Evaluation of Methods for Approximating Shapes Used to Synthesize 3D Solid Textures

ROBERT JAGNOW
MIT
JULIE DORSEY and HOLLY RUSHMEIER
Yale University

In modern computer graphics applications, textures play an important role in conveying the appearance of real-world materials. But while surface appearance can often be effectively captured with a photograph, it is difficult to use example imagery to synthesize fully three-dimensional (3D) solid textures that are perceptually similar to their inputs. Specifically, this research focuses on human perception of 3D solid textures composed of aggregate particles in a binding matrix. Holding constant an established algorithm for approximating particle *distributions*, we examine the problem of estimating particle *shape*. We consider four methods for approximating plausible particle shapes—including two methods of our own contribution. We compare the performance of these methods under a variety of input conditions using automated, perceptually motivated metrics, as well as a psychophysical experiment. In the course of assessing the relative performance of the four algorithms, we also evaluate the reliability of the automated metrics in predicting the results of the experiment.

Categories and Subject Descriptors: I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling—*Geometric algorithms, languages, and systems*; H.1.2 [**Models and Principles**]: User/Machine Systems—*Human factors*

General Terms: Algorithms, Human Factors

Additional Key Words and Phrases: Shape estimation, shape perception, texture synthesis, solid textures, volumetric textures

ACM Reference Format:

Jagnow, R., Dorsey, J., and Rushmeier, H. 2007. Evaluation of Methods for Approximating Shapes Used to Synthesize 3D Solid Textures. ACM Trans. Appl. Percpt. 4, 4, Article 24 (January 2008), 27 pages. DOI = 10.1145/1278760.1278765 http://doi.acm.org/10.1145/1278760.1278765

1. INTRODUCTION

One of the most common objectives in computer graphics is to replicate the appearance of real-world objects in a simulated digital environment. In practice, this frequently involves the use of *texture maps*—images that are mapped to the surface of synthetic objects to convey the appearance of spatial variation within a material. For many objects, 2D images are sufficient; however, for objects formed from materials with inherent 3D structure, it can be difficult to find a 2D image mapping that adequately represents the material variation. As an alternative to 2D image maps, the use of 3D solid textures has proven successful for modeling materials with internal 3D structure [Gardner 1984; Peachey 1985;

Author's addresses: Robert Jagnow, MIT, Combridge, Massachusetts 03139; Julie Dorsey and Holly Rushmeier, Yale University, New Haven, Connecticut 06520; email: Holly.rushmeier@yale.edu.

(a) (b)

Fig. 1. Frequently a 3D solid texture can only be sampled as a 2D image, such as shown in (a). To synthesize objects made of this material a 3D structure, such as shown in (b), needs to be generated. There is no unique solution for the 3D structure associated with a 2D image.

Perlin 1985]. Three-dimensional solid textures are volumes that contain a material description at each point in space.

In this work, we consider the class of aggregate materials composed of discrete particles in a binding matrix. A distinguishing feature of objects made of such materials is that the shapes of the cross sections of the discrete particles are clearly visible. Common examples of such materials include concrete, asphalt, terrazzo, and granite. For these material types, an explicit volumetric representation can convey a compelling material appearance. For instance, Figure 1 shows a 2D image of material (a) that was used to extrapolate a plausible 3D structure (b). In this case, we generated the resulting object by splitting the material along its aggregate boundaries, thus exposing the internal structure. The general problem of reconstructing a 3D volume from a 2D image is heavily unconstrained with no unique solution. Furthermore, the solution cannot be compared point-by-point to some ground truth, so we must instead use appearance as a mechanism to assess the effectiveness of various computational methods.

In prior research, we studied forming 3D structures when the particle shapes are known *a priori*. For this case, we can use a mathematical relationship from stereology for recovering a unique 3D particle *distribution* given only a 2D image of a planar sample of the material [Jagnow et al. 2004]. In the original 2D image, we compute the distribution of cross-sectional areas. Using the mathematical relationship between 2D and 3D distributions and the known particle shape, we can generate a 3D volume of particles. New 2D images can be generated from this 3D volume. We were able to demonstrate that new 2D slices through this volume produced the same distribution of cross-sectional areas as the original sample image.

In this paper, we instead focus on approximating plausible *shapes* for individual particles, a problem that does not have unique mathematical solution. The difficulty in estimating a plausible shape from 2D images obtained from blocks of material is illustrated in Figure 2. The top of Figure 2 shows a simpler problem of estimating the shape of a particle by taking multiple slices. If the spacing and orientation of the slices are controlled, the sequence of circular profiles shown at the right would be obtained. With

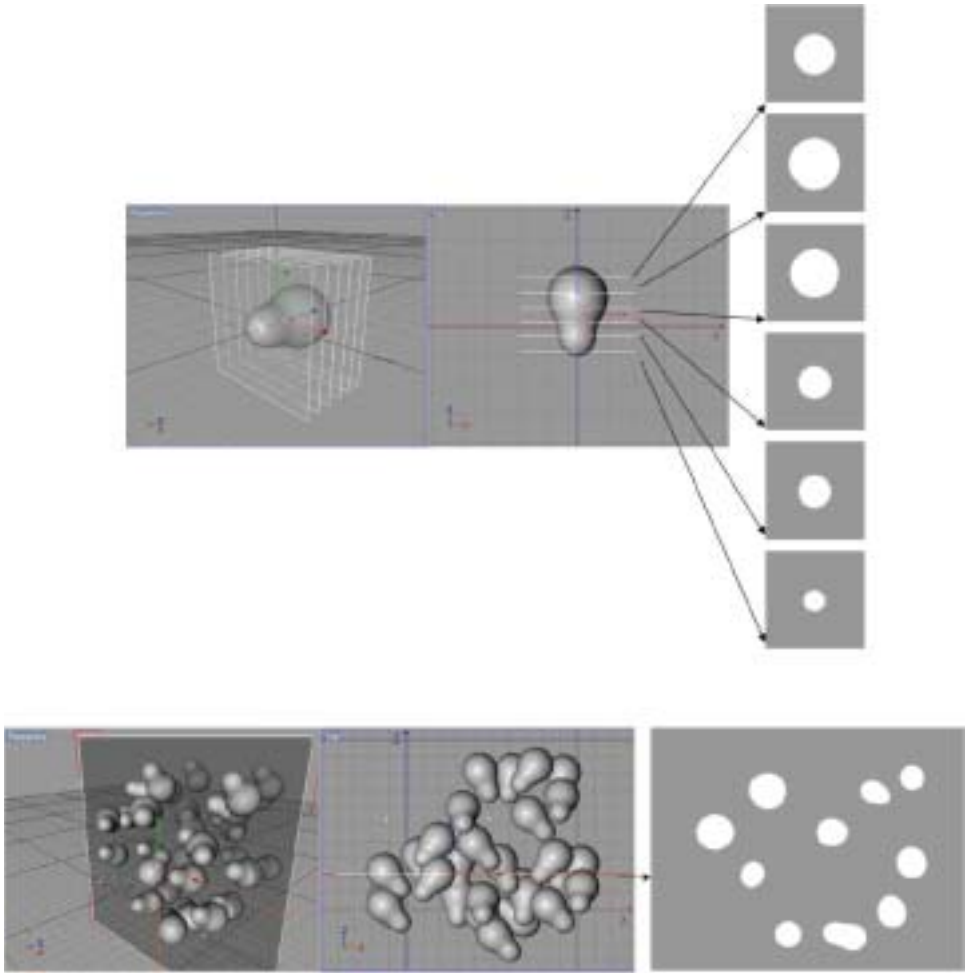


Fig. 2. The difficulty of estimating 3D shape from 2D images of slices depends on how much is known about the sampling process. The examples at top and bottom show a perspective view (left), top view (center), and section view (right) of planes slicing shapes. In the example at the top, the location and orientation of slices through a simple shape are known, generating profiles on the right that are relatively easy to link together to estimate 3D shape. In the example at the bottom, a single slice is obtained through many instances of the shape. Since the orientation and position of the shapes relative to the slice plane are unknown, it is not as clear how to connect the resulting 2D profiles to estimate the shape.

known slice planes, the points on all of the profiles can be located in a single 3D coordinate system as a "point cloud." By simply stitching together the profiles, we could estimate particle shape. More sophisticated algorithms for interpolating point clouds exist with geometric guarantees for the quality of the approximation of the original shape [Amenta et al. 2001].

The bottom of Figure 2 shows the more complicated problem of estimating the shape of particles in an aggregate material. We obtain a single slice, rather than a series of slices. The single slice passes through many instances of the particle and so produces many profiles in the single image. The particles are arbitrarily oriented and positioned relative to the slice plane, so we cannot easily stitch them together to make a shape estimate, in this case. Analogous to the distribution problem, we would like to find a relationship between the 2D shape of these profiles and the 3D shape. Unlike the distribution

problem, there is no unique mathematical relationship that we can use. Also analogous to the distribution problem, we would like to be able to demonstrate that the shapes of the cross sections produced from the estimated 3D shape are the same as the shapes in the original sample. Unlike the distribution problem, there is no simple metric that we can rely on to measure if the 2D shapes are the same.

In this paper, we consider different algorithms for estimating the 3D particle shape from a 2D sample. In the absence of an established metric for comparing the similarity of the 2D cross sections derived from our 3D estimates to the 2D cross sections in the original sample, we evaluate our results from a psychophysical standpoint with perceptual experiments. Our objectives in this research are twofold. Our first and primary goal is to use direct observer evaluations to determine which algorithms are most effective for generating 3D particle shapes that exhibit 2D profiles with a comparable appearance to the input data. Our second objective is to assess how reliably simple numerical metrics serve as predictors of the psychophysical evaluation. If simple metrics can be found to predict the psychophysical evaluation, they could be used in further algorithm development in place of observer evaluations.

While the specific contribution of this work is the description and evaluation of a particular method for texture synthesis, at a more general level our contribution is the description of a problem in computer graphics that can *only* be addressed with perceptual principles and experiments. Unlike much recent work in applying perceptual principles to computer graphics rendering and modeling, we are not seeking an approximation to an image, model or animation that is perceived as identical to gold standard. For example, in modeling a shape, there are metrics from outside of psychophysics (mininum distance between surfaces) that can be used to assess how close an approximation of the shape is to a gold standard exact shape. It is possible to show that one shape approximation is more accurate than another using purely numerical metrics without psychophysics. Observer experiments are performed to relate the numerical metrics to their perceptual importance and refine the use of numerical metrics [Watson et al. 2001]. In another example, simulating visible light transfer, there are numerical metrics (difference in radiance estimated for a pixel) that can be used to assess how close an approximation of a lighting solution is to a gold standard exact solution. It is possible to show that one method for computing light transfer is better than another by showing that the radiances calculated are closer to the gold standard. Observer experiments are performed to relate numerical accuracy of radiance to its perceptual importance and refine the requirements for accuracy of light transfer for image synthesis [Martens and Myszkowski 1998]. In the case of generating solid textures, we are not attempting to approximate pixel by pixel identical images or identical shapes. We do not have an accepted numerical metric outside of psychophysics that can guide algorithms and allow us to determine that method A is better than method B. For solid texture synthesis we need psychophysical experiments and metrics both to evaluate algorithms and to establish numerical metrics that can be used in future algorithm development. In general, computer graphics algorithms for simulating the appearance of materials fall into this class.

Recently, there has been considerable work on texture synthesis for computer graphics [Turk 2001; Wei 2003; Ghazanfarpour and Dischler 1996; Dischler and Ghazanfarpour 1999; Wei and Levoy 2001; Zhang et al. 2003; Lefebvre and Poulin 2000]. Despite the fact that these computer graphics techniques do not produce textures that are identical to previous texture synthesis techniques, they have not been tested with psychophysical metrics or experiments. Texture synthesis techniques in computer graphics are evaluated with respect to computational efficiency and casual visual inspection. While texture similarity has been studied in great depth in the vision literature, texture synthesis applied to 3D objects for the generation of realistic images has not. A great deal of work in understanding perception of textures in the vision literature has concentrated on understanding textures in the abstract without context in order to understand basic visual mechanisms. The graphics texture synthesis problem differs in that whether textures are successful depends on context. If an image of fur is used as an example

input, the texture generated should look like fur, and should look like natural fur when applied to a 3D model of a dog or cat. More relevant to the work presented here, if an image of an aggregate material is used as example input, the texture generated should look like an aggregate. An object should look like it was shaped from the solid when the texture is applied, rather than looking like it was papered over with prints of an aggregate texture. To achieve these results, texture synthesis techniques in computer graphics often model physical processes, such as cutting through a 3D solid, rather than relying on purely artificial processes, such as adding and modifying signals of varying spatial frequency. The work presented here is a small step in the perceptual evaluation of texture synthesis as part of the simulation of material appearance for realistic 3D modeling.

Our primary goal in this paper is evaluating methods for estimating the 3D shapes of particles used in solid texture synthesis. In our experiments, we are interested in finding methods that are clearly superior or inferior. As a consequence, we have different requirements in our experimental methods as compared to prior work investigating fundamentals of human vision. For methods that clearly fail, we do not need to find precisely how poorly they perform by conducting extensive experiments. Furthermore, we can only test a finite number of cases for a type of input data for which there is no definitive test set that covers all possible problems. In the end, our results indicate trends, not precise measures of the quality of output for each algorithm for any possible input.

In the next section, we review relevant previous work in texture mapping, texture synthesis, shape estimation, and shape evaluation. We follow this with a description of four candidate methods for estimating particle shapes from their 2D profiles—spherical harmonics, constructive solid geometry, generalized cylinders, and morphed cylinders. Next, we describe the evaluation of the results of these methods using simple numerical metrics (total curvature and area to perimeter squared ratio) and by a study with human observers. We found that the simple numerical metrics were not successful in predicting the results of the observer study. In the observer study, we found that two of the methods, spherical harmonics and constructive solid geometry, always produced shapes that were easily distinguishable from shapes produced from a synthetic ground-truth model for some classes of particles. The other two methods were not significantly different from each other in their success when the results of all particle classes were combined. The morphed-cylinder method produced the greatest percentage of successful results when the floor of average results minus standard deviation is separately considered for each class of particle. We found that even the morphed cylinder method, however, did not produce results comparable to sampling the original synthetic ground truth volume, indicating that there is room for improvement in the shape estimation problem.

2. PREVIOUS WORK

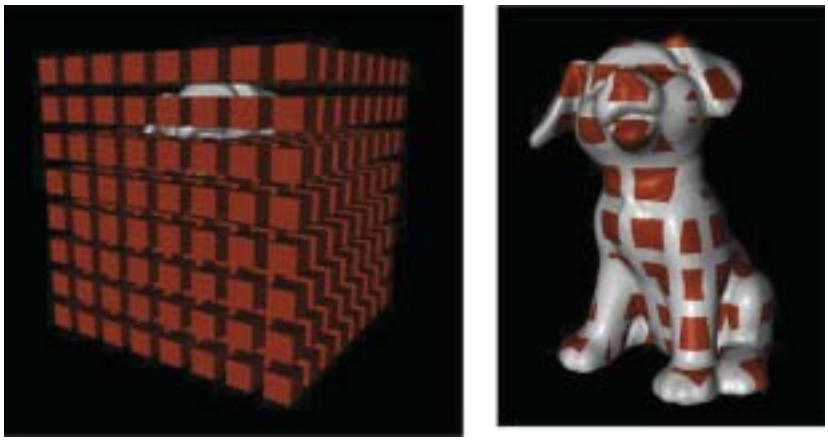
This research builds on a variety of related work in the areas of computer graphics, computer vision, and *stereology*—the study of 3D structure from 2D samples developed in the fields of biology and material science [Hagwood 1990; Underwood 1970]. The most relevant research is described below.

2.1 2D versus 3D Texture Mapping

Texture mapping is used to add spatial variation to the appearance of a surface. The variation may be solely in color, but can also include variations in reflectance (e.g., diffuse versus specular). In Figure 3, we illustrate the 2D texture mapping process. Figure 3a shows an image of a sample pattern of brown squares on a white background, and a sample 3D object—a small dog. As shown in a, the image is mapped to the surface by identifying corresponding points in the 2D image and on the 3D object. Many possible correspondence schemes are possible and b shows some possible results for mapping this 2D image on the dog. The results in b would be consistent with stretching some thin material over the surface of the dog.

(a) (b)

Fig. 3. In 2D texture mapping, an image is assigned to a 3D object (a). The mapping is performed by establishing correspondences between locations on the 2D image and the 3D object. Many possible correspondences are possible, giving significantly different results (b).



(a) (b)

Fig. 4. In 3D solid texture mapping a 3D structure is defined, (a), as a set of brown cubes encased in a white binder (binder not shown for clarity). The object is immersed in the structure and texture is assigned in (b) by intersecting the object with the structure.

A 3D solid texture map is a numerically defined volume in which every point in space has a material value defined either by interpolating between discrete values defined at lattice points, by a function of 3D position, or by packing sets of small geometric primitives into the volume (as illustrated in Figure 4). An extensive discussion of defining and representing 3D solid textures can be found in Ebert et al. [1994].

Fig. 5. 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.

In Figure 4, we illustrate the 3D solid texturing process. The brown and white image shown in Figure 3 could be interpreted as a slice of a material composed of an array of brown cubes embedded in a white binding material. In Figure 4a, we show the dog model embedded in the array of brown cubes with the white binding material hidden to expose the internal structure. The surface of the dog is then colored by finding the intersections of the dog with the array of cubes to produce b. Finding a 2D mapping that would produce the same result would be extremely difficult without an intermediate 3D representation.

In addition to properly applying the 2D image to curved shapes to avoid obvious distortions, a further problem with using 2D images to texture map even simple 3D objects is consistency, as illustrated in Figure 5. The cube at the left is mapped with 2D images of an aggregate and the cube at the right is mapped with 2D images of wood grain. Because of inconsistencies at the texture boundaries, the cubes appear to be wallpapered with the images rather than being carved out of solid materials. If the left cube were composed of a 3D aggregate, we would expect the images of particles at cube edges to be continued on adjoining faces. If the right cube were composed of wood, we would expect the images to demonstrate a consistent grain orientation.

Both 2D and 3D texture mapping are standard features in current modeling/rendering systems. To use these techniques though, the textures must be synthesized in some way to completely cover the object being designed.

2.2 Texture Synthesis

In a computer graphics modeling system, both 2D and 3D solid textures can be created by purely synthetic means. A user can specify functional parameters to procedurally generate texture of arbitrary spatial extent. Alternatively, textures can be synthesized from 2D photographs of real materials. For 2D textures, the major problem is generating textures of arbitrary size from example images. Substantial research over the past 10 years has resulted in numerous 2D texture synthesis algorithms to address this problem. Most of these algorithms fall into the categories either of methods that work by matching image statistics, e.g., Heeger and Bergen [1995] and De Bonet [1997] or by copying pixels or patches selected by probability sampling the example image, e.g., Efros and Leung [1999] and Efros and Freeman [2001]. Two-dimensional texture synthesis algorithms have been extended to apply them to the surfaces of geometric models [Turk 2001; Gorla et al. 2001] and to apply synthesis to 2D maps representing bidirectional textures [Tong et al. 2002]. These are only a few examples of recently developed techniques. A comprehensive review of 2D texture synthesis algorithms is outside the scope of this

paper. We note, however, that no perceptual evaluation of the large number of methods for synthesizing 2D textures on 3D shapes has been performed. The closest work is that of Kim et al. [2004], studying the effect of texture on the perception of hape. That work does not consider how well a synthesized texture on a 3D surface conveys the appearance of a corresponding 2D input.

Methods for 2D texture synthesis can readily be applied to 3D volumes, if data for 3D structure are available for the material to be modeled. However, unlike 2D image samples, which can be captured with any camera, sample 3D solid texture images must be obtained by imaging closely spaced slices through a real-world object or by reconstructed 3D volumes from measurements of x ray transmissions. Such 3D samples are not available for common materials to be used in computer graphics modeling systems. Instead, synthesis techniques need to be devised, which use only a 2D image of the 3D solid texture.

One common class of natural and man-made materials consists of discrete particles distributed in a binding medium, such as concrete, terrazzo, asphalt, or granite. An example image of one such material is shown in Figure 1a. A single 2D slice through an aggregate material can hint at the embedded particle shapes and their distribution, but the exact structure is unknown unless additional information is available *a priori*. For example, for the image used in Figure 3, the corresponding structure could have been long brown sticks rather than cubes, brown flakes, or some completely unrelated structure covered by a veneer.

In comparing the simple array of cubes in Figure 4 to the real aggregate imaged in Figure 1, we can observe that we need to estimate four things to synthesize a particular 3D structure:

1. The individual particle shapes
2. The spatial distribution of particles
3. The size distribution of particles
4. The color variations of particles and binder

In prior work [Jagnow et al. 2004], we considered estimating (2–4) given a 2D slice of the material and estimates of the particle shapes. For a given particle shape, the statistics of the distribution of particles in a 2D slice are determined by the 3D particle distribution. This relationship is mathematically invertible using techniques from stereology so that the statistics of the 3D distribution can be computed from the analysis of the 2D slices. We demonstrated that after particle size distributions are estimated, the particles can be spatially distributed using simulated annealing, and previous 2D texture synthesis algorithms can be adapted to produce color variations.

In this prior work, particle shapes were estimated by hand—a tedious and error-prone process. To have a completely automatic method, we need a reliable means to estimate the 3D particle shapes from the 2D slices observed in the sample images. Unlike the size distribution problem, there is not a mathematically unique solution for computing the 3D shape from the 2D slices.

2.3 Particle Shape Estimation

The challenge of reconstructing 3D shape from 2D images is a common problem within both the graphics and computer vision communities. A number of specialized solutions have been presented, each targeted to particular data constraints. Many methods rely on 2D images of the 3D shape captured by a camera, rather than on scanning 2D slices taken through the shape. In the case where camera locations are known, Martin and Aggarwal [1983] developed a computational method for approximating the bounding volume of an object by intersecting extruded silhouettes of the object as seen from various viewpoints. The resulting shape is known as the *visual hull*. Matusik et al. [2000] adapted this approach to operate on dynamic scenes. In contrast, for our data, we typically do not have the convenience of a known correspondence between individual 2D profiles.

When only a single profile is available, Igarashi et al. [1999] constrain the reconstruction problem by assuming that the 3D shape exhibits a circular profile along its chordal axis. However, this approach is suitable only for producing smooth, rounded surfaces.

Within the computer vision community, the generalized cone and generalized cylinder have long been recognized as effective primitives for shape representation because of their ability to express a wide variety of shapes with few parameters [Binford 1971; Marr and Nishihara 1978]. 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.

Particle shape estimation is also of great interest in the stereological community. In *histology*—the study of microscopic structures of plants or animals—the analysis of particle distribution and shape can be used to determine the presence of cancerous or precancerous cells [Keiding and Jensen 1972]. The spherical-deformation model is one of the most commonly used methods within the stereological community for estimating 3D shapes [Hobolth and Jensen 2002; Hobolth 2003]. However, this approach is often unsuitable for representing particles with sharp corners or distinct features.

Solid texture synthesis for aggregate materials was considered from a stereological perspective in our previous work [Jagnow et al. 2004]. In this prior work, particle shapes were approximated by manually editing nonuniform rational B-spline (NURBS) surfaces. We seek to accelerate this process by applying automated methods for extrapolating particle shapes.

2.4 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. We cannot expect to reconstruct the identical 3D shapes that were present in the original material. A more appropriate objective is to approximate a particle shape that exhibits perceptually similar profiles to those seen in the input. To evaluate our methods, we need some way of quantifying the perceptual similarity of the results of various shape approximation algorithms.

To compare the similarity of profiles, a wide variety of metrics are available. For an extensive overview of such techniques, we refer the reader to the summary by Loncaric [1998]. Unfortunately, few existing 2D shape comparisons metrics are perceptually motivated. In an attempt to establish a relationship between computational shape similarity and human shape similarity, Scassellati et al. [1994] consider the use of algebraic moments, spline curve distances, cumulative turning angle, sign of curvature, and Hausdorff distance—all with limited success. The shape metrics that we consider here 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, we 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*, because of the fact that the metric yields a maximal value for circular shapes.

Simple metrics cannot be expected to reliably capture all psychophysical aspects of profile similarity, so direct human evaluation also plays an essential 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. [2000, 2001], who rate the performance of automated metrics for predicting experimental results. The perceptual experiments described in their research measure *naming time*—the elapsed time from when

Table I. Summary of Input Parameters and Data Assumptions for Shape Approximation Methods

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

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 [1998] presented technique for evaluating the performance of perceptually motivated appearance metrics. In their psychophysical experiments, users were asked to numerically rank the visual similarity of image pairs. Unlike this previous work however, the images we create from estimated 3D solid textures are not expected to be identical. We cannot use image visible difference predictors, such as the model validated by Martens and Myszkowski.

Recently, Ledda et al. [2005] evaluated the performance of tone-mapping operators relative to viewing a high dynamic range image. This is similar to our task, in that this evaluation can only be performed by perceptual experiment. In their work, they have subjects observe three images at a time and perform paired comparisons. We use a similar approach for evaluating the quality of results of different particle estimation methods. However, unlike the work of Ledda et al., we do not present comparisons that record a direct preference between two estimation methods.

3. PARTICLE SHAPE ESTIMATION

Our goal is to use 2D shape profiles, such as those illustrated on the right of the lower series of images in Figure 2, and estimate a 3D shape. Our 3D shape estimation will be judged successful if 2D slices, through a volume populated by instances of the shape, produce 2D profiles that are similar to the profiles in the original sample. To estimate 3D shapes from the 2D profiles, we draw on previous methods discussed in Section 2.3 and develop alternative techniques extending previous work.

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. Without any *a priori* knowledge about the particles, we need to rely on limited assumptions about their shape.

The *spherical harmonics* and *generalized cylinder* models described below were introduced in prior publications; the *constructive solid geometry* and *morphed generalized cylinder* methods are our own contribution to the problem. A detailed comparison of all four algorithms is discussed in Section 4.

For all of the methods considered here, we 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 I. For the spherical harmonics method, only the frequency information of the 2D curve is considered (i.e., the coeffiecients of a Fourier expansion of the curve expressed as a radius vector function), 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 here, three curves are always used.

The constructive solid geometry algorithm takes exactly three profiles as input. In this case, we assume that 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 one-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 here 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. Overall, we would expect that methods that use the greatest amount of data from the original sample would produce the best results. We would also expect that more sophisticated method used to interpolate or extrapolate the input would produce better results. We need experiments, however, to determine whether the number of profiles used has a significant impact, and whether more sophisticated methods improve results or produce unforseen noticeable artifacts.

3.1 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,

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

Similarly, 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.

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]. 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:

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

Likewise, the Fourier–Legendre coefficients a_n^m are assumed to exhibit a Gaussian distribution with mean 0 and variance λ_n. Hobolth and Jensen demonstrate that the coefficients a_1^m can be ignored, and the remaining variance values are related by the expression

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

where P_n^m are the associated Legendre functions of the first kind.

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 unique particles can be generated with the recovered variance values.

This approach for particle modeling is mathematically rigorous but is poorly suited for particles that exhibit sharp corners or distinct features. Coefficients of the Fourier–Legendre expansion are assumed to be mutually independent. This tends to eliminate the presence of sharp angles or other features that may characterize the input. Furthermore, Hobolth [2003] uses only ten expansion terms to approximate each shape, noting that the variance of the terms becomes excessively noisy after that point. For our results, we use 30 expansion terms, but still fail to capture the sharp features of the input profiles.

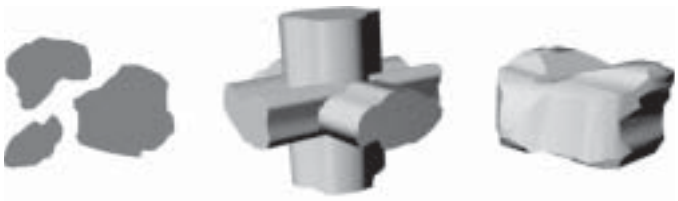
One inherent drawback of any spherical deformation method is the requirement that all resulting particles must be star-shaped. This constraint becomes increasingly problematic for long, slender particles with anisotropic shape characteristics.

3.2 Constructive Solid Geometry

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; Matusik et al. 2000]. 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, we select three characteristic profiles, which we assume are silhouettes of a single target particle shape, projected orthographically onto three orthogonally oriented viewing planes. We 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 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 the 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},



(a) (b) (c)

Fig. 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.

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, we 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, we 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 6b. The CSG intersection 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.

3.3 Generalized Cylinder Model

Dischler and Ghazanfarpour [1999] model macroscopic particles using a generalized cylinder method. 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 one-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. To enable the algorithm to operate on a larger class of geometric inputs, we 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 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

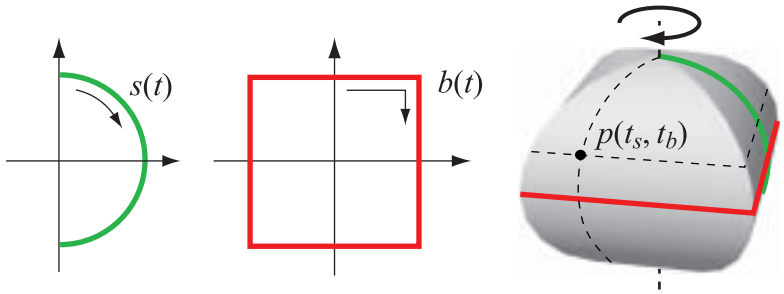


Fig. 7. Generalized cylinder method for particle construction. The construction curves shown on the left are used to create the particle shape on the right.

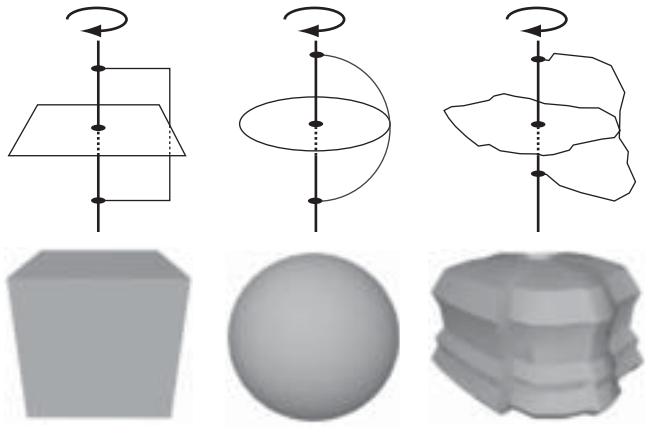


Fig. 8. Example shapes created by the generalized cylinder method. The construction curves shown in the top row result in the particle shapes in the bottom row.

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 ensure that curve s remains tangent to b.

Figure 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.

3.4 Morphed Generalized Cylinder Model

The final particle generation algorithm that we consider is a novel approach introduced here. 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.

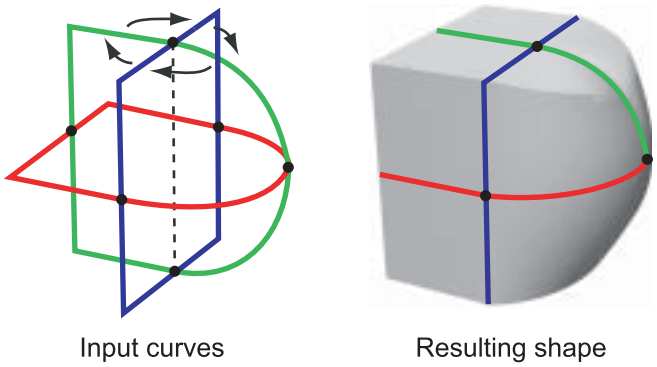


Fig. 9. Morphed generalized cylinder method for particle construction. The base curve (red) and morph curves (green and blue) result in the shape shown at the right.

To begin, the input profiles are reoriented and sorted as indicated in Section 3.2. As described in the previous section, the base profile b—shown in red in Figure 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 research, 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 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, we choose to meet the constraints by cutting the morph profiles at each point where we expect two profile curves to meet. We then scale each region of the profile independently.

To ensure that each morph curve lies tangent to the base curve, morph profiles are scaled inward or outward from the axis of rotation. Next, we scale the curves along the axis of rotation to ensure 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. We 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

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

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

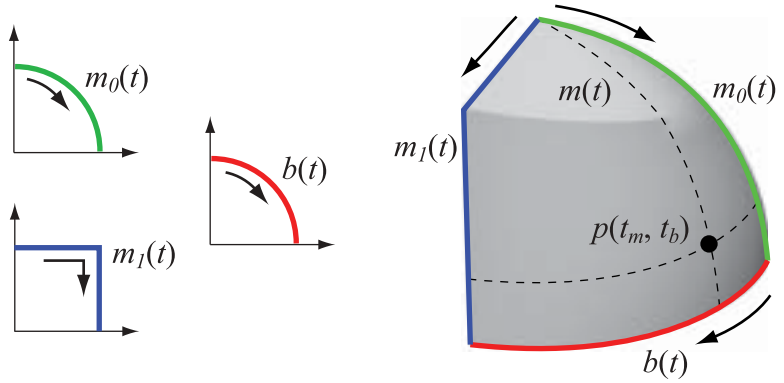


Fig. 10. Construction process for the morphed generalized cylinder method. The construction curves shown on the left define the portion of the particle shape shown on the right.

shown in Figure 10. We 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

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

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 Eq. (1) is necessary for enforcing this condition. An example shape generated by this process is shown on the right side of Figure 9.

4. ALGORITHM COMPARISON

All of the methods described in the previous section produce 3D shapes from 2D profiles and could be used to approximate a 3D solid texture. We developed the CSG and morphed generalized cylinder techniques in the belief that they would better approximate a wide range of shapes. To evaluate the relative performance of each of the four proposed algorithms, we 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. Our hypothesis was that simple

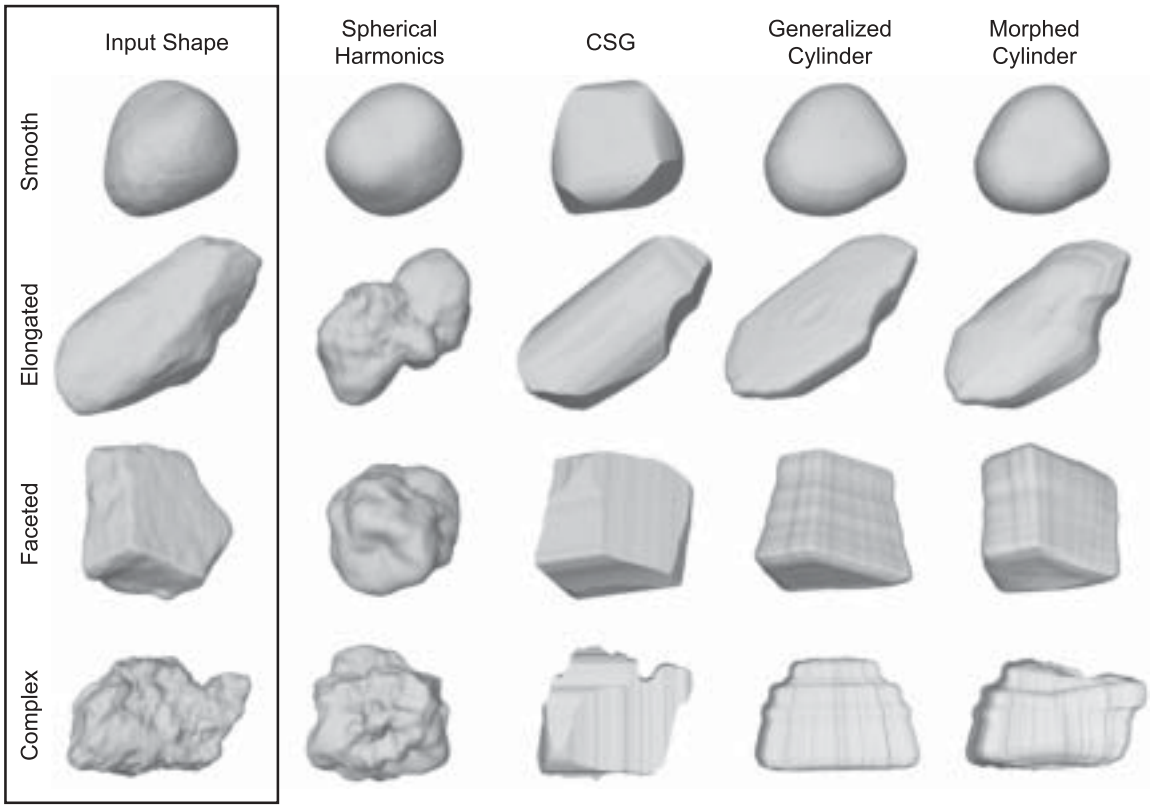


Fig. 11. In the left column are the four scanned particle shapes used as the inputs to the shape approximation algorithms. To the right are the decoy particle shapes created by each of the four particle approximation methods.

2D shape metrics would be adequate to evaluate the relative performance of the various algorithms. If the automated shape metrics were successful, they would be easy to use in the evaluation of further refinements of our shape-estimation algorithms.

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 nonconvex 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 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 12.

4.1 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,

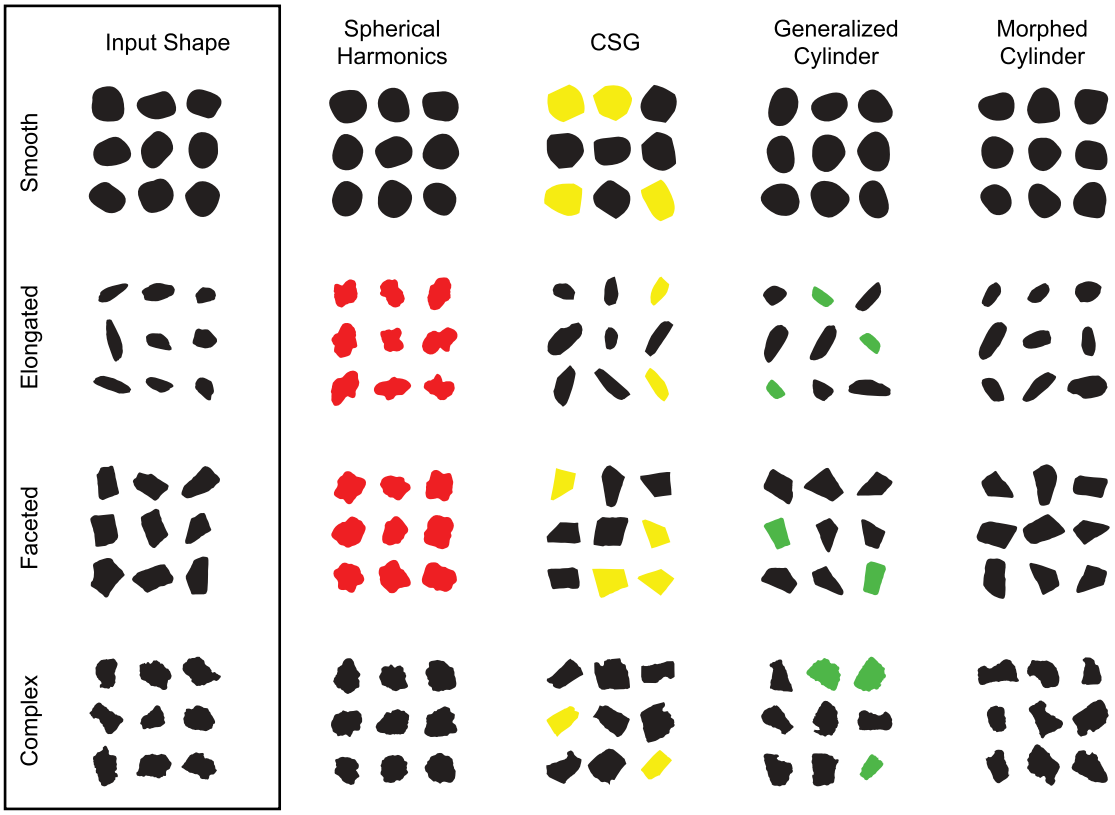


Fig. 12. This table shows randomly selected central profiles for each original and synthesized particle shape. Profiles that we judged to be poorly matched to the original data are highlighted in color. Red profiles are overly curved or blobby, yellow profiles exhibit excessively sharp corners, and green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

we 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 we 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. 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 13, we created a utility that sliced through each particle 10,000 times at uniformly distributed random orientations. We 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.

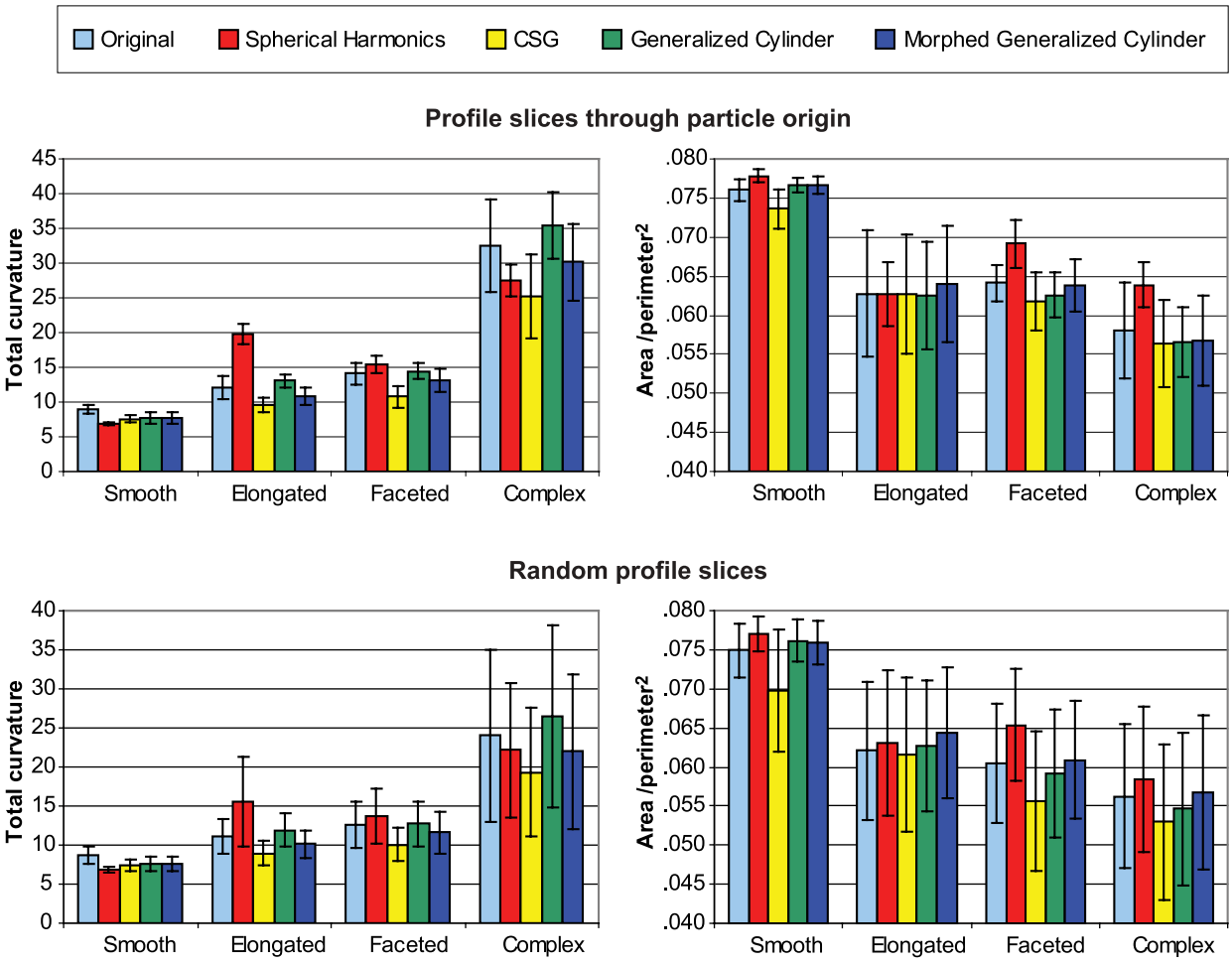


Fig. 13. Original and synthetic particle shapes are compared via two measures of 2D visual complexity: total curvature and area/perimeter^2. These charts show the mean and standard deviation values obtained from 10,000 random slices through each particle shape. The top row shows values obtained from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

In the second row, the values were obtained by slicing particles at both a random orientation 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 1a. As expected, these profiles exhibit a greater diversity of appearances, resulting in higher standard deviation values.

To analyze the significance of these results, we apply analysis of variance (ANOVA). For all of the datasets considered, we set a threshold of significance at $p \leq 0.01$. Our first concern is whether the measures we used were sensitive to the different particle shapes. Second, we wish to see if they reveal differences in the particle estimation methods. For each of the four charts in Figure 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 ($p < 0.001$).

Table II. Average Deviation from Original Data Values for Each Algorithm as Measured for Each of the Two Metrics and Profile Slicing Techniques

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%

Table II 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 attempt 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 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. Our original hypothesis that the simple automated shape metrics would be adequate to evaluate the proposed algorithms was contradicted by our own casual observation. To obtain more definitive results, we constructed an experiment to obtain more formal input from human observers, as described in the following section.

4.2 User Study

To obtain a direct evaluation of the four proposed algorithms for shape approximation, users were shown a series of images in a computer-based test in which they were asked to evaluate shape similarity. Since any two particles are clearly going to have different shapes, we constructed a test where the users would judge the similarity of clusters of 2D profiles. We authored a user study in which participants attempt to distinguish between original and synthetic 2D slices obtained from 3D volumes of particles.

4.2.1 *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. Further, we chose to use particles of uniform size rendered as solid black and white images to eliminate the effects of variations in particle size distribution and color estimation.

The study used the same four particle shapes described previously and shown in Figure 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. (The geometric nature of the complex particles does not allow them to be packaged as tightly, so fewer of the complex particles of the constant size could be packed into the test volume.) For each shape, five synthetic volumes were generated for each particle type—one for the original particle and one each for each of the shape estimation methods. Images were formed for use in the study by taking random 2D slices through these synthetic volumes.

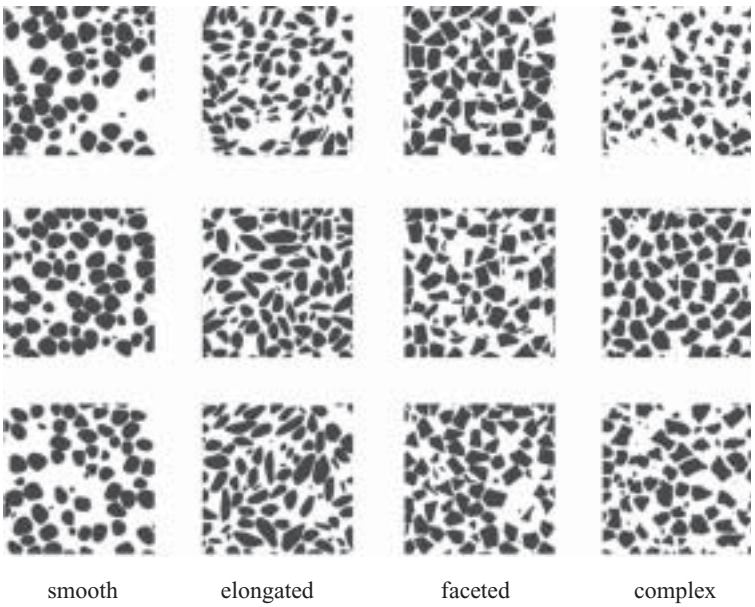


Fig. 14. Examples of the variation in appearance evident in the original distributions. All three of the images in each column are taken from the same 3D distribution of particles.

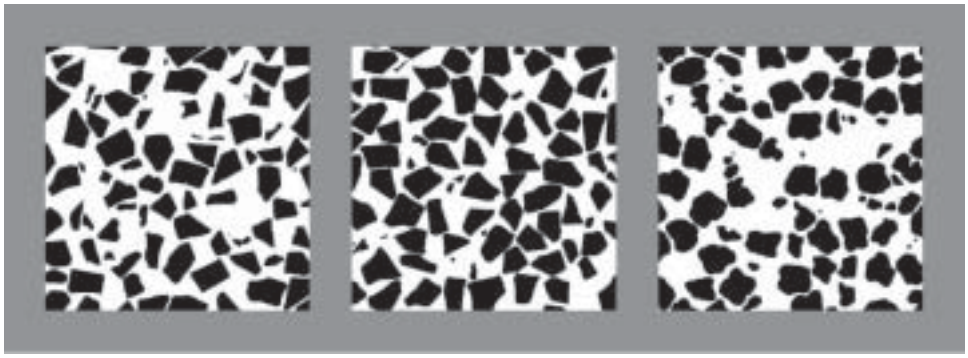


Fig. 15. Example screen from user study. Users are asked to select the image that is least similar to the image in the center. Here, a decoy image is shown on the far right.

Figure 14 shows the variation in appearance of 2D slices taken through the volumes containing the original particles.

4.2.2 *Procedure.* To evaluate the effectiveness of the methods, study participants were given a computer-based test to compare image similarity. Similar to the approach used by Gurnsey and Fleet [2001], users were shown a series of screens, each containing three images in a row, as depicted in Figure 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

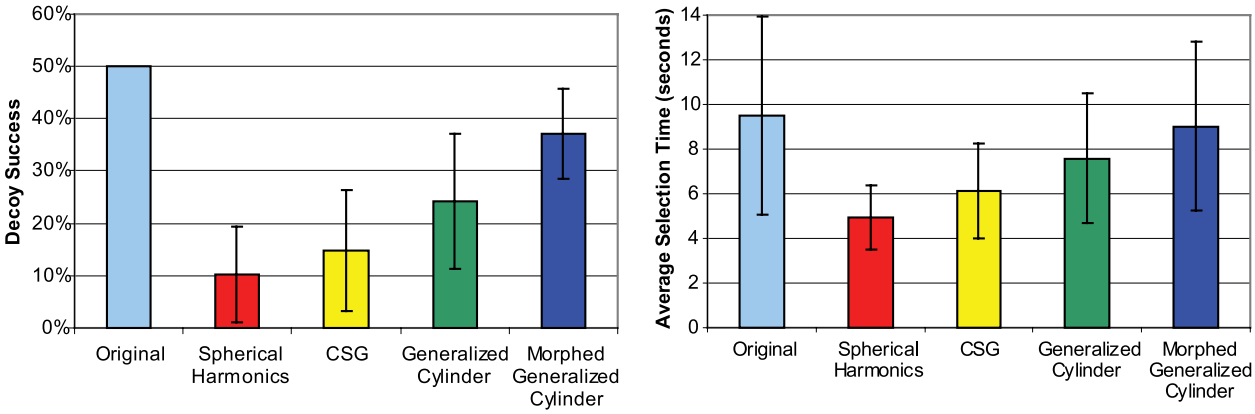


Fig. 16. User study results showing the relative performance of four different algorithms for extrapolating particle shape based on limited shape information. The graph on the left shows the frequency with which users were "fooled" by a decoy model. Average selection times are shown on the right. Mean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

test interface measured the accuracy and time required to make each selection. Note that unlike the Ledda experiment described earlier, the observer is not selecting a preference between two estimation methods, but is always making a choice between an image generated by an estimation method and an image generated from the same test volume as the center reference image.

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 250×250 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 55 cm from the computer monitor. Each image was shown at a size of 7.8 cm, forming a visual angle of 8.1°.

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. All study participants are members of the computer graphics and computer vision communities at MIT; none had detailed 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 1 s.

4.2.3 *Results.* Sixteen users participated in the study, yielding 1280 individual data points. The results of the study are summarized in Figure 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.

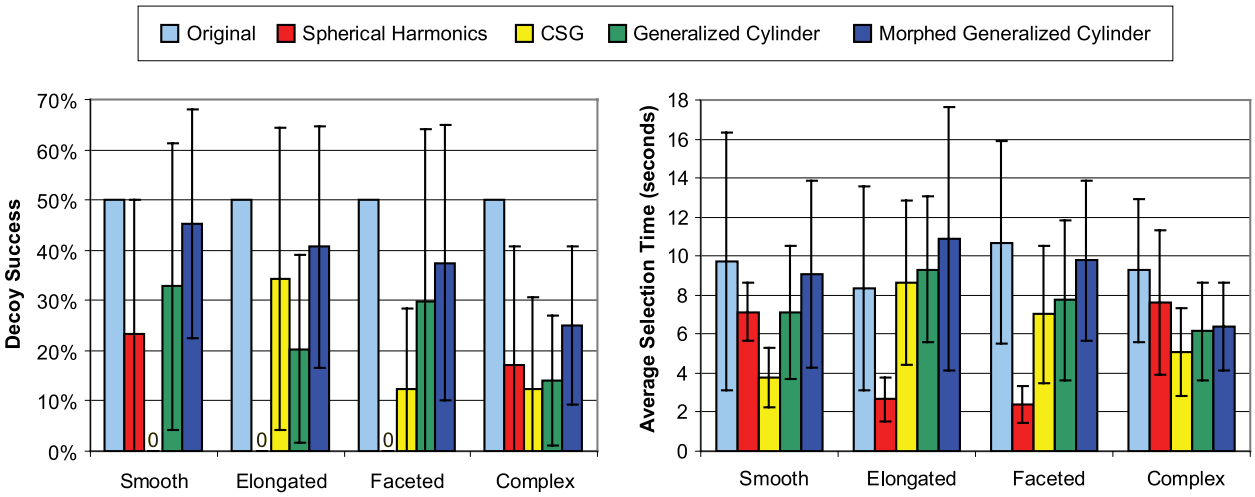


Fig. 17. Detailed graph of user study results showing variation between input datasets. The graph on the left shows the frequency with which users were "fooled" by a decoy model. Average selection times are shown on the right. Mean and standard deviation values are shown for both graphs.

These values indicate an ordering of algorithm effectiveness, ranking the morphed generalized cylinder method clearly higher than the constructive solid geometry and spherical harmonic methods. 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 17, 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, we 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 17.

As shown in Figure 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, ANOVA indicates a significant interaction between the algorithm and shape ($p < 0.001$).

5. DISCUSSION

Considering all datasets combined, the morphed generalized cylinder algorithm generates significantly more successful decoys than the spherical harmonics or CSG method. While the overall success rate is higher for the morphed generalized cylinder than the generalized cylinder, the difference is not statistically significant in our data. The desired 50% success rate that would be obtained by chance is outside the range of plus or minus a standard deviation for even the morphed generalized

cylinder method. Overall this indicates that there is room for improvement in estimating particle shape.

The detailed analysis by particle class indicates that the CSG and spherical harmonics algorithms are completely unacceptable for some particle classes. The spherical harmonics algorithm is acceptable for replicating the appearance of smooth surfaces that lack sharp features, but always failed for elongated shapes and faceted particles. In contrast, CSG provides a reasonable approximation for elongated particles or surfaces with sharp edges, but always failed for smooth shapes.

The detailed results by particle class do not show a significant difference in the generalized and morphed generalized cylinder methods. In choosing between them though it should be noted that the average minus standard deviation in the success rate is always highest for the morphed generalized cylinder method.

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 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 17, it should be remembered that 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, but this does not diminish the statistical significance of the findings. For both graphs, ANOVA demonstrates a significant dependence on the algorithm, indicating that spending effort on improving the shape estimate algorithm is worthwhile. Even by the most optimistic estimate, none of the algorithms can produce particle shapes that are indistinguishable from the originals, suggesting that time would be better invested in improving the two more successful approaches rather than running more experiments to refine the ordering of the methods.

The relatively poorer performance of the spherical harmonics and CSG methods produced by the user study should come as no surprise, based on the input data for each approach and how that information is used. The spherical harmonics method considers only particle curvature while ignoring the first-order shape statistics of the input profiles. The CSG modeling approach incorporates three profiles directly into a resulting shape, but does not make an effort to interpolate smoothly between the shapes.

The lack of greater difference in the performance between the other two methods is somewhat surprising. The generalized cylinder performs smooth shape interpolation, but only considers two profiles as input. The morphed generalized cylinder performs smooth shape interpolation and incorporates three input profiles directly into the resulting 3D particle shape. In general, we would expect the results to favor those algorithms that incorporate more information from the original data set using an intelligent interpolation algorithm. In this case, we did not observe as strong an effect as we had expected.

Comparing the user study and the automated metrics, the automated metrics do not successfully capture the perceptual similarity of shapes to the degree needed to evaluate the four estimation methods. Each of the metrics 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. The significant shortcoming of the automated metrics is that they do not reveal the failure of some of the methods to ever adequately simulate some particles shapes—i.e., the zero success rate of the CSG method to simulate smooth shapes or the spherical harmonic methods to simulate elongated or faceted shapes.

The automated metrics presented here were clearly 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. 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.

We believe it would be particularly 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, we have had difficulty in deriving such a metric, as the methods that we have considered are sensitive to polygon tessellation and coincident or near-coincident vertices.

The metrics described here may be considered as coarse predictors for shape similarity, but these automated techniques should not be treated as equivalent replacements for direct user evaluation. Further improvements in estimating shape for solid texture synthesis will need to continue to rely on psychophysical experiments rather than on existing simple shape metrics.

6. CONCLUSIONS AND FUTURE WORK

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

The automated techniques considered here are marginally successful for predicting algorithm performance from a psychophysical standpoint. Methods that better capture sharp features on numerically defined shapes are needed. Even modest contributions in this area could 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.

Although this research was primarily focused on the problems of shape perception and texture synthesis, the algorithms described here might also be applied to other problems within computer vision and computer graphics. Since profiles offer a compact representation of a shape, some of these algorithms may be considered for geometry compression. Future research may also consider the use of the morphed generalized cylinder algorithm as a general modeling approach for creating 3D geometry from simple, limited user input.

The work presented here is an example of a class of computer graphics problems that can only be approached using perceptual principles and experiments. Three-dimensional solid texture modeling is one aspect of the broader area of materials modeling. Real materials exhibit highly detailed spatial variations because of weathering and aging effects that we cannot expect to perfectly replicate in simulation. Physically accurate spatial variations of texture as a function of object geometry are expensive to simulate or capture. Rather than the informal evaluations used for material modeling today, perceptual methods will play an increasingly important role in the future in developing efficient texture synthesis methods. Additional techniques are needed for evaluating similarity on a higher level rather than evaluating whether two images are simply identical.

ACKNOWLEDGMENTS

We would like to thank Maxwell Planck for his contributions to this research as an undergraduate intern at MIT. We also thank Prof. Steve Zucker for his advice in the design of the user study.

REFERENCES

AMENTA, N., CHOI, S., AND KOLLURI, R. K. 2001. The power crust. In *SMA '01: Proceedings of the Sixth ACM Symposium on Solid Modeling and Applications*. ACM Press, New York. 249–266.

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.

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

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., New York, Reading, MA. 361–368.

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

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 (Oct.), 438–441.

EBERT, D. S., MUSGRAVE, F. K., PEACHEY, D., PERLIN, K., AND WORLEY, S. 1994. *Texturing and Modeling: A Procedural Approach*. Academic Press Professional, New York.

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, New York. 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, Los Alamitos, CA. 1033.

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

GARDNER, R. J., HOBOLTH, A., JENSEN, E. B. V., AND SØRENSEN, F. B. 2005. Shape discrimination by total curvature, with a view to cancer diagnostics. *Journal of Microscopy 217*, 1, 49–59.

GHAZANFARPOUR, D. AND DISCHLER, J.-M. 1996. Generation of 3D texture using multiple 2D models analysis. *Computer Graphics Forum, Proceedings of Eurographics 1996 15*, 3, 311–324.

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

GURNSEY, R. AND FLEET, D. J. 2001. Texture space. *Vision Research 41*, 3, 745–757.

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

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

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

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

IGARASHI, T., MATSUOKA, S., AND TANAKA, H. 1999. Teddy: A sketching interface for 3D freeform design. *Proceedings of SIGGRAPH 99*, Los Angeles, CA. 409–416.

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. New York. 329–335.

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 (Sept.), 813–829.

KIM, S., HAGH-SHENAS, H., AND INTERRANTE, V. 2004. Conveying three-dimensional shape with texture. In *APGV '04: Proceedings of the 1st Symposium on Applied Perception in Graphics and Visualization*. ACM Press, New York. 119–122.

LEDDA, P., CHALMERS, A., TROSCIANKO, T., AND SEETZEN, H. 2005. Evaluation of tone mapping operators using a high dynamic range display. *ACM Trans. Graph. 24*, 3, 640–648.

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

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 (Oct.), 2584–2593.

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

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., New York, Reading, MA. 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*.

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.

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, New York.

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

PEACHEY, D. R. 1985. Solid texturing of complex surfaces. In *Computer Graphics (Proceedings of ACM SIGGRAPH 85)*. ACM, New York. 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., New York, Reading, MA. 55–64.

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

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.

SCASSELLATI, B., ALEXOPOULOS, S., AND FLICKNER, M. 1994. Retrieving images by 2D shape: A comparison of computation methods with human perceptual judgments. In *Storage and Retrieval for Image and Video Databases (SPIE)*. Vol. 2185. 2–14.

TONG, X., ZHANG, J., LIU, L., WANG, X., GUO, B., AND SHUM, H.-Y. 2002. Synthesis of bidirectional texture functions on arbitrary surfaces. In *SIGGRAPH '02: Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*. ACM Press, New York. 665–672.

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

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

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

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, New York. 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, New York. 213–220.

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

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. New York. 355–360.

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

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.

Received December 2005; revised May 2006 and July 2006; accepted September 2006