Acquisition, Compression, and Synthesis of Bidirectional Texture Functions

Melissa L. Koudelka
Electrical Engineering
Yale University
New Haven, Connecticut

Sebastian Magda
Computer Science
University of Illinois Urbana-Champaign
Urbana, IL

Peter N. Belhumeur
Computer Science
Columbia University
New York, New York

David J. Kriegman
Computer Science
UC San Diego
San Diego, CA

***Abstract*— Real world surfaces such as tree bark, moss, sponge, and fur often have complicated geometry that leads to effects such as self-shadowing, masking, specularity, and interreflection as the lighting or viewpoint in a scene changes. We use image based techniques to analyze and represent bidirectional texture functions, or BTFs, with correct geometric and lighting effects. A basis for the apparent BRDF of points on the surface is determined and used to compress the texture datasets, as well as provide a space for comparison of texture elements across all lights and views. The compression method reduces the approximately 10,000 images in each 6-D lighting, viewpoint, and spatial variation texture dataset to under 2 MB.**

I. Introduction

Recently, a reasonable amount of effort has been placed on producing realistic 3-D textures[1] such as flora, fabric, bark, hair and fur, skin, etc. Although some advances have been made, these textures have proven difficult to represent compactly and render under variable viewpoint and lighting conditions because they exhibit complex reflectance properties and intricate small scale geometric structure. Consider the images shown in Fig. 1, noting how differently the sponge appears as the viewpoint (top row) and illumination direction (bottom row) vary.

Fig. 1. Sample images of kitchen sponge texture with viewpoint variation, top, and lighting direction variation, bottom. Each image has been rectified to a frontal view.

In this paper, we present a set of methods that we believe are a significant advance in the study of 3-D textures under lighting and viewpoint variation. Our main contributions are as follows:

- Acquisition of densely sampled 6-D texture datasets – with 2-D variation in lighting, 2-D in viewpoint, and 2-D in the spatial dimensions of the image – with approximately 50 times the number of images captured in previous work [2].
- Determination of a tailored basis for the apparent BRDF of surface points on the texture that can be used to compress the 2.5GB raw texture datasets to under 2 MB - a compression ratio of more than 1000:1.
- Demonstration of the utility of coefficients on the apparent BRDF basis as a criterion for matching texture features, allowing for synthesis of textures that can be rendered under any viewing and lighting condition, while maintaining a consistent mesostructure as those conditions change.

Our BTF compression and synthesis methods are designed in keeping with the following two observations. First, real-world textures (in particular natural textures) have a geometric and reflectance complexity that is difficult to capture with a small number of sample images or a simple parametric model, and so require relatively fine sampling of their appearance to capture the full variation with lighting and viewpoint. Second, real-world textures often exhibit a randomness that is not well modeled by a limited set of distinct texture elements.

An exhaustive list of related work is not possible in a short paper such as this, so we will limit comments to only the most relevant prior work. Dana et al. [2] introduced the Bidirectional Texture Function (BTF) to account for complex variation in both reflectance and geometry-induced effects such as self-shadowing and masking. The BTF is analogous to the more familiar Bidirectional Reflectance Distribution Function (BRDF), [10], although it also includes spatial variation, making it a six dimensional function of viewpoint, lighting, and texture image coordinates. In [2], a groundbreaking database (CUReT) of 205 images of 61 materials was reported, and these textures were directly mapped onto objects by appropriate indexing into the BTF function. Our datasets are captured in the same vein, but have much greater sampling density.

Our analysis of the texture datasets makes no assumptions

[1]In this work, we consider texture to be the mesostructure – the fine scale yet still visible geometric structure – present on the surface of an object as discussed in [6]. Examples of mesostructure include the ridges of bark on a tree, the dimples on a golf ball, and the plant structure of moss.

about the underlying reflectance function of the surface, or the geometric structure of the texture. This is in contrast to polynomial texture maps [9], which assume that the surface has smoothly varying reflectance in compactly representing and rendering textures with lighting variation only. In [8], a low order parametric model for reflectance (e.g., Phong [12]) is assumed in recovering the texture's surface geometry for use in rendering additional images in a 6-D dataset. This limits the textures that can be represented to those with well-behaved surface geometry and reflectance. In [14], 6-D textures are discretized to a set of representative texture elements for analysis, thus assuming that textures are made up of a limited number of such elements. The complete BTF is needed for synthesis in [14], whereas our method requires only a small number of coefficient textures and basis vectors to render a surface under any viewing conditions.

BTF datasets with 6-D variation are inherently large and require compression to make them tractable for distribution and use. In [16], 3-D surface light field datasets that express surface variation across multiple viewpoint images were compressed using modifications of both principal component analysis (PCA) and vector quantization (VQ). Notable improvement in compressing the data was also achieved using the BRDF reparameterization in [13]. Similar algorithms using SVD and tensor product expansions were presented in [11] and [5] respectively. Interactions between pixels were modeled in [17] to capture changes in viewpoint. In the aforementioned algorithms, only viewpoint is addressed while lighting is fixed. PTMs [9] can be considered as compressing textures with lighting variation for fixed viewpoint, with the compression achieved largely by assuming smoothly varying reflectance at each surface point. In [7], 3-D textons (i.e., feature vectors computed for each pixel) were used to compress textures when a limited number of representative texture elements are sufficient to characterize the entire texture. A statistical representation for BTF histograms is derived in [1] for compression and recognition. However, this method does not maintain the spatial element of the BTF and so is not useful for synthesis.

The remainder of this paper is organized as follows: Section II describes our data acquisition method. Section III outlines our apparent BRDF representation and associated texture compression method, and provides analysis of the compression results. Finally, Section IV describes an extension to currently existing texture synthesis methods that uses our BTF representation.

II. 3-D Texture Data Acquisition

Six dimensional BTF datasets were acquired with variation in viewing and lighting angle, as well as spatial variation within the image. This was done using the rig shown in Figure 2. An Adept robot arm moved a white light emitting diode (LED) over a hemisphere above the surface of the texture sample, providing two degrees of freedom of lighting variation. The texture sample was mounted on a pan/tilt head whose axes intersect at the texture surface, providing two degrees of freedom in viewpoint. The camera, a 3-chip digital video camera (Canon XL-1) in our set-up, was fixed for all images in the dataset. Differently colored squares surround the texture sample and serve as fiducial points so that correspondence between texture positions can be found.

Each texture dataset contains about 10,000 color, 480 $\times$ 360 images – about 50 times the number of images for a single sample in the CUReT database [2], as mentioned previously. This corresponds to a sampling of 20 degrees in both viewing angles for 90 possible views, and 15 degrees in lighting for 120 possible light source angles. The lighting angle is relative to a coordinate system attached to each sample. Due to obstructions of the camera by the robot arm, small portions of the dataset are unusable. The obstructed images were removed from the datasets, and replaced with images linearly interpolated from the closest available samples. Nine distinct samples were captured - kitchen sponge, lichen, green moss, spanish moss, velvet, gravel, carpet, faux fur and lego/plastic. Given the significant effort to acquire these large datasets of images and their potential usefulness, we are making the datasets freely available to other researchers.[2]

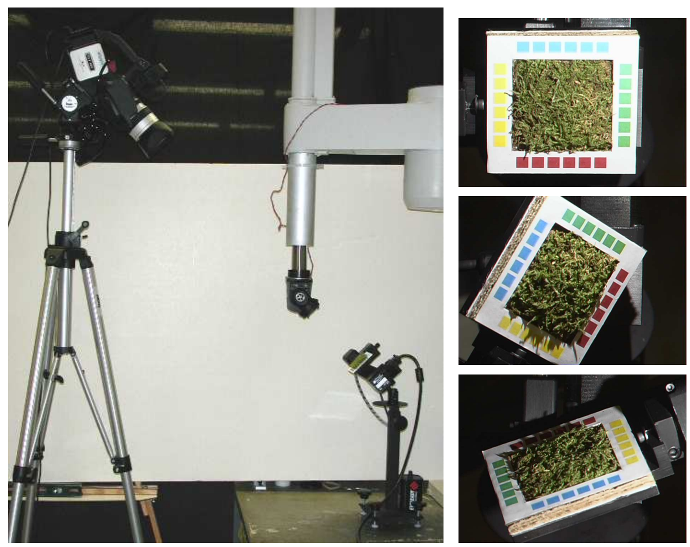


Fig. 2. Images of each texture sample were acquired using a 3-chip digital video camera while a white LED source was moved by an Adept robot arm, left. The texture sample was mounted on a pan/tilt head that provided the viewpoint variation. The texture samples were surrounded by a fiducial marker for use in rectifying the dataset. Three sample images from a raw dataset are shown.

III. Texture Compression

To address the storage and distribution issues associated with such dense BTF datasets, we present a method for texture compression. Our compression method has the additional benefit of representing the entire BTF in a low dimensional space that lends itself easily to pixel by pixel comparisons of texture elements across all captured conditions.

We first align all of the images in the dataset, establishing correspondence between texture elements. Each texture image

[2]Upon publication of this paper, the datasets will be available via our web site.

is rectified to the plane of a fronto-parallel texture using a 2-D homography. The homography is determined by marking corresponding points on the fiducial marker surrounding the texture sample in each image. The rectification makes all images the same size with texture elements at the same pixel locations. The rectification also crops the image to only the texture sample area, and in so doing reduces the dataset from the captured 2.5 GB to about 700 MB, a compression ratio of 3.5:1. This step is key, because it removes any gross geometric differences between the images – leaving primarily effects due to the mesostructure on the surface.

Now that the texture images are rectified, we can define the "apparent BRDF", $b(i,j)$, at each pixel location (i,j) as the value of that pixel across all lighting and viewing conditions in the dataset. It is not the true BRDF, since it includes shadows and interreflections caused by surface details, as well as the cosine foreshortening term. Intuitively, one can consider the apparent BRDF as the BRDF of the surface at a given point modulated by a visibility function induced by surrounding geometric structure. Since textures are repetitive in nature, we believe that there exists a low dimensional basis for these visibility functions, and in turn, for the apparent BRDF. Since the textures are rectified, only a single basis vector would be required for a planar Lambertian surface. For textured surfaces, the basis need only account for variation due to the nonplanarity of the surface (i.e., effects due to the mesostructure and reflectance changes over the surface), and so a significant compression is expected.

To compute the apparent BRDF basis, we first form a matrix M whose columns are apparent BRDF vectors. Due to memory and computational constraints, only a limited number of pixels from the original BTF are used. The pixels are chosen from a contiguous region in the image, and due to the repetitive nature of textured surfaces, should be representative of the texture as a whole.

$$M = [b(0,0) \cdots b(m,n)] \tag{1}$$

The singular value decomposition (SVD) of M is computed, and those eigenvectors corresponding to the k largest eigenvalues are kept as the apparent BRDF basis. Next, all pixels in the input BTF are projected onto the linear apparent BRDF basis, resulting in a set of k coefficients for each color channel at each pixel. Those k coefficients encapsulate all variation in lighting and viewpoint for a particular pixel (or $3 \times k$ for RGB images).

The coefficient textures can be scaled to the range [0,1] and stored as JPEG images, and the basis vectors scaled and stored as a single uncompressed image. This final step results in a compressed texture dataset of size 1.57 MB for $k = 150$ and an image size of 128 by 128. The finished size represents a compression of about 470:1 from the rectified dataset, and more than 1000:1 from the raw captured dataset. Storage size was determined to increase with $O(k)$ at approximately 11 kB for each additional vector. Figure 3 shows the percent error between original and reconstructed images versus the number of basis vectors used for a particular light and view angle. Note that the percent error levels off for greater than 150-200 basis vectors in each case. Figure 4 shows several original and reconstructed texture image pairs, along with the associated storage space.

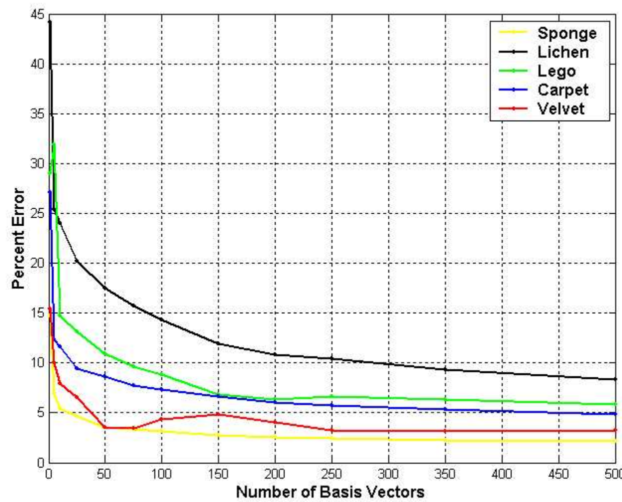


Fig. 3. Plot of percent error versus number of basis vectors used in reconstruction. Percent error is computed as the sum of the differences between original and reconstructed pixels divided by the sum of the original pixels.

IV. Texture Synthesis and Rendering

Our compressed representation for the BTF (i.e., a set of apparent BRDF basis vectors and associated coefficient textures) can be used directly to synthesize novel textures. The representation can be used with any available synthesis algorithm that explicitly compares image intensities from the input image to generate synthesized pixels. Instead of comparing image intensities as is done in the 2D algorithm, we simply compare coefficients. We have chosen to demonstrate this by extending the image quilting algorithm in [3], but could have easily extended [4] or [15] among others in a similar way. Although the current implementation is limited to square primitives (and thus simple geometries), it demonstrates the effectiveness of matching in the apparent BRDF coefficient space.

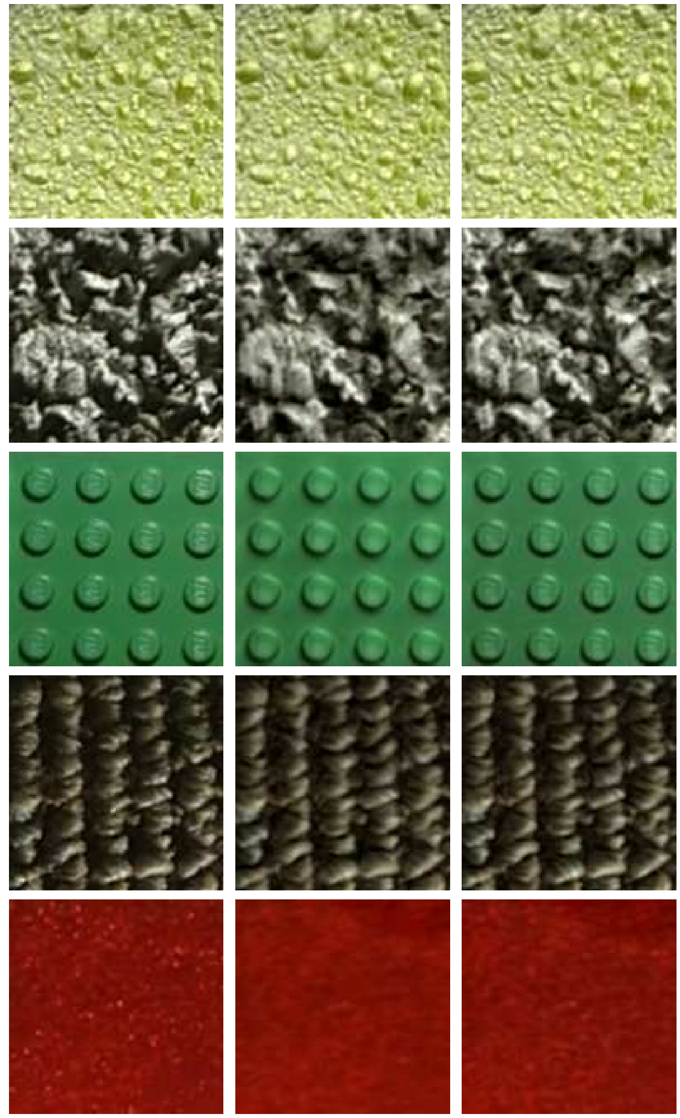
Synthesized images are generated by selecting patches based on a best match in texture coefficient space, forming a set of synthetic coefficient textures. The final image is then rendered by determining the lighting and viewing angle of the current surface point, $[\theta_v, \phi_v, \theta_l, \phi_l]$, and computing the intensity of that point, $I(i,j)$ using Equation 2.

$$I(i,j) = U([\theta_v, \phi_v, \theta_l, \phi_l]) \times c(i,j) \tag{2}$$

Each row of U corresponds to a single light and view combination, the columns of U are basis vectors, and c is the stack of k coefficient textures. Since every possible lighting condition is not covered in the dataset, $I(i,j)$ is actually computed for the three closest sources and linearly interpolated. Synthesis results for three of the textures on simple geometries are shown in Figure 5. In each case, a point light source is positioned at varying distances from the surface.

V. Discussion

We have introduced a new method for effectively representing and synthesizing textures whose appearance varies with lighting and viewpoint. Still, there are many directions to pursue. Our representation was simply integrated into existing intensity-based 2-D synthesis methods to produce the results shown here. We are currently working on implementing a near real-time algorithm that uses the apparent BRDF representation to synthesize BTFs on arbitrary surfaces. Texture mapping in general is effective for the interior of an object's projection, but fails to capture appearance near the occluding contour, especially when the texture has relief greater than a pixel. We hope to overcome this using datasets of curved samples along with "mixed pixel" representations. While our individual textures are sampled more finely than those in CUReT [2], there are fewer distinct textures in our database. With time the database will be enlarged.



(a)	(b)	(c)
Original	50 Bases	150 Bases
$\approx$ 700 MB	$\approx$ 500 kB	$\approx$ 1.57 MB

Fig. 4. Original and Reconstructed Results: Column (a) shows the original images; (b) and (c) show reconstructed images using 50 and 150 basis vectors; The approximate storage space for each representation is given. The samples are kitchen sponge, lichen, Lego^{TM}(plastic), carpet, and velvet.

References

[1] O. Cula and K. Dana, "Compact representation of bidirectional texture functions," in *Proc. IEEE Conf. on Comp. Vision and Patt. Recog.*, Kauai, Hawaii, December 2001.

[2] K. Dana, B. Van Ginneken, S. Nayar, and J. Koenderink, "Reflectance and texture of real-world surfaces," *ACM Transactions on Graphics*, vol. 18, no. 1, pp. 1–34, January 1999.

[3] A. Efros and W. Freeman, "Image quilting for texture synthesis and transfer," in *SIGGRAPH*, Los Angeles, CA, August 2001, pp. 341–346.

[4] A. Efros and T. Leung, "Texture synthesis by non-parametric sampling," in *ICCV*, September 1999.

[5] R. Furukawa, H. Kawasaki, K. Ikeuchi, and M. Sakauchi, "Appearance based object modeling using texture database: Acquisition, compression and rendering," in *Eighth Eurographics Workshop on Virtual Environments*, 2002.

[6] J. Koenderink and A. van Doorn, "Illuminance texture due to surface mesostructure," *J. Opt Soc. Am.*, vol. 13, no. 3, pp. 452–462, March 1996.

[7] T. Leung and J. Malik, "Representing and recognizing the visual appearance of materials using 3d textons," *Int. J. Computer Vision*, vol. 43, no. 1, pp. 29–44, June 2001.

[8] X. Liu, Y. Yu, and H. Shum, "Synthesizing bidirectional texture functions for real-world surfaces," in *SIGGRAPH*, Los Angeles, CA, August 2001, pp. 97–106.

[9] T. Malzbender, D. Gelb, and H. Wolters, "Polynomial texture maps," in *SIGGRAPH*, 2001.

[10] F. Nicodemus, J. Richmon, J. Hsia, I. Gisberg, and T. Limperis, "Geometric considerations and nomenclature for reflectance, NBS monograph 160," National Bureau of Standards, October 1977.

[11] K. Nishino, Y. Sato, and K. Ikeuchi, "Eigen-texture method: Appearance compression based on 3d model," in *Proc. IEEE Conf. on Comp. Vision and Patt. Recog.*, 1999, pp. 618–624.

[12] B. Phong, "Illumination for computer-generated pictures," *Comm. of the ACM*, vol. 18, no. 6, pp. 311–317, 1975.

[13] S. M. Rusinkiewicz, "A new change of variables for efficient brdf representation," in *Eurographics Workshop on Rendering*, June 1998, pp. 11–22.

[14] X. Tong, J. Zhang, L. Liu, X. Wang, B. Guo, and H. Shum, "Synthesis of bidirectional texture functions on arbitrary surfaces," in *SIGGRAPH*, 2002.

[15] L. Wei and M. Levoy, "Fast texture synthesis using tree-structured vector quantization," in *SIGGRAPH*, July 2000, pp. 355–360.

[16] D. N. Wood, D. I. Azuma, K. Aldinger, B. Curless, T. Duchamp, D. H. Salesin, and W. Stuetzle, "Surface light fields for 3d photography," *SIGGRAPH*, pp. 287–296, July 2000.

[17] A. Zalesny and L. V. Gool, "Multiview texture models," in *Proc. IEEE Conf. on Comp. Vision and Patt. Recog.*, Kauai, Hawaii, December 2001.

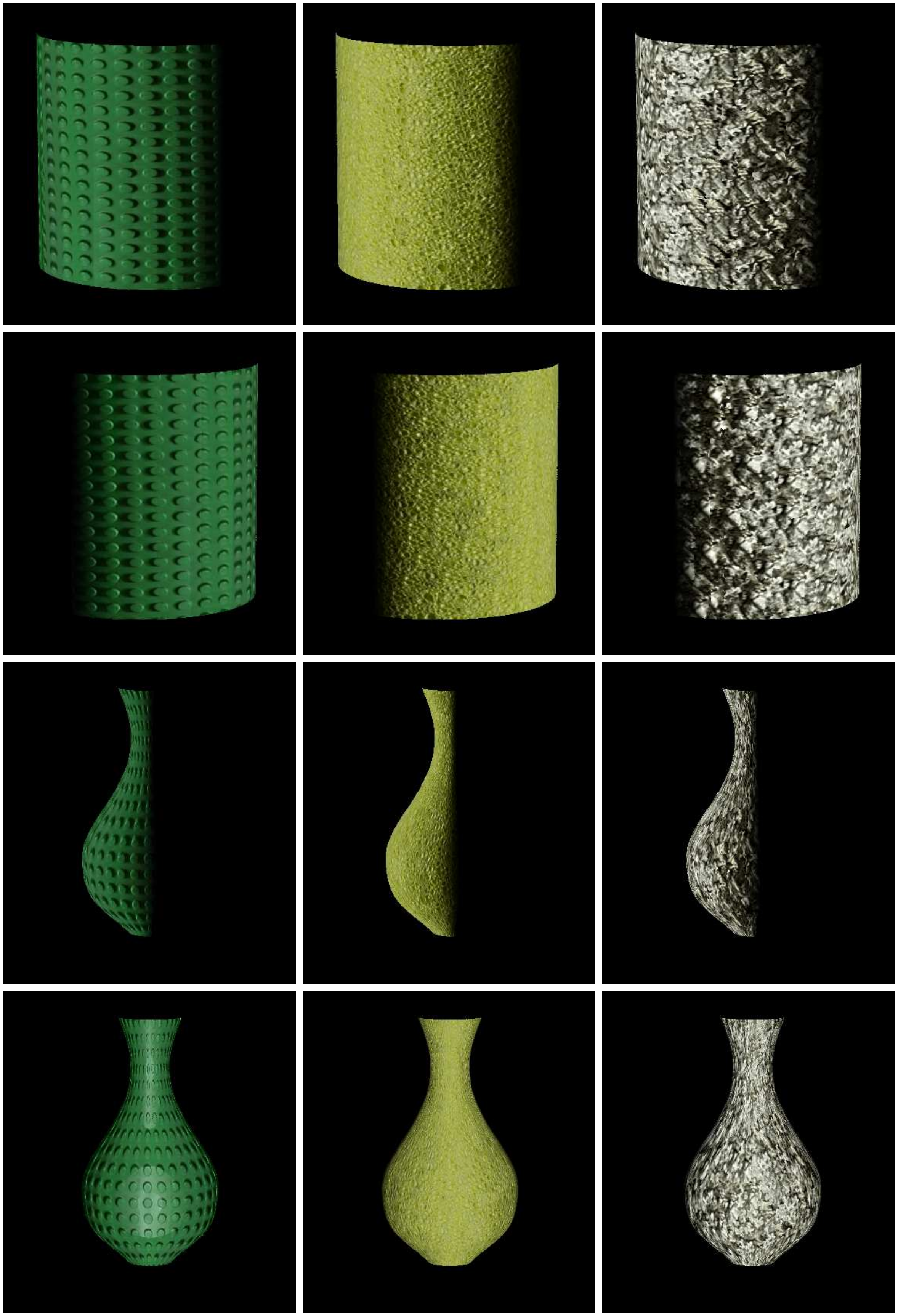


Fig. 5. Synthesized texture (Lego^{TM}, kitchen sponge, and lichen) rendered on the surface of simple geometries. In each example, a single point light source is used. Note that although the lighting conditions change, the mesostructure on the surface of the cylinder examples remains consistent for both lighting conditions.