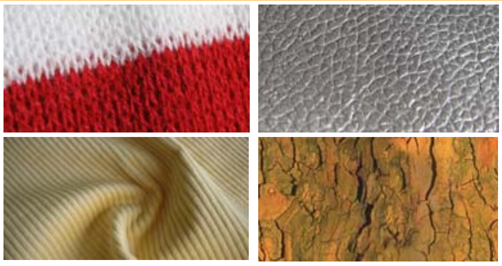
Capturing Reflectance
From Theory to Practice

Reflectance Fields for Distant Lights

Gero Müller

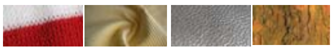
University of Bonn

Introduction



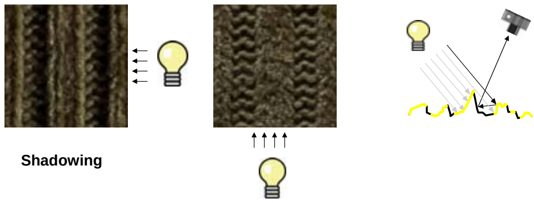
- Opaque materials with complex mesogeometry (rough textures)

Introduction

- Goal
  
  - Capture „look-and-feel“ of those materials independent of a specific physical object
- Capture appearance from material samples
- Standard: single RGB-image
  - Appearance captured only for one view and one lighting situation
  - Valid only for flat and diffuse materials (paper, cardboard,...)

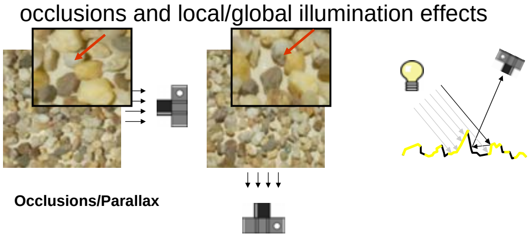
Introduction

- Images of rough textures with meso-structure contain view- and light dependent shadows, occlusions and local/global illumination effects



Introduction

- Images of rough textures with meso-structure contain view- and light dependent shadows, occlusions and local/global illumination effects



Introduction

- Lighting distance large compared to extent of material sample
- Materials are applied to opaque physical objects (furniture, walls, car interior, cloth, ...)

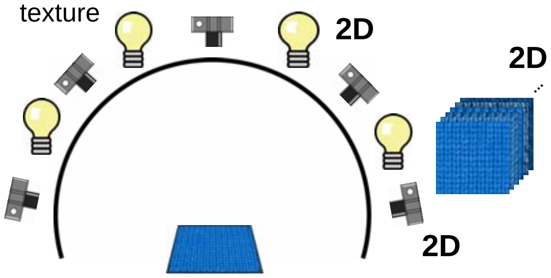
➔Neglect near-field illumination and explicit light-transport between surface points

➔Measure only far-field reflectance field of sample

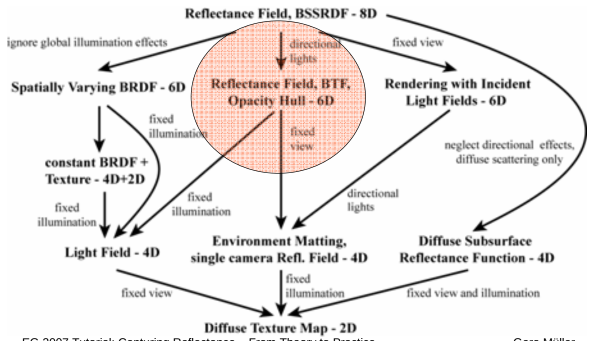
➔**B**idirectional **T**exture **F**unction [Dana et al. 1997]

Introduction

- BTF ⇔ 6D far-field reflectance field of texture



Taxonomy



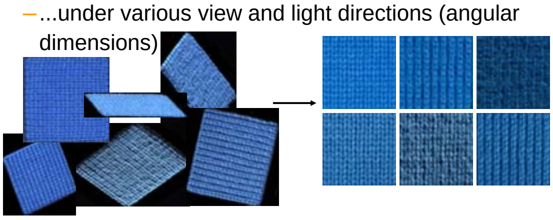
Overview

- Acquisition
- Compression
- Rendering
- Non-planar objects

Acquisition

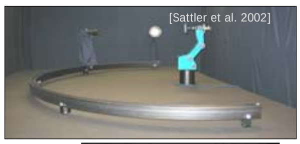
BTF Acquisition

- Sampling a 6D-function $BTF_{rgb}(\mathbf{x}, \mathbf{v}, \mathbf{l})$
  - Take pictures… (spatial dimension)
  - ...under various view and light directions (angular dimensions)



Measurement setups

- Gonioreflectometer-like
  - Advantages
    - fully automatic
    - flexible sampling rate
  - Problems
    - measurement time: ~14h (81x81=6561 images)
    - moving parts: camera, light, sample carrier



Measurement setups

- Using a camera array [Müller et al. 2005]
  - Advantages
    - fast, parallel ~1 hour (151x151=22501 images)
    - no moving parts
    - high resolution
  - Problems
    - fixed angular sampling
    - complex control apparatus

BTF Camera Array

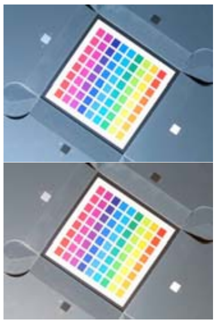
- Custom built hemi-spherical aluminium gantry (80 cm radius) mounted on aluminium base rack
- 151 Canon Powershot A75 digicams (3.2 mpixel)
  - cheapest consumer camera with powerful SDK
  - built-in light source (supports different intensities)
- USB-controllable 160-port relay box for on/off toggling
- Custom built power supply

Tasks

- Synchronized control
  - One camera flashes while all cameras take picture
  - High dynamic range
    - 4 passes with different flash intensities and exposures
  - 8 PCs (~19 cameras/PC)
  - 1 Master PC for synchronization

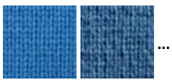
Tasks

- Calibration
  - Response curve
  - Color calibration
  - Varying flash intensity
  - Camera mapping
    - Lens distortion
    - Intrinsic + extrinsic camera parameters

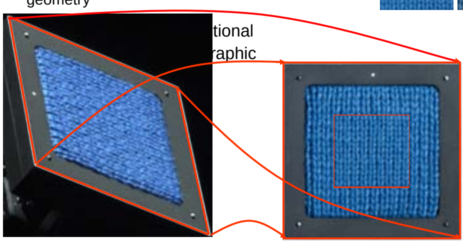


Post-processing

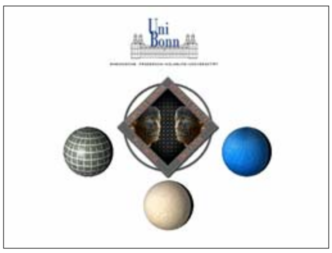
- Extraction of consistent ROI
  - Backprojection onto planar base geometry



- tional raphic

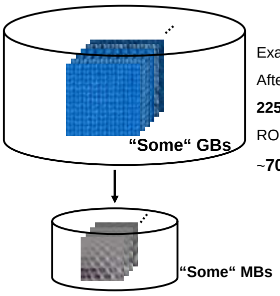


BTF Database Bonn



www.cg.cs.uni-bonn.de/btf

Compression



Example (Camera array):
After postprocessing:
22501 hdr-images (OpenEXR)
ROI-size **1024x1024**
~**70-90 GB**

Compression

- Preferable properties:
  - fast (real-time), random access decompression
  - preservation of visual important features
  - maximum of a few MBs
- Two main approaches
  - Fitting analytical functions
  - Statistical data analysis

Fitting Analytical Functions

- Spatial variation (texture domain) too complex

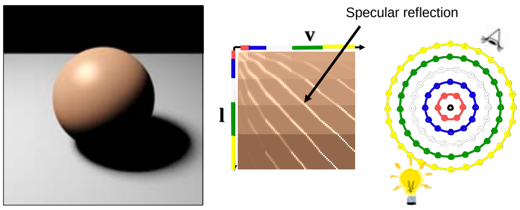
➔ Fixing spatial position

$$B_{\mathbf{x}}(\mathbf{v},\mathbf{l}) := BTF(\mathbf{x},\mathbf{v},\mathbf{l})$$

Apply techniques from BRDF modeling

Fitting Analytical Functions

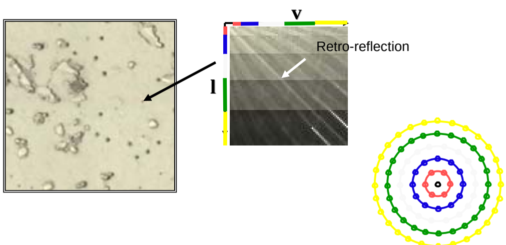
- How do these functions look like?
- How do typical BRDFs look like?



shiny plastic

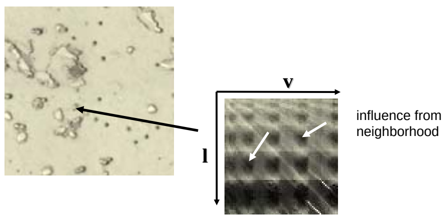
Fitting Analytical Functions

- How do these functions look like?



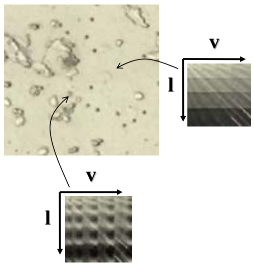
Fitting Analytical Functions

- Are these functions typical BRDFs?



Fitting Analytical Functions

- ABRDF = Apparent BRDF [Wong et al. 97]
  - Contains also influence from neighborhood:
    - Self-Shadowing
    - Self-Occlusion
    - Sub-Surface Scattering
    - Resampling artefacts
    - …



Fitting BRDF-Models

- Generalized Cosine-Lobe Model [Lafortune et al. 97] :

$$\Rightarrow BTF(\mathbf{x},\mathbf{v},\mathbf{l}) \approx f_{\mathbf{x}}(\mathbf{v},\mathbf{l},\mathbf{p}) = \rho_{d,\mathbf{x}} + \sum_{i=1}^{k} \rho_{s,\mathbf{x}} \cdot \left(\mathbf{v}^T \mathbf{D}_{\mathbf{x},i} \mathbf{l}\right)^{n_{\mathbf{x},i}}$$

diffuse color | specular color | specular lobe

- Non-linear least-squares fitting (Levenberg-Marquardt)
- typically around 2 lobes
- Improvement [Daubert et al. 2001]: view-dependent scale factor per texel to account for shadowing effects

Fitting BRDF-Models

- Advantages
  - High compression
  - Efficient evaluation
- Problems
  - loss of depth impression
  - Non-linear fitting
    - expensive
    - results depend on initialization

McAllister 2002

Statistical Data Analysis

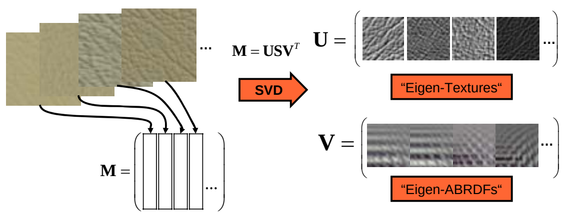
- Motivation
  - Assuming general basis functions (polynomials, lobes, etc…) suboptimal for a given measured BTF-dataset
- Idea
  - Find customized basis functions adapted for the actual data set
  - Exploit the inherent redundancy more effectively

Statistical Data Analysis

- Linear approaches
  - Full BTF-matrix factorization [Koudelka et al. 2003] [Liu et al. 2004]
  - Per-texel ABRDF factorization [Suykens et al. 2003]
  - Per-view factorization [Sattler et al. 2003]
  - Per-cluster factorization [Mueller et al. 2003]
- Tensor approaches
  - TensorTextures [Vasilescu et al. 2004]
  - Out-of-Core Tensor Approximation [Wang et al. 2005]

Full BTF-Factorization

- Stack images as column vectors into large matrix



Full BTF-Factorization

- Write the BTF as sum of products of two functions

$$\Rightarrow BTF(\mathbf{x},\mathbf{v},\mathbf{l}) = \sum_j^N \sigma_j \cdot t_j(\mathbf{x}) b_j(\mathbf{v},\mathbf{l}) \approx \sum_j^c \sigma_j \cdot t_j(\mathbf{x}) b_j(\mathbf{v},\mathbf{l})$$

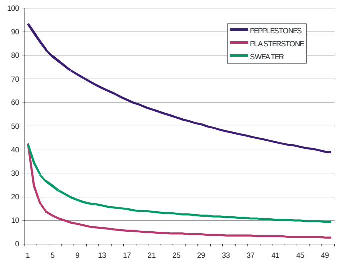
"Eigen-Textures" "Eigen-ABRDFs"

Full BTF-Factorization

- Number of terms

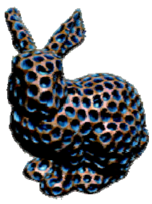
RMS-Error

$$\varepsilon = \sqrt{\sum_{i=c+1}^{N} \sigma_i^2}$$



Full BTF-Factorization

- Advantages
  - simple and straight-forward
- Problems
  - complex materials require many terms
  - ➔not suitable for real-time reconstruction



Liu et al. TVCG2004

Per-View Factorization

- Apply SVD to BTF-slices with fixed view direction (Spatially varying Hemispherical Functions)
- Idea
  - Increase quality of low-term factored approximations by factorizing fixed subsets of the data

$$\Rightarrow BTF(\mathbf{x},\mathbf{v},\mathbf{l}) \approx \sum_{v \in N(\mathbf{v})} w_v \sum_j^c r_{v,j}(\mathbf{l}) t_{v,j}(\mathbf{x})$$

"Eigen-Hemispherical-Functions" "Eigen-Textures" (per view)

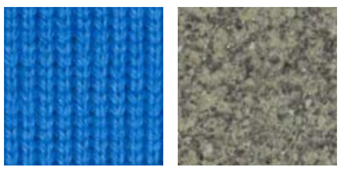
Per-View Factorization

- Advantages
  - Low-term factorization enables high-quality interactive rendering on graphics hardware
- Problems
  - Memory consumption
  - Coherence between different views not exploited

Sattler et al. 2003

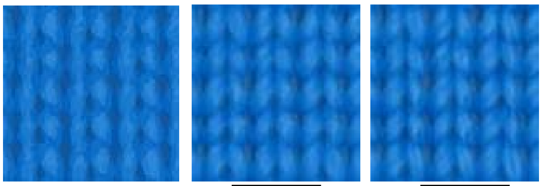
Statistical Data Analysis

- Per-texel or per-view factorization factorize fixed subsets of the BTF data
- Use clustering across spatial dimension to find better subsets



Statistical Data Analysis

- Clustering alone leads to quantization artifacts



3 components   8 components

- Solution: linear approximation of data in each cluster (Local-PCA)

Per-Cluster Factorization

- Clustering BTF-texels (ABRDFs) leads to

$$\Rightarrow BTF(\mathbf{x},\mathbf{v},\mathbf{l}) \approx \sum_{j}^{c} t_j(\mathbf{x}) b_{k(\mathbf{x}),j}(\mathbf{v},\mathbf{l})$$

"Eigen-Textures" (segmented)   "Eigen-BRDFs" (per cluster)

- Clustering with generalized Lloyd-algorithm and re-construction error as distance metric

Per-Cluster Factorization

- Advantages
  - Low-term factored representation suitable for GPU implementation
  - Good compression
  - Reconstruction per cluster reduces quantization artifacts
- Problems
  - Expensive fitting
  - Mip-Mapping

Mueller et al. 2003

Storage Requirements

Model	Storage (|L|,|V|,|T|)	|V|=|L|=81, |T|=256² 8-Bit per channel
Raw BTF	|L|*|V|*|T|	1.2 GB
Analytical BRDF-Model	f(k)*|T|	(k=2) 2.4 MB
Hemispherical Function	f(k)*|V|*|T|	(k=2) 95 MB
BTF Factorization	c*(|V|*|L| + |T|)	(c=40) 8.6 MB
ABRDF Factorization	d*(|V|+|L|)*|T|	(d=2) 63 MB
Per-View Factorization	|V|*c*(|L|+|T|)	(c=4) 64 MB
Per-Cluster Factorization	c*(k*(|V|*|L|) + |T|)	(k=32, c=8) 6.6 MB

Practical Issues

- Factorization approaches require computing SVD of large matrices (up to several GBs)
- Use incremental/online SVD methods
  - Arnoldi iteration
  - EM-PCA [Roweis 1998]
  - Online SVD [Brand 2003]
  - ...

Using geometry information

- Fitting local coordinate systems
  - In-between image- and geometry-based BTF representation
- Can be done efficiently using FFT over the group of rotations SO(3) [Müller et al. EG2006]

2 SVD components, 200 KB   aligned, 2 SVD components, 200 KB   uncompressed 900 MB

Rendering

Rendering

- Determine color / visible radiance for every point

„Exitant Radiance = Emitted Rad. + Reflected Rad.“

$$L_r(\mathbf{x},\mathbf{v}) = L_e(\mathbf{x},\mathbf{v}) + L_{ref}(\mathbf{x},\mathbf{v})$$

„Reflected Rad. = Incoming Rad. combined with reflection properties“

$$L_{ref}(\mathbf{x},\mathbf{v}) = \int_{\Omega_i} \rho_{\mathbf{x}}^*(\mathbf{v},\mathbf{l}) \cdot L_i(\mathbf{x},\mathbf{l})\, d\mathbf{l}$$

spatially varying reflectance includes foreshortening term

Point- and Directional Light Sources

- Finite number of light directions

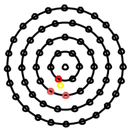
$$L_{ref}(\mathbf{x},\mathbf{v}) = \int_{\Omega_i} \rho_{\mathbf{x}}^*(\mathbf{v},\mathbf{l}) \cdot L_i(\mathbf{x},\mathbf{l})\, d\mathbf{l}$$

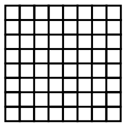
$$\downarrow$$

$$L_{ref}(\mathbf{x},\mathbf{v}) = \sum_{i=1}^{n} \rho_{\mathbf{x}}^*(\mathbf{v},\mathbf{l}_i) \cdot \hat{L}_i(\mathbf{x},\mathbf{l}_i)$$

Rendering with GPUs

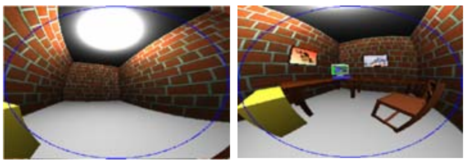
- Measured BTFs
  - Evaluation for directions not in the measured set
    - Interpolation in angular domain
  - Interpolation rather expensive
    - ➔graphics hardware
    - ➔Interpolation from regular samples





Hardware Supported Angular Interpolation

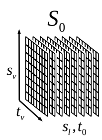
- Reparameterization
  - Approximately uniform sampling of hemisphere
  - Suitable for hardware filtering
  - ➔Parabolic Maps



Heidrich + Seidel 1998

Hardware Supported Angular Interpolation

- 2D Data
  - Bilinear filtering on graphics hardware
- 4D Eigen-ABRDFs
  - Quadrilinear filtering
  - Hardware: trilinear filtering
  - ➔Trilinear filtering of s_v, t_v, s_l
    - ➔3D textures S_i for fixed t_l
  - ➔Interpolate t_l in fragment shader

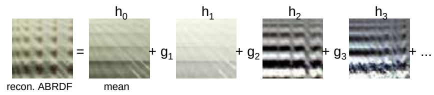


Anti-Aliasing

- Mip-Mapping compressed BTFs
  - No problem for Eigen-Texture based compression (full-matrix factorization, per-view factorization)
  - Other techniques depend non-linear on compression parameters
  - GPU supported Mip-Mapping not possible
  - ➔ Standard Mip-Mapping on uncompressed data
  - ➔ Compression of each individual Mip-Map level

Decompression on GPU

- Full-BTF Factorization/Per-Cluster Factorization
  - Store 4D ABRDFs in 3D texture
  - Use 4D interpolation and combine in pixel shader
  - Cluster look-up



Results

Image-Based Lighting of BTFs

HDR environments wood, beach, kitchen, building and uffizi from www.debevec.org

Software Rendering

- Global Illumination
  - Decompression on CPU

Non-Planar Objects

- BTF techniques can be applied to non-planar objects
  - [Furukawa et al. EGRW 2002]
  - [Matusik et al. SIG 2002]
  - [Mueller et al. VAST 2005]
- Use 3D reconstructed base-geometry instead of planar base geometry

Non-Planar Objects

- Processing steps
  - –Image acquisition
  - –Image-based 3D-reconstruction
  - –Mesh parameterization
  - –BTF generation
  - –BTF compression

Non-Planar Objects

Conclusions

- BTFs capture 6D-slice of the reflectance field of a complex material
- Represents the "look-and feel" of a material
- Several high-quality acquisition setups
- Effective and appearance preserving compression algorithms available
- Real-time rendering possible with point light sources and image-based lighting

Challenges

- Editing and modeling
  - [Kautz et al. SIG 2007]
  - [Müller et al. EGSR 2007]
- Material Perception
- Time variation (recent work only SVBRDFs)
- Spectral measurements
- Highly reflective materials

Acknowledgements

- Reinhard Klein, Ralf Sarlette, Dirk Koch, Jan Meseth, Mirko Sattler
- EPOCH NoE
- RealReflect
- University of Bonn