

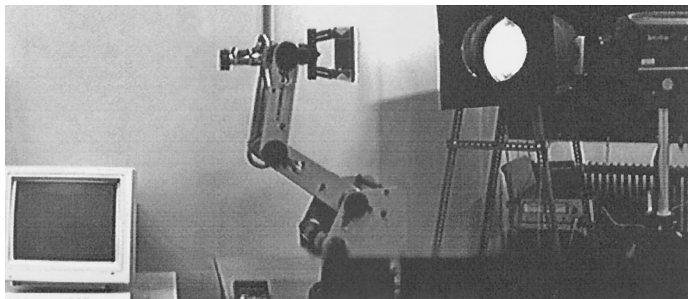
Interactive Bidirectional Texture Functions

Lecture 18

BTFs

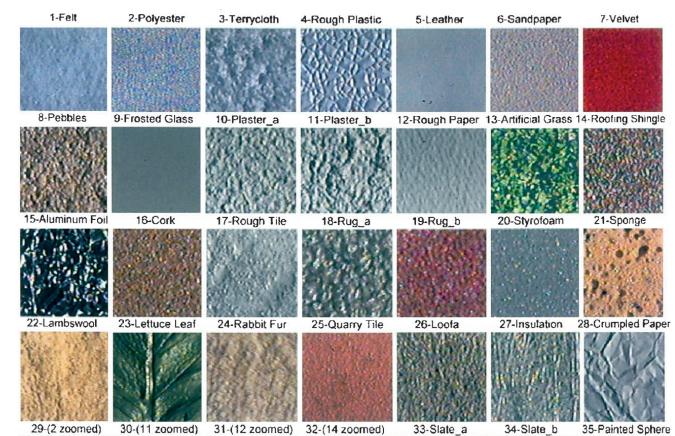
- A form of captured surface appearance
- Generalization of BRDF from single color to a texture
- Data format like SVBRDF, but philosophically distinct
- Several methods exist for interactive rendering
- Introduced by Dana et al.—Columbia/Utrecht BTF data data: <http://www.cs.columbia.edu/CAVE/software/curet/>
- Later implementations: McAllister (SVBRDF) and Uni. Bonn data: <http://sbrdf.cs.unc.edu>
data: <http://btf.cs.uni-bonn.de>

Dana et al. BTF capture system



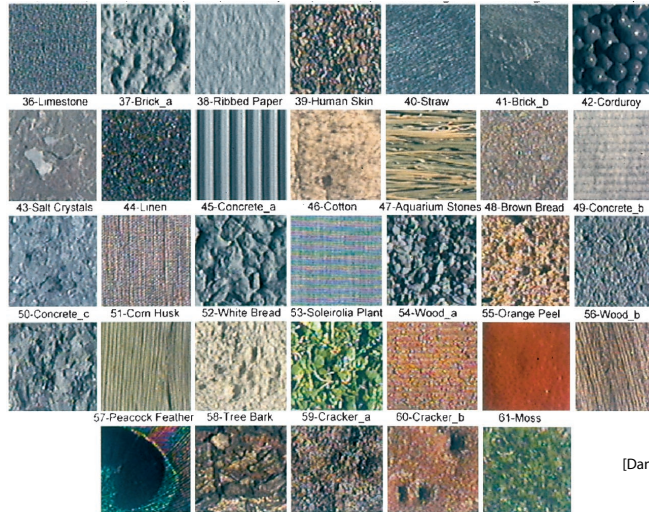
[Dana et al. 1999]

CURet BTF database



[Dana et al. 1999]

CUREt BTF database



[Dana et al. 1999]

Cornell CS569 Spring 2008

Lecture 18 • 5

Uni. Bonn BTF capture system

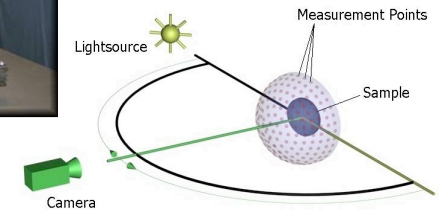


Figure 5: Capturing the BTF of planar sample using gonioreflectometer-like setup with a fixed light source, sampleholder and a moving camera.

[Müller et al. 2004]

Cornell CS569 Spring 2008

Lecture 18 • 6

Uni. Bonn BTF data

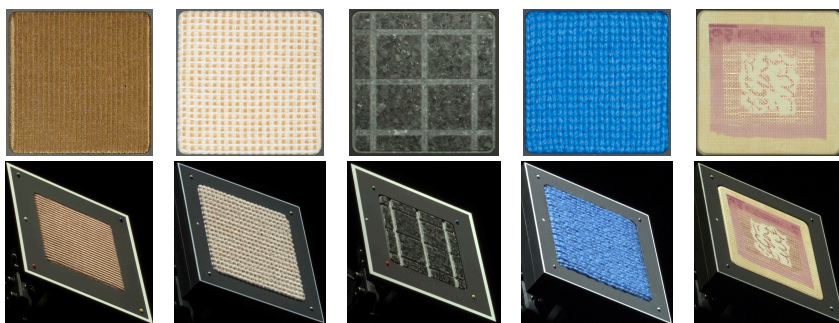


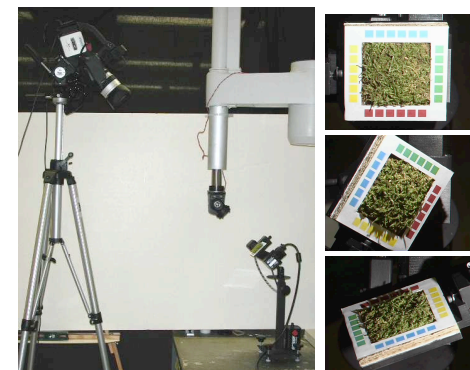
Figure 7: Measured BTF samples; from left to right (top row): CORDUROY, PROPOSTE, STONE, WOOL and WALLPAPER. Bottom row: perspective views ($\theta = 60, \phi = 144$) of the material and sample holder with illumination from ($\theta = 60, \phi = 18$). Note the mesostructure and changing colors.

[Müller et al. 2004]

Cornell CS569 Spring 2008

Lecture 18 • 7

Koudelka et al. BTF capture system

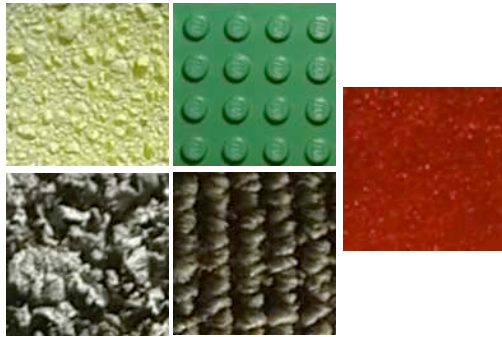


[Koudelka et al. 2003]

Cornell CS569 Spring 2008

Lecture 18 • 8

Koudelka et al. BTF data



[Koudelka et al. 2003]

Kaleidoscope BTF measurement

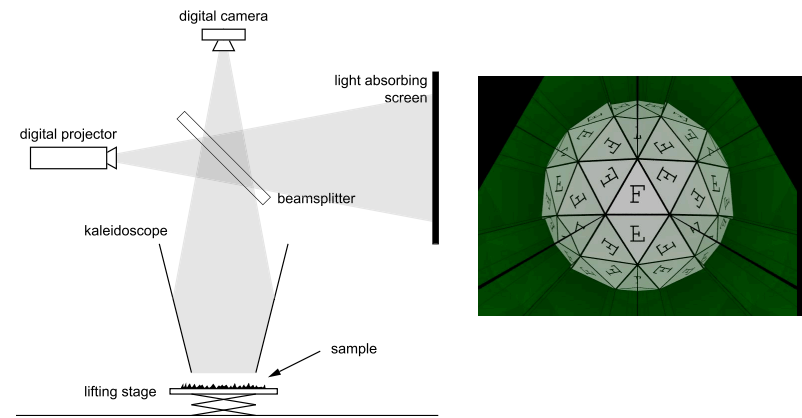
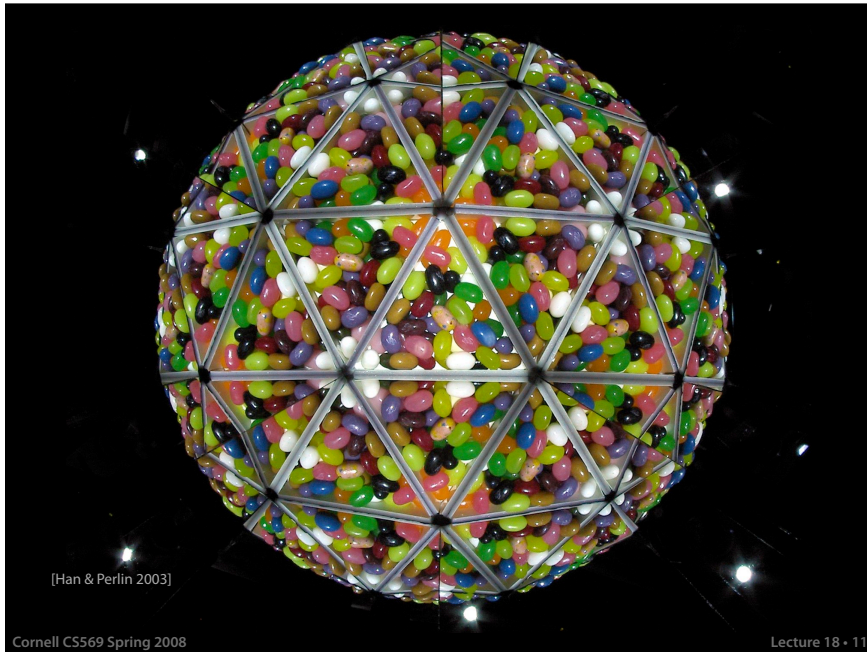


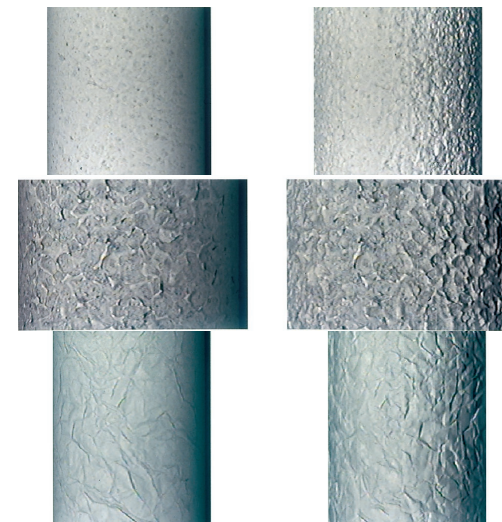
Figure 2: Schematic of optical components

[Han & Perlin 2003]



[Han & Perlin 2003]

Rendering using BTF




[Dana et al. 1999]

BTF rendering

- Dana et al. cylinders are just examples—no general alg.
- Requirement: texture for any view/illum
- Requirement: continuity
- Requirement: fast rendering
- Desirable: conserve texture memory
- Desirable: ability to tile/synthesize on arbitrary surfaces
- Key tool: low-rank approximation

SVD

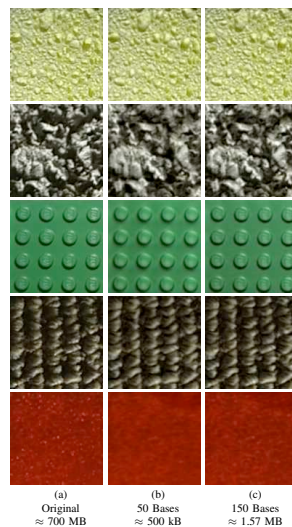
- If you are only going to remember one thing from linear algebra...

$$A = U \Sigma V^T$$


$$A = \sum_{k=0}^n \sigma_k u_k v_k^T$$

<http://www.uwlax.edu/faculty/will/svd/index.html>

Applying SVD on the whole matrix



[Koudelka et al. 2003]

Applying SVD on the whole matrix

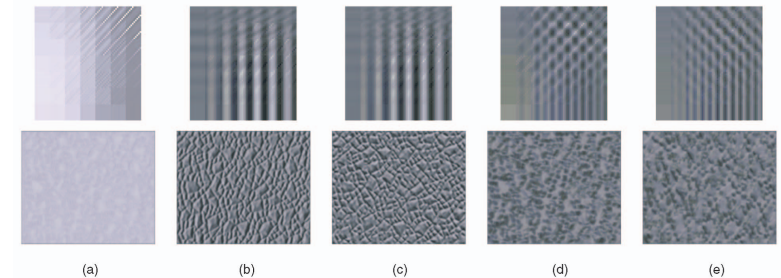


Fig. 1. The five most significant PAFs (top row) and the corresponding geometry maps (bottom row) of the BTF data "plaster04." As Table 1 shows, the singular value decreases from (a) to (e). A 4D PAF is packed into a 2D image consisting of a 2D array of subimages in which each row corresponds to a change in the lighting direction and each column corresponds to a change in the viewing direction. The azimuth angle is the faster changing parameter in each row and column.

[Liu et al. 2004]

Per-view PCA



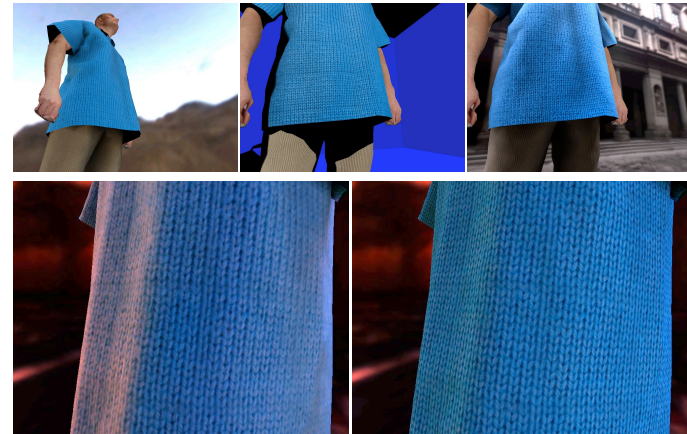
Figure 2: Comparing simple texture mapping and rendering from a sampled BTF

[Müller et al. 2004]

Cornell CS569 Spring 2008

Lecture 18 • 17

Per-view PCA

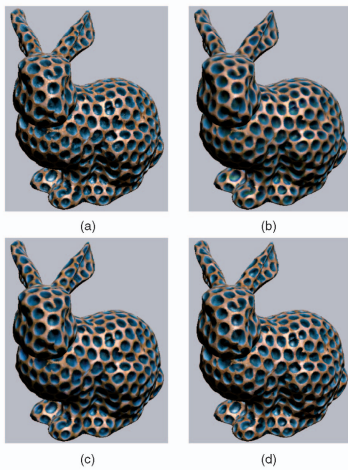


[Sattler et al. 2003]

Cornell CS569 Spring 2008

Lecture 18 • 18

BTF rendering by whole-matrix PCA



[Liu et al. 2004]

Fig. 10. Quality of surface BTF synthesized with N -term approximations for the BTF data "hole." (a) uses the original BTF data. (b), (c), (d) use 6, 12, 40-terms approximations, respectively. The synthesized results are visualized with the same view and light settings.

Cornell CS569 Spring 2008

Lecture 18 • 19