{n-1,n}N_{Vn})$$

As long as the number of observed profiles N_{An-1} is greater than or equal to the number of anticipated profiles $\bar{H}K_{n-1,n}N_{Vn}$, then the expression evaluates to a positive result and the solution continues to be physically plausible. However, if the difference of the values is negative, then the expression results in a negative density. The same logic can be applied to the remaining steps necessary to obtain a complete solution for the particle densities N_V.

This analysis suggests that particles that exhibit primarily large profiles can reduce the interdependence between steps in the back-substitution process. Thus, this type of particle tends to be a more likely candidate for guaranteed equivalence. This explains why when a volume is populated with cubes or ellipsoids, we are guaranteed to be able to match the 2D profile statistics using spherical particles. As can be seen from Figure 4-3, the majority of the profiles exhibited by a sphere are clustered into the histogram bin of the largest size.

5.4.2 Physical Plausibility in Real-World Distributions

Thankfully, guaranteed particle equivalence is not necessary to solve for physically plausible distributions for most real-world examples. To demonstrate this, we again consider the physical interpretation of obtaining a solution for the particle densities using back-substitution. As described in the previous section, when performing the second step in the process, we hope to obtain a positive value for the following expression:

$$N_{An-1} - \bar{H}K_{n-1,n}N_{Vn}$$

When we considered the problem from the perspective of the particle shape, we sought a small value for $K_{n-1,n}$. Alternatively, we can obtain a positive result with a sufficiently large profile density N_{An-1}. For each additional step in the back-substitution process, physical plausibility is improved by having increasingly large profile densities.

The good news is that this relationship is exhibited by lognormal particle distributions, which are by far the most common in both natural and man-made materials [DeHoff 1965].

To test robustness on real-world data, I considered four input profile distributions based on photographs of real-world materials. For each example, I approximated a particle shape using each of the methods described in Chapter 3 and then estimated the particle densities using the algorithm outlined in Chapter 4. The resulting particle distributions N_V are shown in Figure 5-3

Only two of the sixteen histograms resulted in negative values—concrete with the particle shape approximated with the generalized cylinder method and terrazzo 16 with the shape derived from the CSG algorithm. In neither case do the negative values exceed -0.2. As discussed previously, these negative densities may be attributed to the problem of missing fines, and should not be particularly concerning.

Ultimately, the calculated particle distribution is of little concern as long as it is physically achievable. As with the shape approximation algorithms, we can consider the distribution recovery process to be psychophysically motivated. Rather than being interested in the 3D particle distribution, we are primarily concerned with the appearance of the distribution as seen in 2D slices through the solid texture.

5.5 Summary

In this chapter, I have discussed the interdependence of particle shape and distribution and addressed sources of error in the distribution recovery algorithm. Potential sources of error include poor shape estimation, histogram quantization, and a limited number of profile observations.

In general, it is unlikely that an approximated particle shape can be guaranteed as a suitable proxy for another particle with an arbitrary distribution. However, for the lognormal distributions that are seen in most real-world materials, this problem of guaranteed equivalence is typically not a concern.

From a practical standpoint, we can consider the algorithms for recovering shape and particle density to operate independently. Even if the geometric model obtained by the shape extrapolation process is not physically accurate, the distribution recovery process can insure that the observed 2D profile statistics match those observed in an example image. We benefit from having option of having multiple shape approximation algorithms at our disposal. If the density recovery algorithm yields negative estimates for particle densities, then it is possible that the problem may by alleviated simply by considering a different particle shape. ‘ Between the particle shape and particle density, we now have enough information to construct a synthetic volume with the same parameters as an input example. This process, including techniques for incorporating color and residual noise, will be described in the following chapter.

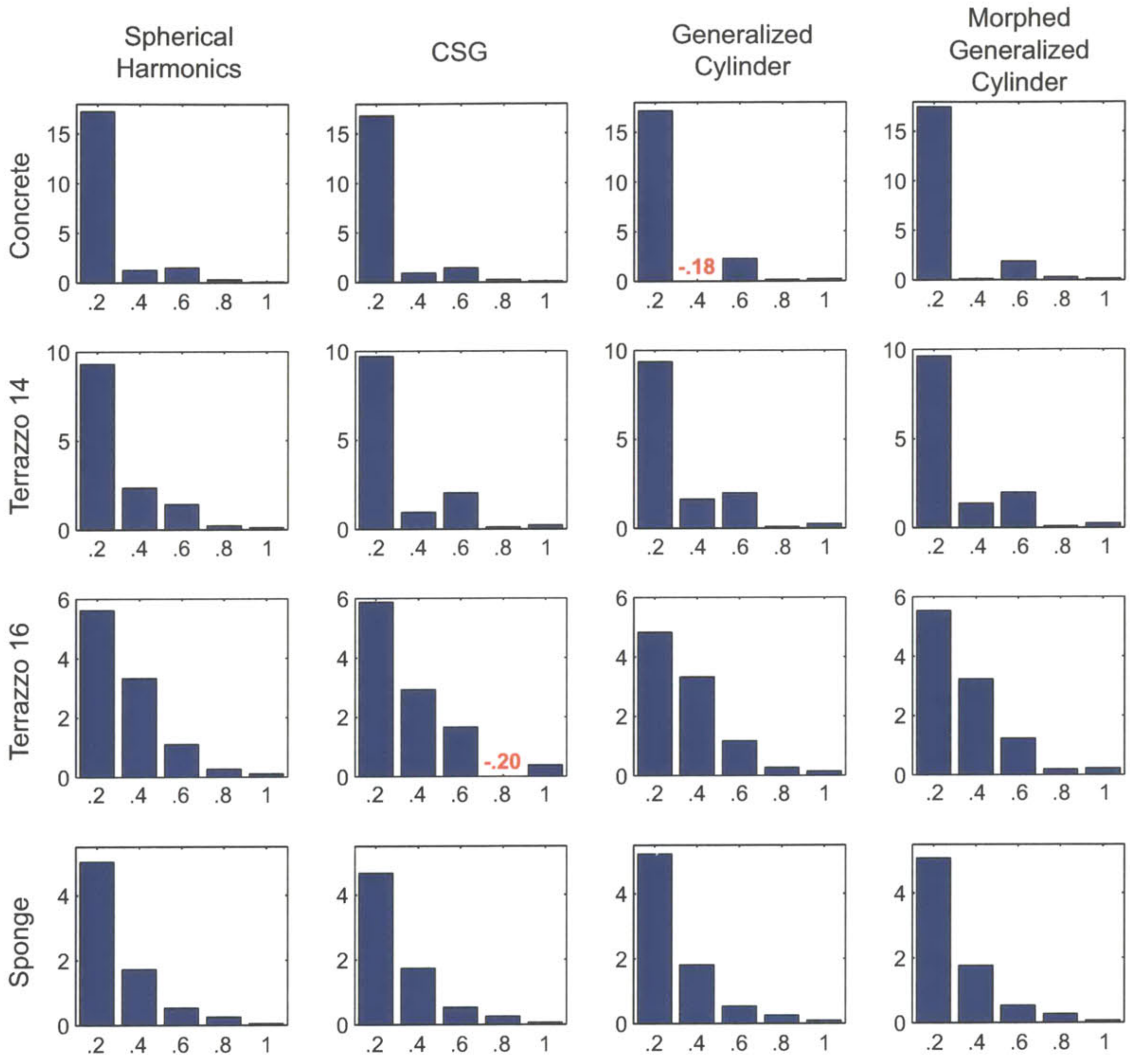


Figure 5-3: Estimated particle densities N_V for several real-world materials. Particle shapes were estimated using each of the four proposed algorithms. Negative density estimates are shown in red.

Chapter 6

Reconstructing the Volume

The preceding chapters introduced techniques for estimating particle shape and distribution based on observations of a 2D image. Once these parameters are known, a synthetic volume can be constructed to match the appearance of the input. The reconstruction process establishes particle positions and colors, as well as a residual noise function to add fine details characteristic of the input.

6.1 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 4.3. The naive approach for populating a volume is to add one particle at a time, randomly testing orientations and translations until sufficient vacant space is found. This approach is historically referred to in the texture community as *bombing* [Peachey 1985]. Unfortunately, texture bombing methods fill space inefficiently and work only for loosely packed volumes.

For all of the example volumes shown in this thesis, I populate the volume with all of the particles, ignoring overlap, and then perform simulated annealing to resolve collisions. This method repeatedly searches for all interpenetrating particles and then relaxes particle positions to reduce the number of intersections.

Both the bombing algorithm and the annealing process consider 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 solid texture.

Simulated annealing is a computationally expensive process, as it requires repeated searching for all interparticle collisions. The relaxation process described here is the most time-consuming part of the texture synthesis pipeline. In order to improve the efficiency, I apply a number of optimizations:

- Only vertex-particle collisions are considered. Although this approach may fail to capture certain types of collisions, it performs well if the vertex density is sufficiently high and the geometry exhibits few sharply pointed features. Since simulated annealing is an iterative process, it is acceptable to ignore some collisions during a finite period of time, as long as the system converges to a solution.

- All particles are hashed into a 3D grid to reduce the number of collisions that must be checked at each time step. This allows queries to be limited to the immediate vicinity of any given particle.

- Point-in-particle queries are performed by transforming the point into the local coordinate system of the particle and casting a ray in the positive x-direction. The point falls inside the particle if and only if the ray encounters an odd number of polygon intersections. This procedure is accelerated by projecting the particle mesh onto the plane $x = 0$ and hashing the flattened polygons into a 2D grid. This limits the number of polygons that must be considered for intersection when the ray is cast.

- Many of the particles used within this thesis are composed of more than 10,000 polygonal faces. Collision detection can be greatly accelerated by using simplified proxy models with a few hundred faces that fully enclose the original mesh.

To guarantee a minimum spacing between adjacent particles, the annealing process can be performed with inflated proxy particles, which serve as an approximation of the original

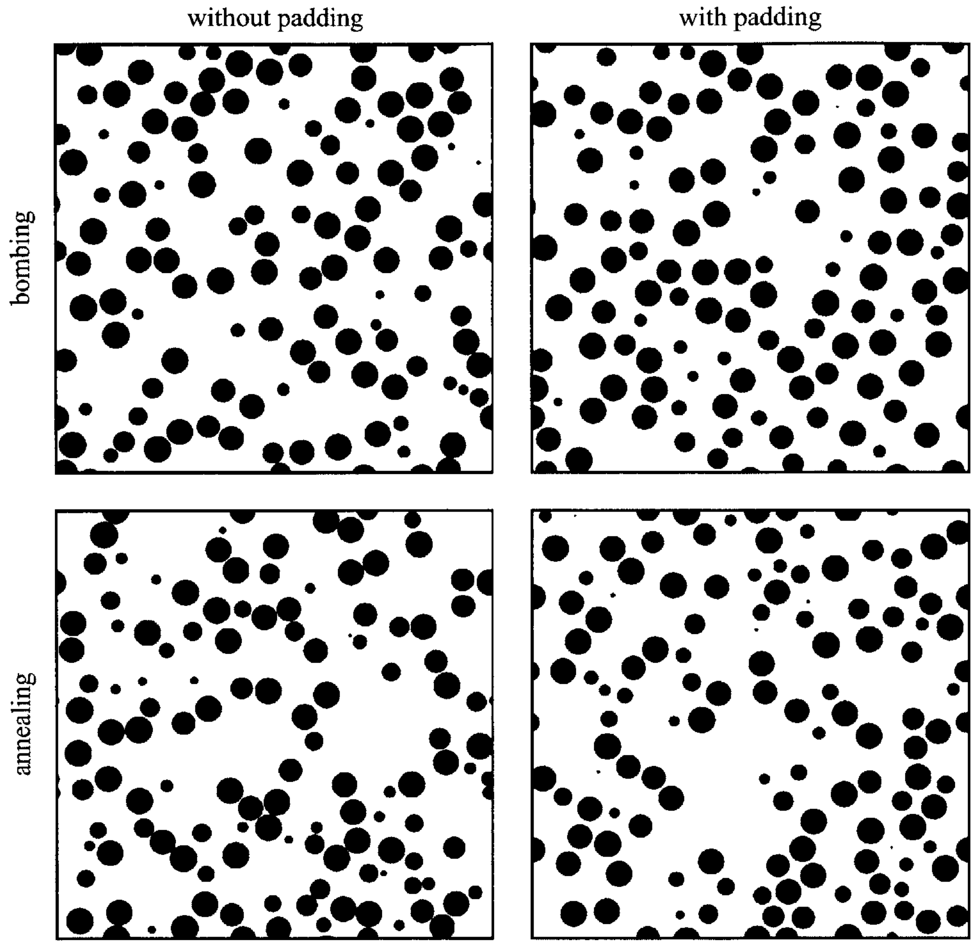


Figure 6-1: Slightly different appearances can be obtained with different annealing methods. This table compares the results obtained with bombing and annealing, without and with a guaranteed particle spacing.

shape. Figure 6-1 shows a comparison of the bombing and annealing methods, without and with a guaranteed particle spacing. When selecting an annealing method to replicate a volume, the user should consider the density and appearance of the input sample, as well as the stochastic process that created the physical material.

6.2 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 of an

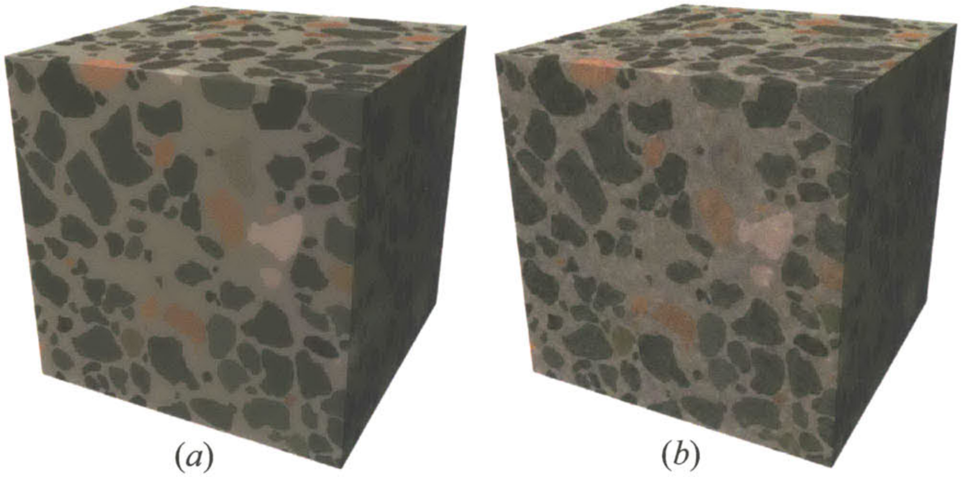


Figure 6-2: A cube of synthesized material, colored using mean profile colors (*a*) and by adding a 3D noise function (*b*).

annealed and colored synthetic volume is shown in Figure 6-2(*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 [Weisstein 2003]. The distribution recovery process can then be applied to the profiles in each color group, and the combined results can be used to populate a synthetic volume.

6.3 Adding Fine Details

As can be seen in Figure 6-2(*a*), assigning mean profile colors to individual particles yields unsatisfying result as it fails to capture color variations at a sub-particle scale. This high-frequency information can result from small-scale variation within the materials, or may be superficial noise introduced by the sample preparation and imaging processes.

To more effectively replicate the input appearance, we start by subtracting the mean color values of each profile—Figure 6-3(*b*)—from the original input image (*a*) to obtain a residual image (*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 texture synthesis algorithm [1995] in three dimen-

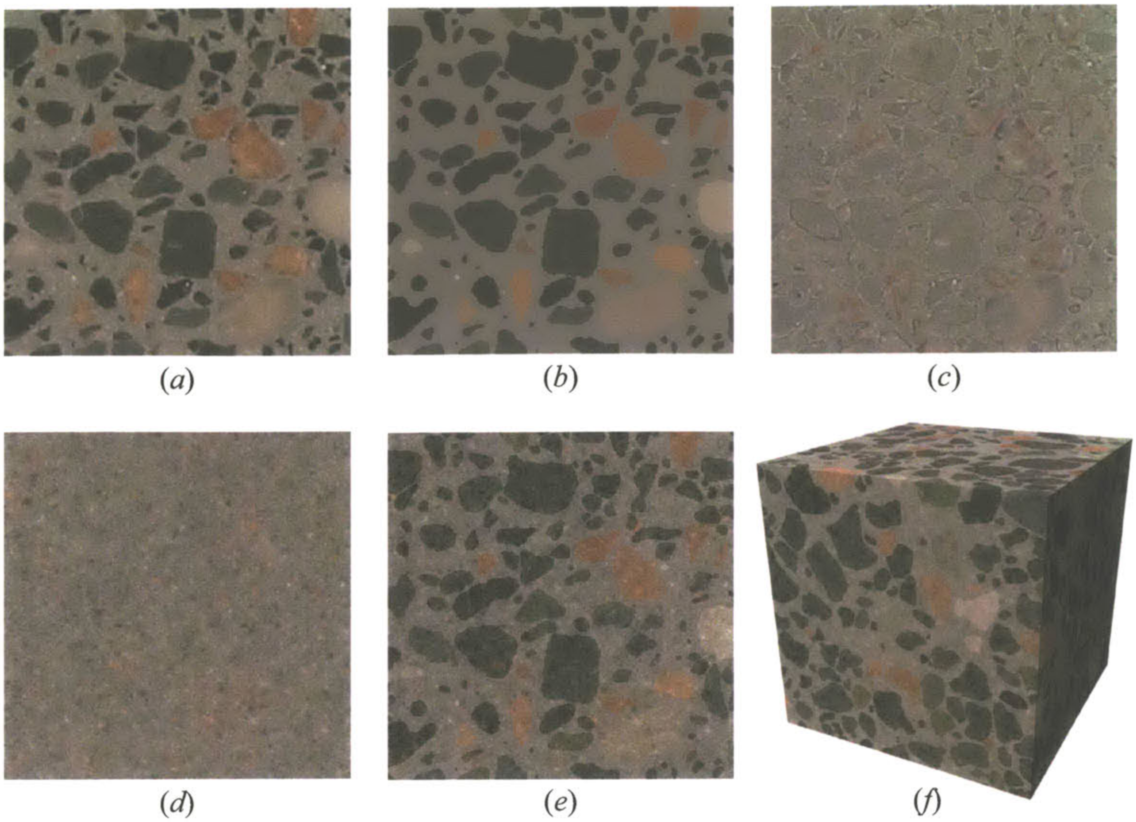


Figure 6-3: The mean color value for each profile (*b*) can be subtracted from the input image (*a*) to yield a residual (*c*). Here, the residual has been recentered around the color of the binding material for clarity. This residual lacks the macroscopic structure of the input and can be resynthesized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

sions (*d*). When the synthesized noise values are added on top of the mean color values, the resulting appearance is similar to that of the input image (*e*). This newly synthesized residual texture volume can also be added to the mean color values in a synthetic volume to yield an image that exhibits both the structure and characteristic noise frequencies of the input (*f*). For comparison, a synthetic volume is shown with and without the residual noise in Figure 6-2.

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 input profiles. Furthermore, I found that applying different noise functions to individual synthetic particles resulted in sharper visible boundaries than appear in the input images. This should be expected if the majority of the high-frequency information in an input image is a product of sample preparation and image noise rather than variation within discrete particles.

6.4 Texture Representation

After a synthetic volume is annealed, it can be stored in a relatively compact data format. Each unique particle shape that is used within the volume is stored as a watertight polygon mesh. In all of the images shown in the following chapter, no more than two distinct particle shapes are used for any given example.

An annealed volume may exhibit more than ten thousand individual particles, but each discrete particle needs only a translation vector, an isotropic scale factor, a rotation quaternion, a color, and a single byte to indicate the corresponding polygon mesh. The transformation values are stored as four-byte floating-point numbers and the color as a triplet of bytes, yielding a total memory requirement of only 36 bytes per particle. An additional three bytes store the color of the binding material.

Only the 3D residual data requires significant storage memory. The synthesized noise values are stored in a voxel cube of RGB triplets. For all of the images in the following chapter, the matrix is 128 units in each dimension, requiring a total of 6Mb of storage.

6.5 Rendering

When viewing a synthetic scene, the rendering function needs to determine if a point within the scene is inside of any particle and where that point should be indexed into the noise volume. The point-in-particle query is identical to that used for the simulated annealing process, so several of the same optimizations are applied. Particles are hashed into a spatial grid so that only the particles in the immediate vicinity of the point need to be considered. Furthermore, the polygons that define the particle geometry are projected onto a planar

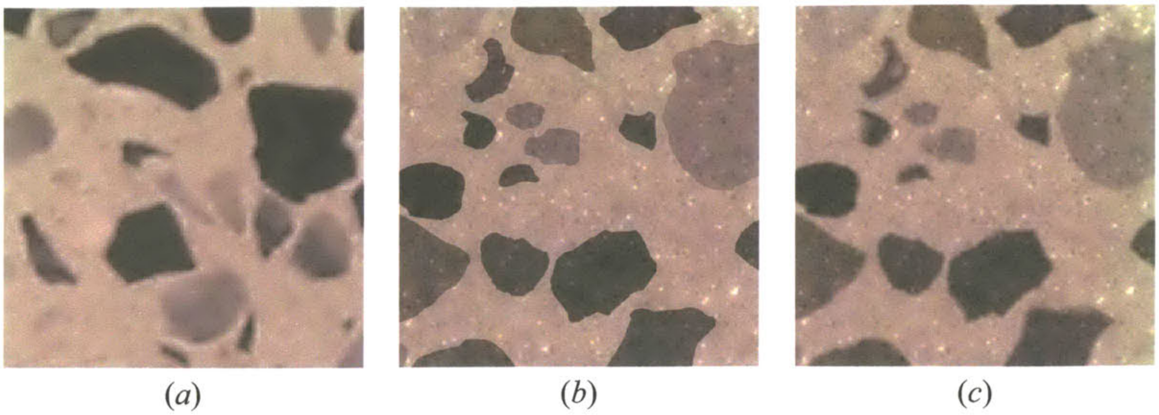


Figure 6-4: If an input texture has soft particle boundaries (*a*), then the hard edges of the synthetic result (*b*) can be softened by blending between colors near the particle surface (*c*).

surface and hashed into a 2D grid to accelerate ray-polygon intersection as described in Section 6.1.

When indexing into the noise volume, a simple modulo function enables seamless tiling in three dimensions. Between discrete samples, trilinear color interpolation is used to guarantee smoothness.

In some cases, an input image may exhibit smooth boundaries between particles and the surrounding medium, as shown in Figure 6-4(*a*). If these boundaries appear too sharp in the synthetic counterpart (*b*), then the lines can be softened by smoothly blending between the color of the particle and the binding material. In Figure 6-4(*c*), I demonstrate a linear blend between the colors as a function of distance from the surface of the particle mesh. Special care should be taken to make sure that the blending function does not change the apparent size of the particles.

6.6 Summary

This chapter has described strategies for populating a 3D volume of particles, including optimizations for accelerating the synthetic annealing process needed to create a solid texture. Also included are methods for incorporating color and noise information into the texture volume.

Colors are randomly selected for each particle from the list of mean profile colors that appear in the input image. A 3D array of residual noise values is created by applying the texture synthesis algorithm introduced by Heeger and Bergen, which performs well on stochastic inputs.

With the exception of the noise function, the data representation is relatively compact, requiring only one triangle mesh for each particle type, plus 36 bytes per particle. The noise function, however, requires 6Mb of storage for a voxel array with dimensions 128^3. To reduce this memory requirement, future research may consider the use of Perlin noise as an alternative representation for the residual function. Furthermore, future research may consider the size threshold at which repetitive artifacts become visually apparent.

As described in Chapter 4, if a volume of anisotropic particles exhibits known, nonuniform orientations, then the distribution recovery algorithm can be adapted to collect appropriate particle statistics. Similarly, if a new volume is then created from the particles, special care must be taken in the simulated annealing process to insure that the orientation constraints are respected. The simplest way to do this is to populate the volume according according to the known constraints and then modify the annealing process to affect only the particle positions—not their orientation.

We now have all of the methods necessary for extrapolating the individual parameters of an aggregate material—particle shape, particle distribution, color, and residual noise. The following two chapters will demonstrate the application of these techniques on physical inputs and will consider additional advantages of the data representation described here.

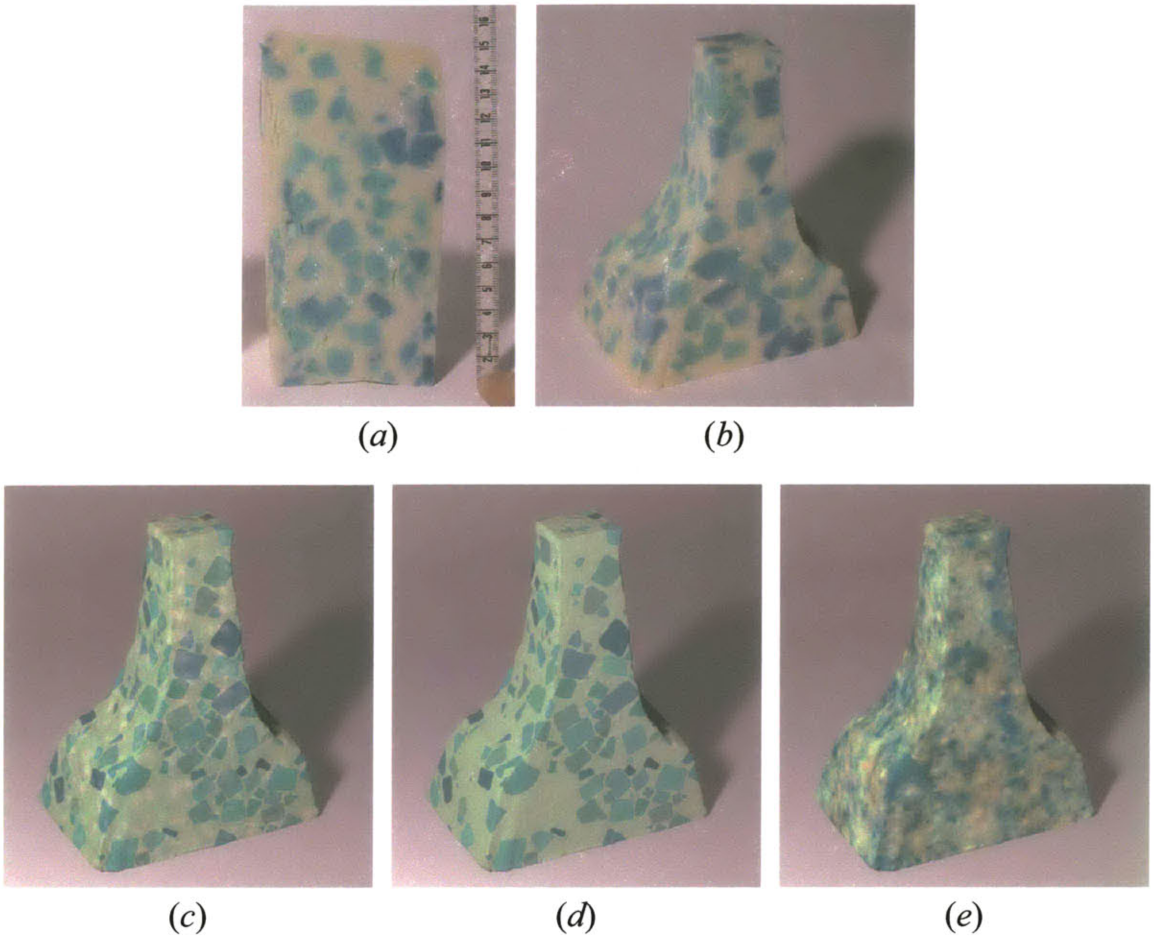
Chapter 7

Results

To test the solid texture synthesis pipeline on real-world data, I applied the algorithm to a physical model with known parameters and several physical datasets with unknown parameters. These results and corresponding analysis are discussed in the remainder of this chapter.

7.1 Test Volume

In order to test the algorithm on physical data under controlled conditions, I constructed a volume with known particle shape and distribution. Part of the volume was sliced into planar regions, as shown in Figure 7-1(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 values recovered by the stereological analysis process. Figure 7-1 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.



(*a*) (*b*)

(*c*) (*d*) (*e*)

Figure 7-1: To test the algorithm under controlled conditions, I 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, I 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*).

7.2 Physical Inputs

To test the shape and density recovery algorithm with physical inputs, I applied the methods to four different segmented textures, portions of which are shown in the left column of Figure 7-2. A single particle shape was extrapolated for each texture example using the morphed generalized cylinder method introduced in Chapter 3. I then estimated volume histograms using five equally sized bins as described in Chapter 4. These values were estimated based on between 650 and 2400 distinct profile observations. Finally, synthetic volumes were populated with the recovered particle density values, and the volumes were annealed as described in Chapter 6.

All of the density histograms shown in Figure 7-2 appear to approximate lognormal distributions. This result is expected, as lognormal particle distributions are known to be common in both natural and man-made materials [DeHoff 1965].

Figure 7-3 shows a series of textures derived from images of real-world materials. 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 $128 \times 128 \times 128$ voxels in less than three minutes.

The image pair in the lower-right corner of Figure 7-3 demonstrates some of the limits of the algorithm. In this case, individual profiles in the input image exhibit secondary structure in the form of lines, cracks, and color gradients. The residual function fails to capture this structure, which is absent in the synthesized image.

Special attention should be given to the image pair shown in Figure 7-4. For this example, the texture input is an image of a bath sponge that is composed of discrete material voids rather than a collection of particles. To obtain the results shown on the right, I analyzed the visible material voids and performed shape approximation and distribution recovery exactly as was done for the examples in Figure 7-3. I then used constructive solid geometry to subtract the particle voids from the face of a cube and rendered the resulting geometry. This highlights one of the benefits of having an explicit geometric representation of the discrete elements within a volume.

Figures 7-5 provides another example of the advantages of having an explicit represen-

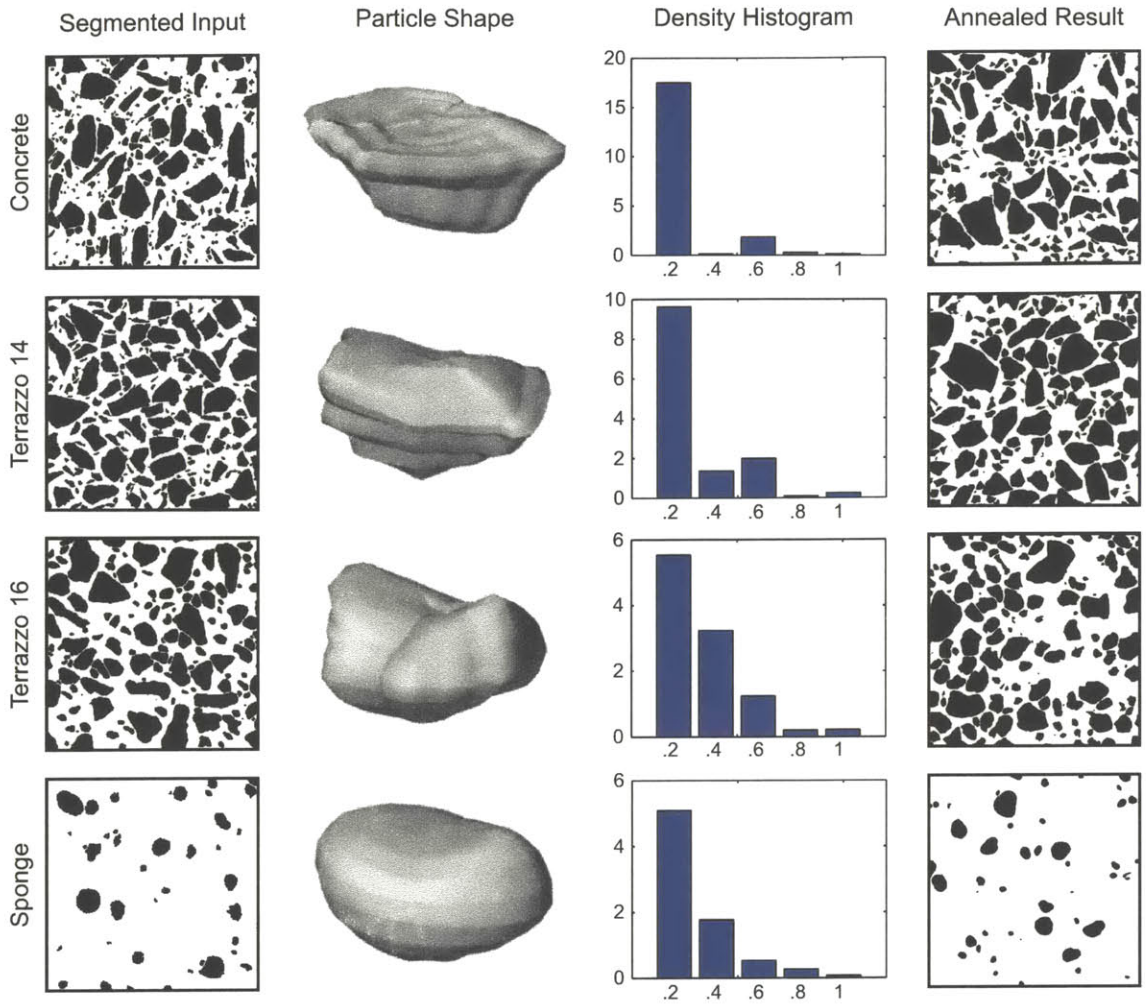


Figure 7-2: The textures shown at the far left and the extrapolated particle shapes shown in the second column are used as inputs to the particle density recovery algorithm. The resulting particle density histograms and example slices through the synthetic volumes are shown on the right.

tation of the volumetric elements of a texture. In this case, the concrete box was split with a simple plane while respecting the particle shape boundaries. Particles that crossed the split plane were assigned to one side or the other according to their center of mass. The result is a compelling image of a concrete block that effectively conveys the texture, scale, and internal structure of the material.

The most time-consuming part of the texture recovery process is the volume annealing, which can require anywhere from a few seconds to half an hour, depending on the particle density, size, and complexity. As described in Chapter 6, this process is accelerated by

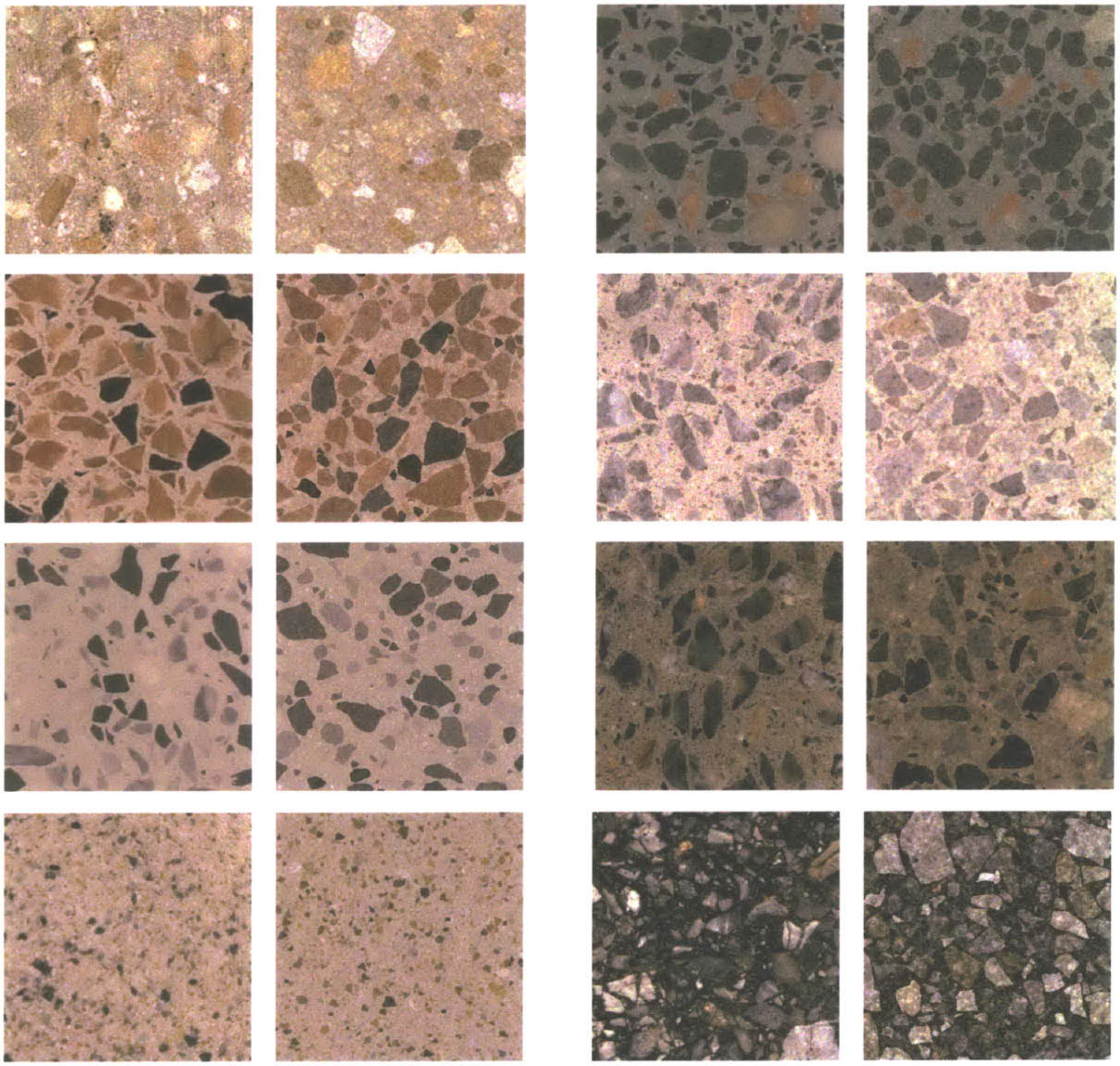


Figure 7-3: In each image pair, physical inputs to the solid texture algorithm are shown on the left and synthetic results are shown on the right.

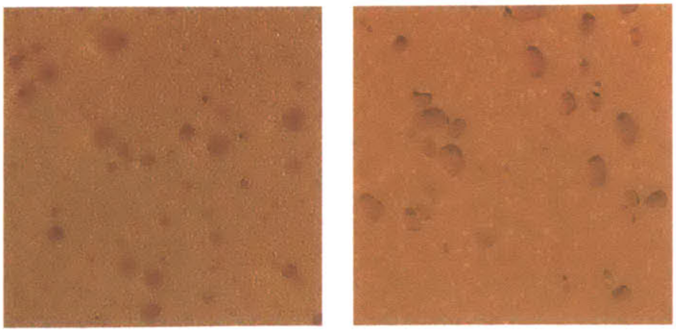


Figure 7-4: The image of the bath sponge shown on the left was reproduced on the right by applying the texture synthesis algorithm to the material voids visible in the input. The 3D model on the right was created by using constructive solid geometry to subtract the material voids from a planar surface.

applying a number of optimizations.

Figures 7-5, 7-6, and 7-7 show synthetic scenes rendered using textures recovered by the methods proposed in this thesis. All of the textures are derived from the input images shown in Figure 7-3.

7.3 Summary

This chapter demonstrated the application of the complete synthesis pipeline for extrapolating solid textures from physical 2D samples. The methods were validated by a direct comparison of a physical model and a digital replica with its appearance derived from the application of these techniques.

Several examples were shown of solid textures derived from photographs of aggregate materials. The synthetic volumes recovered using this approach were used to texture the geometry of three example scenes. The examples of the sponge texture and the concrete block demonstrate the value of having an explicit representation for the internal structure of a solid material.

In the following chapter, I discuss the implications of the methods introduced in this thesis and consider directions for future research.

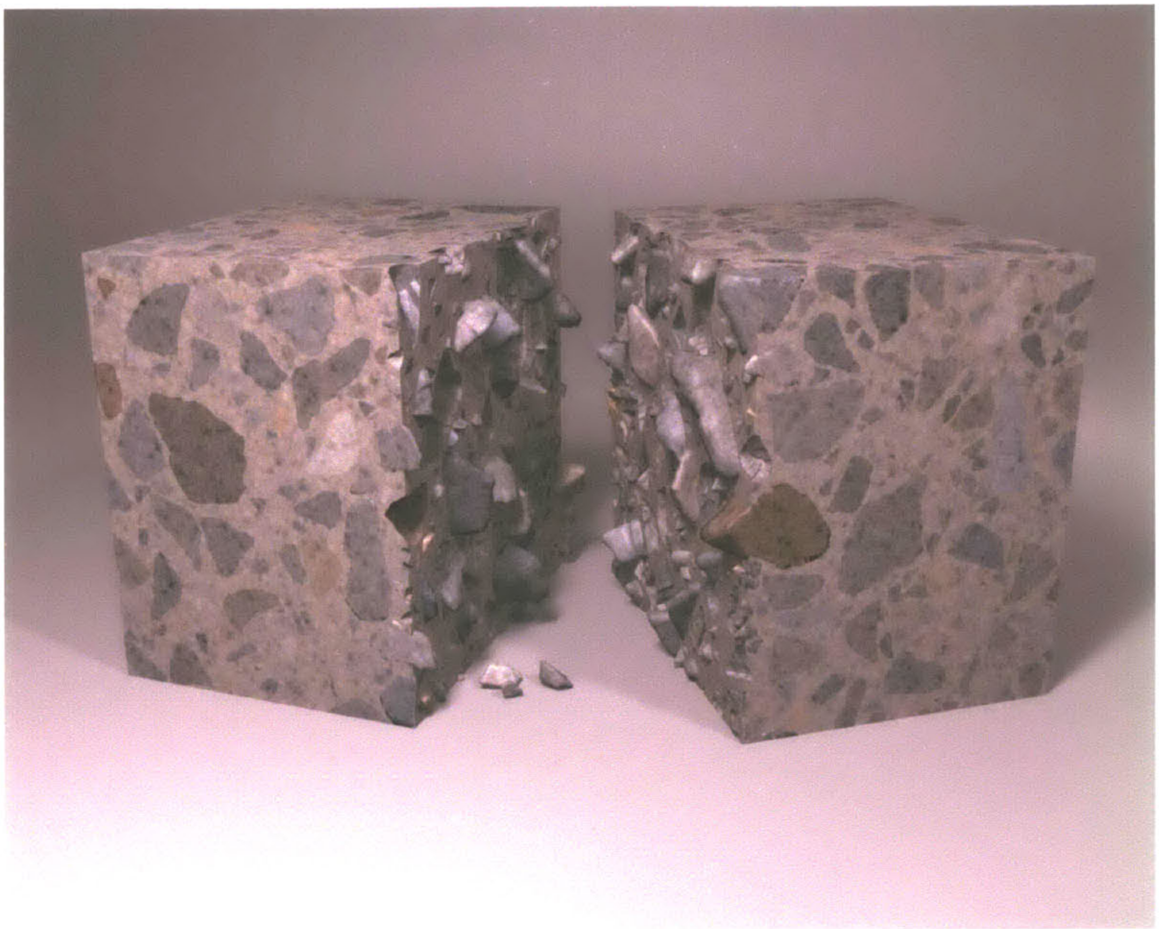


Figure 7-5: A synthetic image of a concrete box is created by splitting the geometry with a simple plane while respecting particle shape boundaries. This demonstrates an advantage of having an explicit spatial texture representation

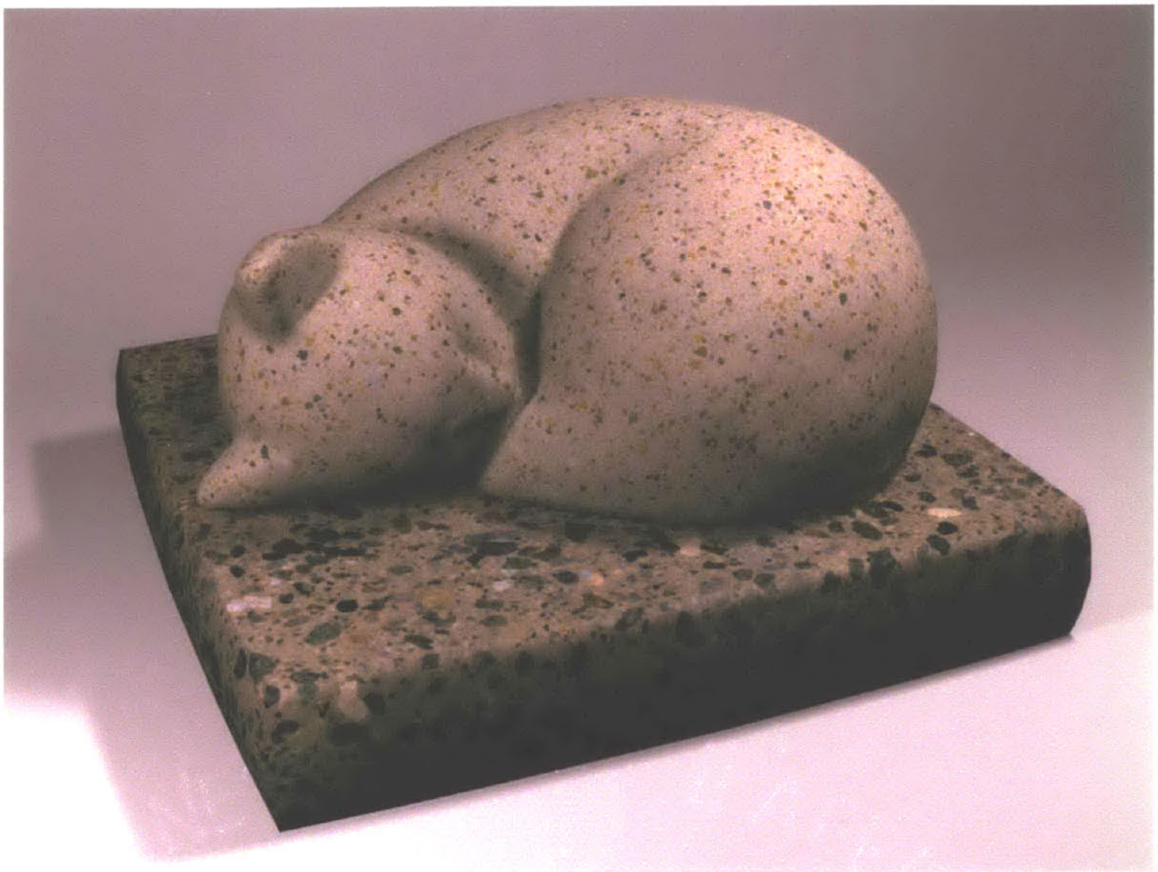


Figure 7-6: A synthetic image of a cat sculpture, rendered with solid textures extrapolated from real-world data.

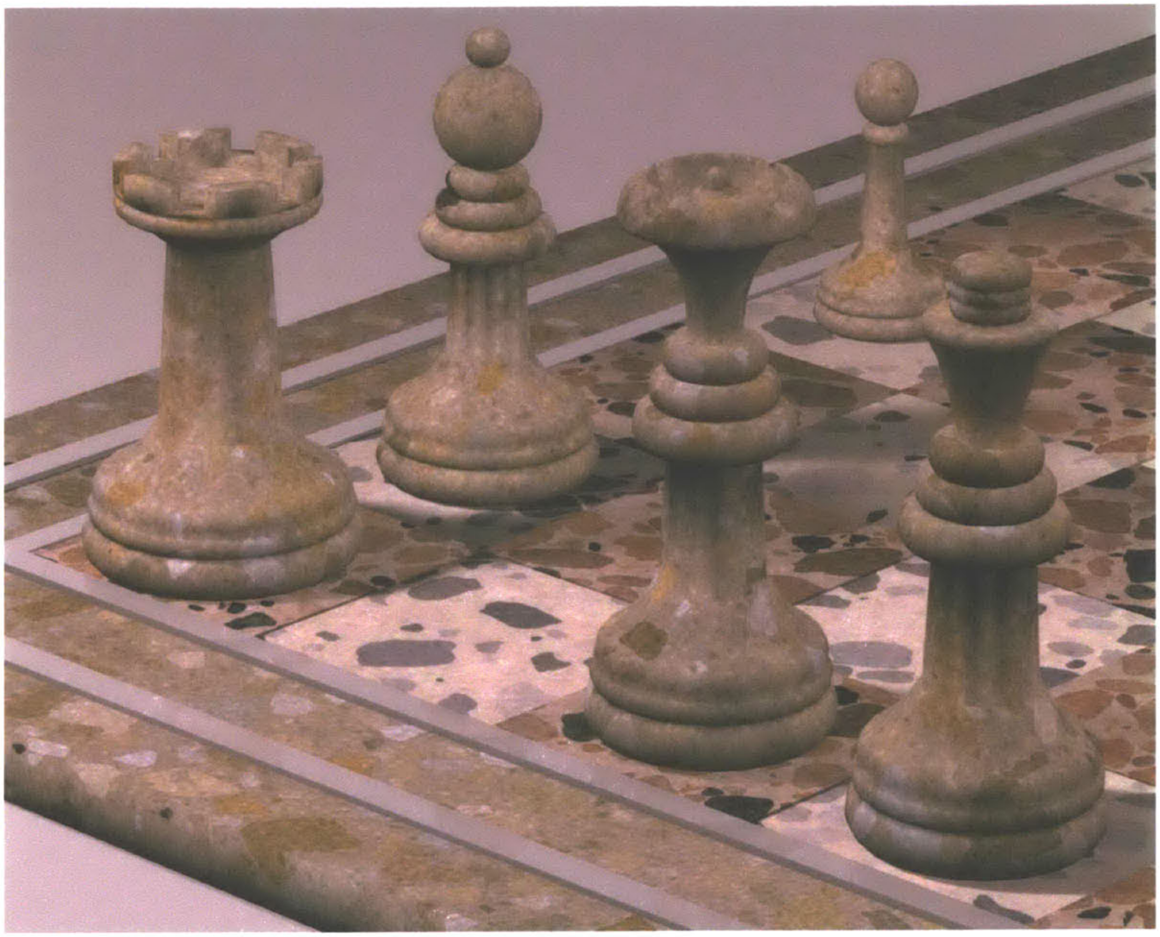


Figure 7-7: A synthetic image of a stone chess board, rendered with solid textures extrapolated from real-world data.

Chapter 8

Conclusions and Future Work

The methods described in this thesis expand the class of 3D solid textures that can be synthesized from 2D photographs. More specifically, I introduce methods that operate on textures with discrete macroscopic structure. Many of these techniques draw from existing classes of literature that offer a number of mathematically rigorous synthesis tools as new additions to the computer graphics toolbox.

The class of textures considered here offers unique challenges to existing texture synthesis algorithms, as aggregate materials exhibit a mix of stochastic and structured characteristics. By considering each texture parameter in the most appropriate domain, the methods introduced here are able to successfully synthesize compelling 3D materials that closely match the appearance of an input image.

8.1 Particle Shape Estimation

As demonstrated in Chapter 3, the morphed generalized cylinder algorithm introduced in this thesis outperforms all of the other considered methods from a psychophysical perspective. The other three methods—spherical harmonics, constructive solid geometry, and generalized cylinders—each perform well within certain domains, but are not well suited for a wide variety of input appearances.

For each of the considered shape estimation algorithms, the output shape relies on user selection of appropriate input profiles. Future research may consider automating this selection process to choose optimal 2D profiles that appropriately represent the general texture appearance. This task is particularly challenging when a wide variety of profile shapes are visible in the input image, suggesting that more than one particle shape may be needed to characterize the dataset.

For some of the shape estimation algorithms, different permutations of the input profiles may yield different particle shapes. For instance, for the generalized cylinder and morphed generalized cylinder methods, selecting a different profile as for the base curve, *b*, can affect the output. Future research may further consider these dependencies and propose alternate heuristics to optimize the 3D shape constraints.

8.2 Predicting Psychophysical Shape Similarity

The user study described in Chapter 3 ranks the performance of the shape extrapolation algorithms based on the similarity of profile shapes. The study resulted in a decisive ordering of the algorithms, ranking morphed generalized cylinders as the most effective, followed by generalized cylinders, constructive solid geometry, and spherical harmonics.

Two automated metrics—total curvature magnitude and area per perimeter squared—were marginally successful at predicting perceived shape similarity. Each of the two metrics ranked the performance of morphed generalized cylinders and generalized cylinders above that of CSG and spherical harmonics. However, there was no consensus on a total ordering.

The use of automated techniques for predicting psychophysical response is challenging because of the inherent complexity and limited knowledge of human perceptual mechanisms. Even modest contributions in this area can be of great assistance to perceptually-driven research, as it is infeasible to always seek direct user evaluation in the form of a thorough, bias-free user study. Future research should consider the use of alternative automated metrics for assessing shape similarity.

To be most applicable within the domain of texture appearance, the user study and automated metrics discussed in Chapter 3 measure shape similarity in two dimensions. If the resulting particles are to be viewed directly, as shown in Figure 7-5, then different attributes should be used as a basis for comparison. For instance, for replicating a desired 2D appearance, spherical harmonics tend to perform poorly; however, as shown in Figure 3-11, the 3D shape produced by the spherical harmonics method is arguably the most similar to the complex input shape.

8.3 Particle Distribution Recovery

In Chapter 4, I demonstrated a distribution recovery algorithm that can be applied to arbitrarily shaped particles and can be adapted for more than one particle shape. One advantage this process is that it has more mathematically rigorous foundations than a number of prior publications in the area of textures synthesis. The statistical approach provides a sound basis for synthesizing material distributions with well-defined assumptions to perform accurate predictive rendering.

8.4 Sources of Error

Due to the unconstrained nature of the solid texture synthesis algorithm, the methods described in this thesis rely heavily on assumptions to obtain plausible solutions. Errors may be introduced by poor shape estimation, quantization from histogram binning, and limited profile observations.

When approximating a distribution of spherical particles, I show in Chapter 5 that the estimation algorithm yields reliable results under a variety of input conditions. If the precise particle shapes are not known *a priori*, as is the case with most real-world materials, then we expect that an estimated particle shape is not exactly physically correct. For the purpose of matching 2D profile statistics, I demonstrate that it is difficult to guarantee that a proxy shape can be substituted for another particle with an arbitrary volumetric distribution. However, for real-world materials—which typically exhibit lognormal particle distributions—the estimation algorithm is robust to errors in the particle shape.

Ultimately, errors in the particle shape and distribution are of little concern, as we are not necessarily interested in recovering a physically accurate volume. Instead, the objective is to obtain a volume that exhibits a desired appearance through a 2D slice. As long as the estimated particle distribution is nonnegative, we may consider the solution to be sufficient from a perceptual standpoint.

8.5 Volume Reconstruction

The most time-consuming part of the texture synthesis pipeline proposed here is the simulated annealing process required for reconstruction of the particle volume. As implemented for this thesis, interparticle collision detection is accelerated with a number of optimization techniques, as described in Section 6.1. The annealing process maintains global particle density and yields a volume that can be seamlessly tiled in space.

The volume reconstruction process includes mechanisms for accurately characterizing both the color and high-frequency noise of an input image. Because the volume has infinite spatial extent, it can be mapped onto the surface of arbitrary geometry without the need for an explicit, surface-specific parameterization.

8.6 Applications

The algorithms described here are well suited for use in architectural rendering, in which faithful reproduction of appearance is important for conveying material properties. The realism of the textures also makes them suitable for use in movies, where digital scenes may be mixed with live action.

By matching input samples from existing structures or sculptures, this technology may also prove useful for historic restoration. In several well-known examples such as the Venus de Milo, the Great Sphinx of Giza, and the Parthenon, large amounts of material have been removed from the original structures. By replicating a sample material appearance, visually accurate digital models of restored structures might be produced.

In order to use the solid textures in real-time application, the textures can be projected onto the surface of a model at a desired resolution [Cutler 2003]. This precomputation step enables the use of the solid textures in applications such as games or architectural walk-throughs.

In contrast to voxel-based representations for texture data, the material description used here explicitly maintains the geometry of the individual particle shapes. This representation requires less memory than a voxel-based approach while allowing for sharp particle boundaries suitable for high-fidelity image rendering. Furthermore, this explicit particle representation is amenable to physical simulation.

Two examples that take advantage of this explicit structure are shown in Figures 7-4 and 7-5. In these images, the particle geometry is incorporated into the scene geometry. These models were created with simple CSG operations, yet the concrete box in particular conveys the complexity of physical simulation.

Another advantage of the data representation is that parameters for particle shape, distribution, color, and noise are each recovered independently. By mixing and matching recovered parameters from a variety of input sources, novel materials can be synthesized in an intuitive way. For instance, an artist might create a hybrid material by choosing particle shapes from one image, particle distribution from another image, and color and noise data from two additional sources. This approach to texture authoring is shown in Figure 8-1, which demonstrates the power of an example-based framework.

8.7 Limitations

The texture synthesis algorithm proposed here has been shown to be effective for replicating a number of example textures. However, in some cases, discrete particles may exhibit secondary structure, which is typically not captured by the residual volume. In future work, additional parameters may be considered to capture this variation in appearance.

One limitation of this research is that it assumes input images to have an isotropic appearance. Without this assumption, it is unclear from a single image how anisotropic structure should be characterized in a 3D volume. Future research may consider the appearance of a material as seen in orthogonal volumetric slices in order to extend these techniques to be applicable to a greater variety of input textures. In some simple examples, a matrix may be computed to reproject anisotropic textures into an isotropic domain where the texture

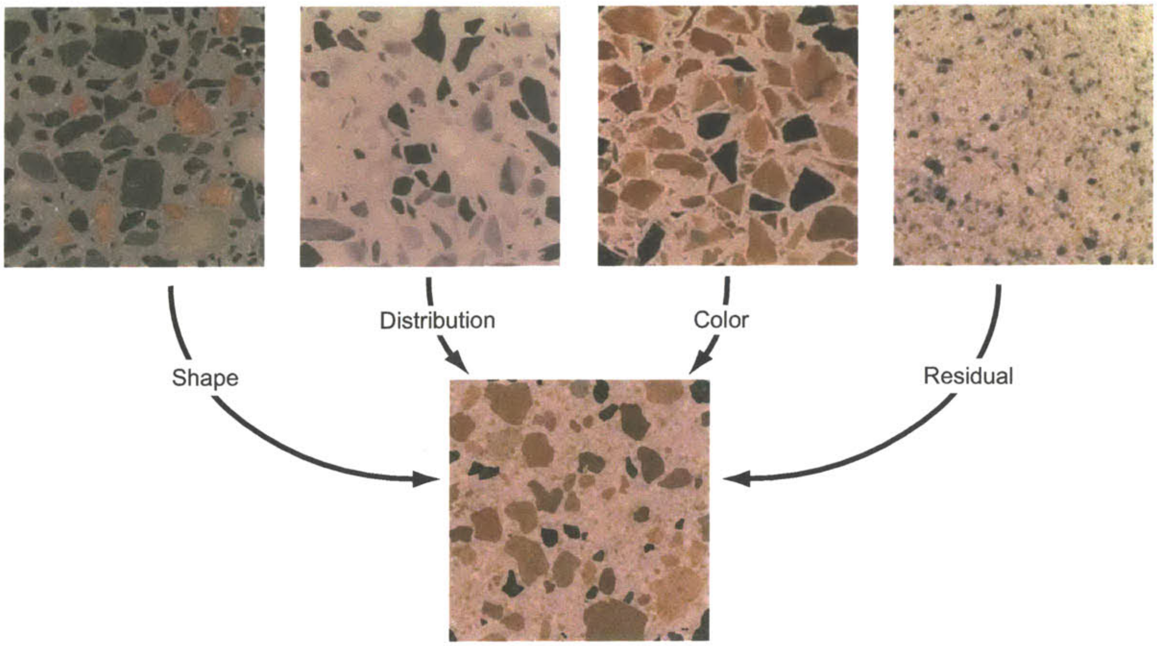


Figure 8-1: A novel texture is produced by mixing texture parameters recovered from four different input images.

synthesis algorithms can be applied as usual.

8.8 Future Work

Just as this thesis builds on a large collection of related research, there is ample opportunity to extend the contributions introduced here into other domains. In this section, I present opportunities for future work, not only in the area of solid texture synthesis, but also in a variety of related disciplines.

8.8.1 Geometry Compression

Within this research, I framed the shape recovery process to operate in the context of texture synthesis, but the algorithms described in Chapter 3 may also be applied to other problems within computer vision and computer graphics. Since profiles offer a compact representation of shape, some of these algorithms may be considered for geometry compression. In particular, the morphed generalized cylinder can be adapted to use any number of input curves to define shape constraints around the axis of rotation. This suggests that an existing geometric model may be compressed by adding wireframe constraints until a desired accuracy is obtained.

8.8.2 Geometric Modeling

The generalized cylinder has a long history of use in the computer vision community for its ability to express a wide array of shapes with very compact parameters. Likewise, the morphed generalized cylinder allows for even greater flexibility in a geometric representation with intuitive 2D constraints. This suggests that the representation may be used as a general modeling tool for authoring 3D geometry from simple, limited user input. Due to the simplicity of the construction algorithm, changes made to a control curve can be immediately reflected in a 3D model.

As described in Chapter 3, the interpolation scheme used by the morphed generalized cylinder algorithm is based on normalized arc length. This approach works reasonably well for simple particle shapes, but it should be expected that this simple heuristic will yield poor results if the shape constraints are sufficiently complicated. Future research may consider different ways of enforcing smoothness between morph curve constraints if more complicated geometric models are desired.

8.8.3 Metrics for Predicting Shape Similarity

Chapter 3 considered the use of two automated metrics for predicting the psychophysical evaluation of shape similarity. These metrics were only moderately successful in anticipating the results of the user study. Future research may consider the use of other perceptually-motivated metrics that measure symmetry, presence of sharp corners, or other 2D shape features.

In particular, I believe it would be useful to derive a metric that computes the distribution of curvature along the edges of a 2D polygon. Such a metric could indicate the presence of sharp corners, as well as smooth lines along the boundary of a 2D shape. In practice, I have had difficult in deriving such a metric, as the methods that I have considered are sensitive to polygon tessellation and coincident or near-coincident vertices.

The user study described in Chapter 3 deliberately eliminates variability in the particle size distribution in order to focus attention on particle shape. However, it should be acknowledge that size distribution may play a role in shape perception. Future research may incorporate this variable into a psychophysical evaluation of shape similarity.

8.8.4 Image Segmentation

One additional remaining challenge in the image processing pipeline is that of robust, fully-automated image segmentation. For most of the examples shown in this thesis, I was able to segment input textures by using simple color filtering. However, the segmented textures often required some degree of manual clean-up, particularly for separating closely-spaced particles that may appear conjoined in the segmented image. Image segmentation is a long-standing problem within the computer vision community, and this research will benefit from future contributions in this area.

8.8.5 Residual Volume Compression

With the exception of the residual volume, the texture representation described here is relatively compact. Other representations for the residual function may be considered, such as an adaptation of Perlin noise [Perlin 1985; Perlin and Hoffert 1989]. If this component of the texture can be represented more concisely, then the textures described here may be considered for real-time rendering with pixel shaders on specialized graphics hardware.

8.8.6 Level of Detail

For the results shown in this thesis, most of the volumes exhibit either one or two particle shapes with approximately 10,000 triangles. With these large triangle meshes, rendering may be unnecessarily time-consuming when most of the geometric detail in the particles cannot be discerned. Since the individual particles within the volume have an explicit geometric representation, it is likely that existing level-of-detail algorithms can be adapted to reduce their geometric complexity. If a particle appears sufficiently small in screen space, then it can be replaced by a proxy model without an appreciable impact on the rendered scene.

8.8.7 Example-Based Texture Authoring

The example-based texture synthesis framework depicted in Figure 8-1 illustrates the advantage of independently recovering different material parameters. This type of interface can allow even a novice user to author textures without having any explicit knowledge of the underlying representation.

In this thesis, I considered methods for recovering four specific parameters for a particular category of materials. It would be useful to extend this approach to operate on input images with a greater variety of appearances.

One of the challenges of procedural shaders is that they tend not to operate within a perceptually uniform parameter space. In contrast, the representation shown here offers possibilities to interpolate between parameters recovered from different textures. For instance, a user may desire a particle distribution that falls somewhere between two given texture examples. This may be obtained simply by interpolating the values in the density histograms. Similarly, if a user desires a hybrid particle shape, then a method may be considered for incorporating profiles from two or more example images into a single geometric model. In contrast to a typical procedural shader, this framework may allow for intuitive texture authoring in a psychophysically-motivated parameter space.

8.9 Summary

This thesis offers a relatively mature, comprehensive system for synthesizing solid textures of aggregate materials from example images. I have demonstrated the effectiveness of the approach with a number of texture inputs, and have illustrated its use in rendering visually compelling scenes.

In addition to the texture synthesis algorithm, this document incorporates a number of contributions, including the morphed generalized cylinder representation, a psychophysical evaluation of shape extrapolation methods, and a baseline for using automated metrics to predict perceptual similarity of 2D shapes.

Just as this thesis draws from prior contributions in a variety of different fields, hopefully this work will demonstrate its utility in areas of research that extend beyond computer graphics, computer vision, human perception, and stereology.

Appendix A

Consent Form for User Study

CONSENT TO PARTICIPATE IN
NON-BIOMEDICAL RESEARCH

Relative performance of 3D shape reconstruction algorithms

You are asked to participate in a research study conducted by Robert Jagnow, from the Department of Electrical Engineering and Computer Science at the Massachusetts Institute of Technology (M.I.T.), and Julie Dorsey, at the Department of Computer Science at Yale University. The results of the study will contribute to the Ph.D. thesis of Robert Jagnow. You were selected as a possible participant in this study because of your affiliation with the computer graphics community. You should read the information below, and ask questions about anything you do not understand, before deciding whether or not to participate.

• PARTICIPATION AND WITHDRAWAL

Your participation in this study is completely voluntary and you are free to choose whether to be in it or not. If you choose to be in this study, you may subsequently withdraw from it at any time without penalty or consequences of any kind. The investigator may withdraw you from this research if circumstances arise which warrant doing so.

• PURPOSE OF THE STUDY

This study is designed to assess the relative performance of four different algorithms for approximating 3D shapes from 2D input. This research is applicable to the field of texture synthesis, in which synthetic images are derived from input photographs that depict a desired appearance. In this work, we start with four different three-dimensional particle shapes, each of which is then approximated using four different algorithms, resulting in a total of 16 different decoy models. Users will be asked to compare the appearance of 2D slices through the resulting 3D particles in order to evaluate the effectiveness of each algorithm from a perceptual standpoint.

• PROCEDURES

If you volunteer to participate in this study, we would ask you to do the following things:

You will be asked to perform approximately 80 different comparisons in a computer-based test to evaluate image similarity. On each screen, you will see three images in a row. The center image will show a collection of black shapes on a white background. Of the two adjacent images, you will click the image that is least similar to the image in the center. More specifically, you should evaluate the similarity of the three images based on the shape of the black regions rather than on the size or density of the regions.

To help you prepare for the study, you will be given an example test with only five to ten questions, which should take less than two minutes to complete. The complete test should take less than 30 minutes to complete.

All testing will take place in an office in MIT building 32.

- **POTENTIAL RISKS AND DISCOMFORTS**

Each participant in the study will be interacting with a mouse and clicking every five to twenty seconds over the course of a twenty- to thirty-minute session. If you have a prior history of a work-related repetitive stress injury, you may want to reconsider your participation in the study.

- **POTENTIAL BENEFITS**

It is not expected that you will receive any direct, personal benefits as a result of your participation in this study.

This research will be used to evaluate the relative performance of four different algorithms for the approximation of 3D shape from incomplete 2D information. As such, the results of the study will help to contribute to the general knowledge of the computer graphics and human perception communities.

- **PAYMENT FOR PARTICIPATION**

No financial compensation will be offered in exchange for participation in this study.

- **CONFIDENTIALITY**

Any information that is obtained in connection with this study and that can be identified with you will remain confidential and will be disclosed only with your permission or as required by law.

The only identifiable information that will be included in this study are the participant's name and e-mail address. This information will be stored electronically and will be accessible only to the researchers who are directly involved in administering the study. Data will be electronically archived following the study. If other researchers use the data in future projects, personal identifiable information will be excluded.

- **IDENTIFICATION OF INVESTIGATORS**

If you have any questions or concerns about the research, please feel free to contact one of the following investigators:

Robert Jagnow, Principal Investigator
Daytime phone: 617-258-5090
E-mail address: rjagnow@graphics.csail.mit.edu

Julie Dorsey, Faculty Sponsor
Daytime phone: 203-432-4249
E-mail address: dorsey@cs.yale.edu

- **EMERGENCY CARE AND COMPENSATION FOR INJURY**

"In the unlikely event of physical injury resulting from participation in this research you may receive medical treatment from the M.I.T. Medical Department, including emergency treatment and follow-up care as needed. Your insurance carrier may be billed for the cost of such treatment. M.I.T. does not provide any other form of compensation for injury. Moreover, in either providing or making such medical care available it does not imply the injury is the fault of the investigator. Further information may be obtained by calling the MIT Insurance and Legal Affairs Office at 1-617-253 2822."

- **RIGHTS OF RESEARCH SUBJECTS**

You are not waiving any legal claims, rights or remedies because of your participation in this research study. If you feel you have been treated unfairly, or you have questions regarding your rights as a research subject, you may contact the Chairman of the Committee on the Use of Humans as Experimental Subjects, M.I.T., Room E32-335, 77 Massachusetts Ave, Cambridge, MA 02139, phone 1-617-253 6787.

SIGNATURE OF RESEARCH SUBJECT OR LEGAL REPRESENTATIVE

I understand the procedures described above. My questions have been answered to my satisfaction, and I agree to participate in this study. I have been given a copy of this form.

Name of Subject

Name of Legal Representative (if applicable)

Signature of Subject or Legal Representative Date

SIGNATURE OF INVESTIGATOR

In my judgment the subject is voluntarily and knowingly giving informed consent and possesses the legal capacity to give informed consent to participate in this research study.

Signature of Investigator Date

Bibliography

ASHIKHMIN, M. 2001. Synthesizing natural textures. In *Symposium on Interactive 3D Graphics*, 217–226.

BAREQUET, G., AND SHARIR, M. 1994. Piecewise-linear interpolation between polygonal slices. In *Proceedings of the tenth annual symposium on Computational geometry*, ACM Press, 93–102.

BAREQUET, G., SHAPIRO, D., AND TAL, A. 1996. History consideration in reconstructing polyhedral surfaces from parallel slices. In *Proceedings of the 7th conference on Visualization '96*, IEEE Computer Society Press, 149–ff.

BAREQUET, G., GOODRICH, M. T., LEVI-STEINER, A., AND STEINER, D. 2003. Straight-skeleton based contour interpolation. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, Society for Industrial and Applied Mathematics, 119–127.

BAUMGART, B. G. 1972. Winged edge polyhedron representation. Tech. rep.

BEN-SHAHAR, O., HUGGINS, P., IZO, T., AND ZUCKER, S. 2003. Cotrical connections and early visual function: Intra- and inter-columnar processing. *Journal of Physiology–Paris 97*, 2–3, 191–208.

BENNIS, C., VÉZIEN, J.-M., AND IGLÉSIAS, G. 1991. Piecewise surface flattening for non-distorted texture mapping. In *Proceedings of the 18th annual conference on Computer Graphics and Interactive Techniques*, ACM Press, 237–246.

BHAT, P., INGRAM, S. F., AND TURK, G. 2004. Geometric texture synthesis by example. In *Geometry Processing 2004 (Eurographics/ACM SIGGRAPH Symposium)*, 43–46.

BIER, E., AND SLOAN, K. 1986. Two-part texture mapping. In *IEEE Computer Graphics and Applications*, 40–53.

BINFORD, T. 1971. Visual perception by computer. In *Proceedings, IEEE Conference on Systems Science and Cybernetics*.

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

BLINN, J. F. 1978. Simulation of wrinkled surfaces. *Communications of the ACM 12*, 3, 286–292.

BOISSONNAT, J.-D. 1988. Shape reconstruction from planar cross sections. *Comput. Vision Graph. Image Process. 44*, 1, 1–29.

BOUTEILLER, J.-M., AND BAUDRY, M. 2002. Neuroanatomical imaging: Constrained 3D reconstruction using variational implicit techniques. In *3DPVT*, 62–65.

BROOKS, R. A. 1981. Symbolic reasoning among 3D models and 2D images. *Artificial Intelligence 17*, 285–348.

CARR, N. A., AND HART, J. C. 2004. Painting detail. *ACM Trans. Graph. 23*, 3, 845–852.

CATMULL, E. E. 1974. *A subdivision algorithm for computer display of curved surfaces*. PhD thesis, University of Utah.

COBB, G. W. 1998. *Introduction to Design and Analysis of Experiments*. Springer-Verlag Telos.

COOK, R. L. 1984. Shade trees. In *Computer Graphics (Proceedings of ACM SIGGRAPH 84)*, ACM, 223–231.

CUTLER, B., DORSEY, J., MCMILLAN, L., MÜLLER, M., AND JAGNOW, R. 2002. A procedural approach to authoring solid models. In *Proceedings of the 29th annual conference on Computer Graphics and Interactive Techniques*, ACM Press, 302–311.

CUTLER, B. M. 2003. *Procedural authoring of solid models*. PhD thesis, Massachusetts Institute of Technology.

DANA, K. J., VAN GINNEKEN, B., NAYAR, S. K., AND KOENDERINK, J. J. 1999. Reflectance and texture of real-world surfaces. *ACM Trans. Graph. 18*, 1, 1–34.

DE ARAUJO, B., AND JORGE, J. 2003. Blobmaker: Free-form modelling with variational implicit surfaces. In *Proceedings of 12^o Encontro Português de Computação Grafica (12^o EPCG), Porto, Portugal*, 17–26.

DE BONET, J. S. 1997. Multiresolution sampling procedure for analysis and synthesis of texture images. In *Proceedings of the 24th annual conference on Computer Graphics and Interactive Techniques*, ACM Press/Addison-Wesley Publishing Co., 361–368.

DEHOFF, R. T. 1965. The estimation of particle-size distributions from simple counting measurements made on random plane sections. *Transactions of the Metallurgical Society of AIME 233*, 25–29.

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.

DOBBINS, A., ZUCKER, S. W., AND CYNADER, M. S. 1987. Endstopped neurons in the visual cortex as a substrate for calculating curvature. *Nature 329*, 6138 (October), 438–441.

DORSEY, J., AND HANRAHAN, P. 1996. Modeling and rendering of metallic patinas. In *Proceedings of the 23rd annual conference on Computer Graphics and Interactive Techniques*, ACM Press, 387–396.

DORSEY, J., PEDERSEN, H. K., AND HANRAHAN, P. 1996. Flow and changes in appearance. In *Proceedings of the 23rd annual conference on Computer Graphics and Interactive Techniques*, ACM Press, 411–420.

DORSEY, J., EDELMAN, A., JENSEN, H. W., LEGAKIS, J., AND PEDERSEN, H. K. 1999. Modeling and rendering of weathered stone. In *Proceedings of the 26th annual conference on Computer Graphics and Interactive Techniques*, ACM Press/Addison-Wesley Publishing Co., 225–234.

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

EDVARDSON, H., AND SMEDBY, O. 2002. Compact and efficient 3D shape description through radial function approximation. In *Computer Methods and Programs in Biomedicine*, vol. 72, 89–97.

EFROS, A. A., AND FREEMAN, W. T. 2001. Image quilting for texture synthesis and transfer. In *Proceedings of ACM SIGGRAPH 2001*, ACM Press, 341–346.

EFROS, A. A., AND LEUNG, T. K. 1999. Texture synthesis by non-parametric sampling. In *Proceedings of the International Conference on Computer Vision-Volume 2*, IEEE Computer Society, 1033.

FANG, Y.-H., CHOU, H.-L., AND CHEN, Z. 2003. 3D shape recovery of complex objects from multiple silhouette images. *Pattern Recogn. Lett. 24*, 9-10, 1279–1293.

FIX, J. D., AND LADNER, R. E. 1998. Multiresolution banded refinement to accelerate surface reconstruction from polygons. In *Proceedings of the fourteenth annual symposium on Computational geometry*, ACM Press, 240–248.

FLEISCHER, K. W., LAIDLAW, D. H., CURRIN, B. L., AND BARR, A. H. 1995. Cellular texture generation. In *Proceedings of the 22nd annual conference on Computer Graphics and Interactive Techniques*, ACM Press, 239–248.

GARDNER, R. J., HOBOLTH, A., JENSEN, E. B. V., AND SØRENSEN, F. B. 2004. Shape discrimination by total curvature, with a view to cancer diagnostics.

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

GEIGER, B. 1993. *Three-dimensional modeling of human organs and its application to diagnosis and surgical planning*. PhD thesis, INRIA.

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

GU, X., GORTLER, S. J., AND HOPPE, H. 2002. Geometry images. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer Graphics and Interactive Techniques*, ACM Press, 355–361.

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

HAKER, S., ANGENENT, S., TANNENBAUM, A., KIKINIS, R., SAPIRO, G., AND HALLE, M. 2000. Conformal surface parameterization for texture mapping. *IEEE Transactions on Visualization and Computer Graphics 6*, 2, 181–189.

HECKBERT, P. S. 1986. Survey of texture mapping. *IEEE Comput. Graph. Appl. 6*, 11, 56–67.

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.

HERTZMANN, A., JACOBS, C. E., OLIVER, N., CURLESS, B., AND SALESIN, D. H. 2001. Image analogies. In *Proceedings of the 28th annual conference on Computer Graphics and Interactive Techniques*, ACM Press, 327–340.

HIROTA, K., TANOUE, Y., AND KANEKO, T. 1998. Generation of crack patterns with a physical model. *The Visual Computer 14*, 3, 126–137.

HOBOLTH, A., AND JENSEN, E. B. V. 2002. Stereological analysis of shape. *Image Analysis and Stereology 21*, Supplement 1, S23–S29.

HOBOLTH, A. 2003. The spherical deformation model. *Biostatistics 4*, 4, 583–595.

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

IGARASHI, T., MATSUOKA, S., AND TANAKA, H. 1999. Teddy: A sketching interface for 3D freeform design. *Proceedings of SIGGRAPH 99*, 409–416. ISBN 0-20148-560-5. Held in Los Angeles, California.

JAGNOW, R., DORSEY, J., AND RUSHMEIER, H. 2004. Stereological techniques for solid textures. In *Proceedings of ACM SIGGRAPH 2004*, Computer Graphics Proceedings, Annual Conference Series, ACM, 329–335.

KAJIYA, J. T., AND KAY, T. L. 1989. Rendering fur with three dimensional textures. In *SIGGRAPH '89: Proceedings of the 16th annual conference on Computer Graphics and Interactive Techniques*, ACM Press, 271–280.

KARPENKO, O., HUGHES, J., AND RASKAR, R. 2002. Free-form sketching with variational implicit surfaces. *Computer Graphics Forum 21*, 3, 585–594.

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.

KUTULAKOS, K. N., AND SEITZ, S. M. 2000. A theory of shape by space carving. *Int. J. Comput. Vision 38*, 3, 199–218.

KUTULAKOS, K. N. 2000. Approximate n-view stereo. In *ECCV (1)*, 67–83.

KWATRA, V., SCHÖDL, A., ESSA, I., TURK, G., AND BOBICK, A. 2003. Graphcut textures: image and video synthesis using graph cuts. *ACM Trans. Graph. 22*, 3, 277–286.

LAURENTINI, A. 1994. The visual hull concept for silhouette-based image understanding. *IEEE Trans. Pattern Anal. Mach. Intell. 16*, 2, 150–162.

LAURENTINI, A. 1995. How far 3D shapes can be understood from 2D silhouettes. *IEEE Trans. Pattern Anal. Mach. Intell. 17*, 2, 188–195.

LAZEBNIK, S., BOYER, E., AND PONCE, J. 2001. On computing exact visual hulls of solids bounded by smooth surfaces. In *Proceedings of Computer Vision and Pattern Recognition*, 156–161.

LEE, A. W. F., SWELDENS, W., SCHRÖDER, P., COWSAR, L., AND DOBKIN, D. 1998. MAPS: multiresolution adaptive parameterization of surfaces. In *Proceedings of the 25th annual conference on Computer Graphics and Interactive Techniques*, ACM Press, 95–104.

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

LÉVY, B., AND MALLET, J.-L. 1998. Non-distorted texture mapping for sheared triangulated meshes. In *Proceedings of the 25th annual conference on Computer Graphics and Interactive Techniques*, ACM Press, 343–352.

LÉVY, B., PETITJEAN, S., RAY, N., AND MAILLOT, J. 2002. Least squares conformal maps for automatic texture atlas generation. *ACM Trans. Graph. 21*, 3, 362–371.

LÉVY, B. 2001. Constrained texture mapping for polygonal meshes. In *Proceedings of the 28th annual conference on Computer Graphics and Interactive Techniques*, ACM Press, 417–424.

LI, Q., LI, F., SHIRAISHI, J., KATSURAGAWA, S., SONE, S., AND DOI, K. 2003. Investigation of new psychophysical measures for evaluation of similar images on thoracic computed tomography for distinction between benign and malignant nodules. *Medical Physics 30*, 10 (October), 2584–2593.

LONCARIC, S. 1998. A survey of shape analysis techniques. *Pattern Recognition 31*, 8, 983–1001.

MA, S., AND LIN, H. 1988. Optimal texture mapping. In *In Proceedings of EUROGRAPHICS '88*, 421–428.

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.

MAILLOT, J., YAHIA, H., AND VERROUST, A. 1993. Interactive texture mapping. In *Proceedings of the 20th annual conference on Computer Graphics and Interactive Techniques*, ACM Press, 27–34.

MARR, D., AND NISHIHARA, H. K. 1978. Representation and recognition of the spatial organization of three dimensional shapes. In *Proceedings of the Royal Society*, vol. B200, 269–294.

MARTENS, W. L., AND MYSZKOWSKI, K. 1998. Psychophysical validation of the visible differences predictor for global illumination applications. In *Proceedings of IEEE Visualization '98*, 49–52.

MARTIN, W. N., AND AGGARWAL, J. K. 1983. Volumetric descriptions of objects from multiple views. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-5*, 2, 150–158.

MATUSIK, W., BUEHLER, C., RASKAR, R., GORTLER, S. J., AND MCMILLAN, L. 2000. Image-based visual hulls. In *Proceedings of the 27th annual conference on Computer Graphics and Interactive Techniques*, ACM Press/Addison-Wesley Publishing Co., 369–374.

MCNAMARA, A., CHALMERS, A., TROSCIANKO, T., AND GILCHRIST, I. 2000. Comparing real and synthetic scenes using human judgements of lightness. In *Proceedings of the Eurographics Workshop in Brno.*

MERILLOU, S., DISCHLER, J.-M., AND GHAZANFARPOUR, D. 2001. Corrosion: simulating and rendering. In *Graphics Interface 2001*, Canadian Information Processing Society, 167–174.

MEYER, G. W., RUSHMEIER, H. E., COHEN, M. F., GREENBERG, D. P., AND TORRANCE, K. E. 1986. An experimental evaluation of computer graphics imagery. *ACM Trans. Graph. 5*, 1, 30–50.

MEYERS, D., SKINNER, S., AND SLOAN, K. 1992. Surfaces from contours. *ACM Trans. Graph. 11*, 3, 228–258.

MOEZZI, S., KATKERE, A., KURAMURA, D. Y., AND JAIN, R. 1996. Reality modeling and visualization from multiple video sequences. *IEEE Comput. Graph. Appl. 16*, 6, 58–63.

MORTENSON, M. 1999. *Mathematics for Computer Graphics Applications: An Introduction to the Mathematics and Geometry of Cad/Cam, Geometric Modeling, Scientific Visualization, and Other Cg Applications*. Industrial Press.

NEYRET, F. 1995. A general and multiscale model for volumetric textures. In *Graphics Interface '95*, Canadian Human-Computer Communications Society, W. A. Davis and P. Prusinkiewicz, Eds., Canadian Information Processing Society, 83–91. ISBN 0-9695338-4-5.

NEYRET, F. 1996. Synthesizing verdant landscapes using volumetric textures. In *Eurographics Rendering Workshop 1996*, Springer Wein, New York City, NY, X. Pueyo and P. Schröder, Eds., Eurographics, 215–224. ISBN 3-211-82883-4.

NO, H. M. B., SANDER, P. V., MCMILLAN, L., GORTLER, S., AND HOPPE, H. 2003. Geometry videos: a new representation for 3d animations. In *SCA '03: Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, 136–146.

OR, Y. H., AND ZUCKER, S. 1989. Texture fields and texture flows: Sensitivity to differences. *Spatial Vision 4*, 2–3, 131–139.

PAGET, R., AND LONGSTAFF, D. 1997. Texture synthesis via a noncausal nonparametric multiscale Markov random field. *IEEE Transactions on Image Processing 7*, 6 (June), 925–931.

PAQUETTE, E., POULIN, P., AND DRETTAKIS, G. 2002. The simulation of paint cracking and peeling. In *Graphics Interface 2002*, 59–68.

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

PELLACINI, F., FERWERDA, J. A., AND GREENBERG, D. P. 2000. Toward a psychophysically-based light reflection model for image synthesis. In *Proceedings of the 27th annual conference on Computer Graphics and Interactive Techniques*, ACM Press/Addison-Wesley Publishing Co., 55–64.

PERLIN, K., AND HOFFERT, E. 1989. Hypertexture. In *Computer Graphics (Proceedings of ACM SIGGRAPH 89)*, ACM, vol. 23, 253–262.

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

POPAT, K., AND PICARD, R. 1993. Novel cluster-based probability model for texture synthesis, classification, and compression. In *Proceedings of SPIE Visual Communications and Image Processing*, 756–768.

PORTILLA, J., AND SIMONCELLI, E. P. 2000. A parametric texture model based on joint statistics of complex wavelet coefficients. *International Journal of Computer Vision 40*, 1, 49–71.

RUSHMEIER, H., TAUBIN, G., AND GUÉZIEC, A. 1997. Applying shape from lighting variation to bump map capture. In *Eurographics Rendering Workshop 1997*, Springer Wein, New York City, NY, J. Dorsey and P. Slusallek, Eds., Eurographics, 35–44. ISBN 3-211-83001-4.

RUSHMEIER, H. E., ROGOWITZ, B. E., AND PIATKO, C. 2000. Perceptual issues in substituting texture for geometry. In *Human Vision and Electronic Imaging V*, vol. 3959, 372–383.

SAITO, H., AND KANADE, T. 1999. Shape reconstruction in projective grid space from large number of images. In *Proceedings of Computer Vision and Pattern Recognition*, 49–54.

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.

SCHOPPER, J. R. 1975. Determination of particle size and shape distribution by automatic feature analysis. In *Proceedings of the Fourth International Congress for Stereology*, 185–188.

SCHPOK, J., SIMONS, J., EBERT, D. S., AND HANSEN, C. 2003. A real-time cloud modeling, rendering, and animation system. In *SCA '03: Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer animation*, Eurographics Association, 160–166.

SEITZ, S. M., AND DYER, C. R. 1999. Photorealistic scene reconstruction by voxel coloring. *Int. J. Comput. Vision 35*, 2, 151–173.

SIMONCELLI, E. P., AND PORTILLA, J. 1998. Texture characterization via joint statistics of wavelet coefficient magnitudes. In *Proc 5th IEEE Int'l Conf on Image Processing, Chicago, Illinois*.

SULLIVAN, S., AND PONCE, J. 1998. Automatic model construction and pose estimation from photographs using triangular splines. *IEEE Transactions on Pattern Analysis and Machine Intelligence 20*, 10, 1091–1097.

SUMNER, R. W. 2001. *Pattern formation in lichen*. Master's thesis, Massachusetts Institute of Technology.

SZELISKI, R. 1993. Rapid octree construction from image sequences. *CVGIP: Image Underst. 58*, 1, 23–32.

TARINI, M., HORMANN, K., CIGNONI, P., AND MONTANI, C. 2004. PolyCube-Maps. *ACM Trans. Graph. 23*, 3, 853–860.

TAYLOR, C. C. 1983. A new method for unfolding sphere size distributions. *Journal of Microscopy 132*, 1 (October), 55–66.

TURK, G. 1991. Generating textures on arbitrary surfaces using reaction-diffusion. In *Proceedings of the 18th annual conference on Computer Graphics and Interactive Techniques*, ACM Press, 289–298.

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.

VASILESCU, M. A. O., AND TERZOPOULOS, D. 2004. Tensortextures. In *Proceedings of ACM SIGGRAPH 2004*, Computer Graphics Proceedings, Annual Conference Series, ACM, 336–342.

VERTH, J. M. V., AND BISHOP, L. M. 2004. *Essential Mathematics for Games and Interactive Applications : A Programmer's Guide*. Morgan Kaufmann.

WATSON, B., FRIEDMAN, A., AND MCGAFFEY, A. 2000. Using naming time to evaluate quality predictors for model simplification. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM Press, 113–120.

WATSON, B., FRIEDMAN, A., AND MCGAFFEY, A. 2001. Measuring and predicting visual fidelity. In *Proceedings of the 28th annual conference on Computer Graphics and Interactive Techniques*, ACM Press, 213–220.

WEI, L.-Y., AND LEVOY, M. 2000. Fast texture synthesis using tree-structured vector quantization. In *Proceedings of the 27th annual conference on Computer Graphics and Interactive Techniques*, ACM Press/Addison-Wesley Publishing Co., 479–488.

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.

WEISSTEIN, E. W. 2003. *CRC Concise Encyclopedia of Mathematics, Second Edition*. Chapman & Hall/CRC, Boca Raton, FL.

WEISTRAND, O. 2001. Shape approximation of starshaped discrete objects. In *SCIA01*, O–M4B.

WICKSELL, S. D. 1925. The corpuscle problem: A mathematical study of a biometric problem. *Biometrika 7*, 1/2, 84–99.

WITKIN, A., AND KASS, M. 1991. Reaction-diffusion textures. In *Proceedings of the 18th annual conference on Computer Graphics and Interactive Techniques*, ACM Press, 299–308.

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

WOO, M., DAVIS, AND SHERIDAN, M. B. 1999. *OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 1.2.* Addison-Wesley Longman Publishing Co., Inc.

WORLEY, S. 1996. A cellular texture basis function. In *Proceedings of the 23rd annual conference on Computer Graphics and Interactive Techniques*, ACM Press, 291–294.

ZHANG, J., ZHOU, K., VELHO, L., GUO, B., AND SHUM, H.-Y. 2003. Synthesis of progressively-variant textures on arbitrary surfaces. *ACM Trans. Graph. 22*, 3, 295–302.

ZHU, S. C., WU, Y., AND MUMFORD, D. 1998. Filters, random fields and maximum entropy (FRAME): Towards a unified theory for texture modeling. *Int. J. Comput. Vision 27*, 2, 107–126.

ZIGELMAN, G., KIMMEL, R., AND KIRYATI, N. 2002. Texture mapping using surface flattening via multidimensional scaling. *IEEE Transactions on Visualization and Computer Graphics 8*, 2, 198–207.