Tutorial

Bidirectional Texture Functions Modelling

Data Visualization

Michal Haindl Jiří Filip

Institute of Information Theory and Automation
of the AS CR

13/06/2010
CVPR 2010

www.utia.cz

Outline - BTF Visualization

- Our Goal
- Steps to achieve it:
  - Texture mapping
  - World vs. local coordinate system transformation
  - Local texture space
  - Interpolation in texture space
  - Measured directions Interpolations
  - Point-light vs. environment illumination
  - Implementation

BTF Visualization

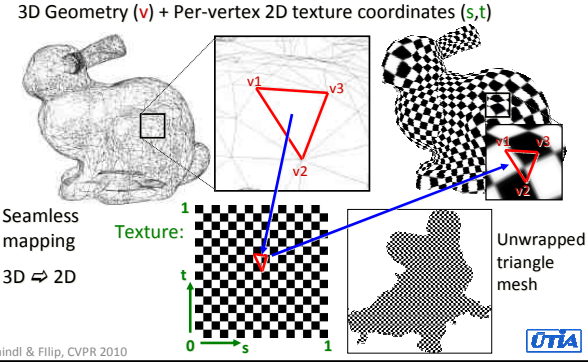
Material measurement → Material Visualization

$$BTF(\lambda, r_1, r_2, \theta_i, \varphi_i, \theta_v, \varphi_v)$$

Spectral Spatial Illumination & view direction dependency

Texture Mapping

3D Geometry (v) + Per-vertex 2D texture coordinates (s,t)

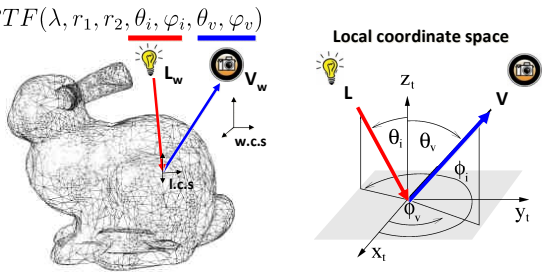


Seamless mapping

3D ⇨ 2D

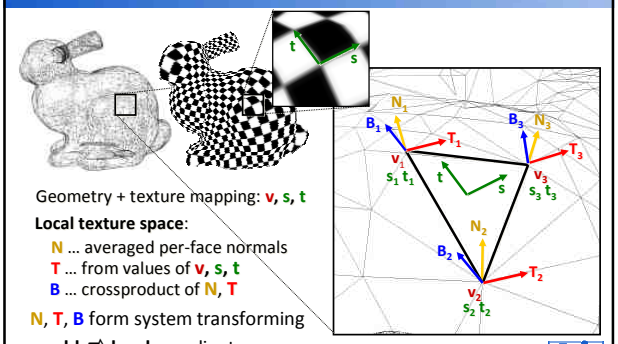
World vs. Local Coordinate System

$$BTF(\lambda, r_1, r_2, \theta_i, \varphi_i, \theta_v, \varphi_v)$$



World light and camera positions - 3D world coordinates L_w V_w .
Coordinates in local texture space **L**, **V** ?

Local Texture Space



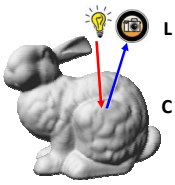
Geometry + texture mapping: **v**, **s**, **t**

Local texture space:
- **N** ... averaged per-face normals
- **T** ... from values of **v**, **s**, **t**
- **B** ... crossproduct of **N**, **T**

N, **T**, **B** form system transforming **world** ⇨ **local** coordinates

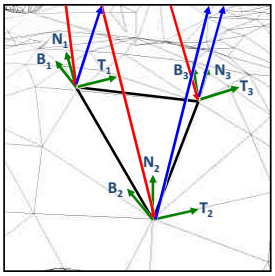
Local Texture Space

Obtaining illumination and view directions in local texture space:

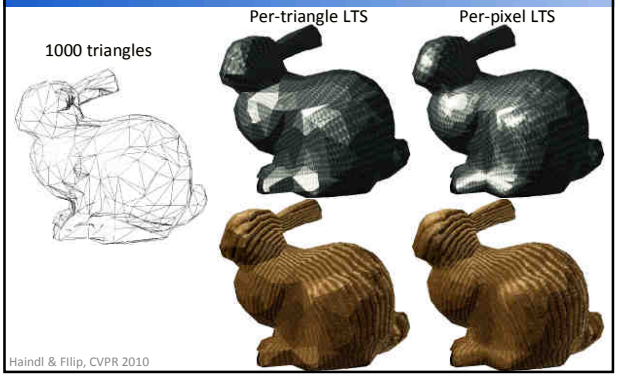


$$L = \begin{bmatrix} T_x & T_y & T_z \\ B_x & B_y & B_z \\ N_x & N_y & N_z \end{bmatrix} L_w$$

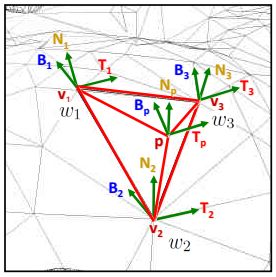
$$C = \begin{bmatrix} T_x & T_y & T_z \\ B_x & B_y & B_z \\ N_x & N_y & N_z \end{bmatrix} C_w$$



Local Texture Space (LTS) Interpolation



Local Texture Space Interpolation



Intra-triangle interpolation of T, B, N is supported in shaders of graphics HW.

Barycentric interpolation obtains weights as a ratios of corresponding triangles areas.

$$w_1 = \frac{\triangle(\mathbf{p}, \mathbf{v}_2, \mathbf{v}_3)}{\triangle(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)}$$

$$w_3 = \frac{\triangle(\mathbf{p}, \mathbf{v}_1, \mathbf{v}_2)}{\triangle(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)}$$

$$w_2 = \frac{\triangle(\mathbf{p}, \mathbf{v}_1, \mathbf{v}_3)}{\triangle(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)}$$

$$w_1 + w_2 + w_3 = 1$$

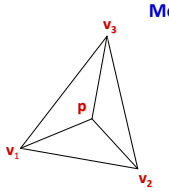
$T_p = w_1 T_1 + w_2 T_2 + w_3 T_3$

$B_p = w_1 B_1 + w_2 B_2 + w_3 B_3$

$N_p = w_1 N_1 + w_2 N_2 + w_3 N_3$

Barycentric Interpolation

Mobius 1827 [Coxeter 1969, Fauvel et al. 1999]



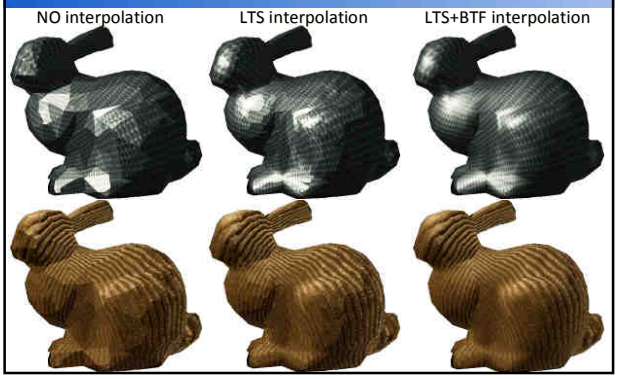
$$w_1 = \frac{\|(\mathbf{v}_2 - \mathbf{p}) \times (\mathbf{v}_3 - \mathbf{v}_2)\|}{\|(\mathbf{v}_2 - \mathbf{v}_1) \times (\mathbf{v}_3 - \mathbf{v}_2)\|}$$

$$w_2 = \frac{\|(\mathbf{v}_3 - \mathbf{p}) \times (\mathbf{v}_1 - \mathbf{v}_3)\|}{\|(\mathbf{v}_2 - \mathbf{v}_1) \times (\mathbf{v}_3 - \mathbf{v}_2)\|}$$

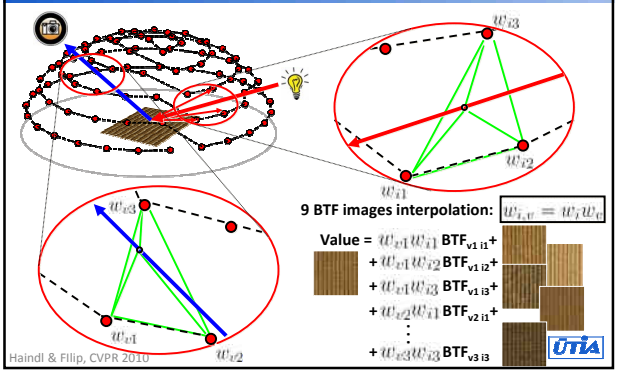
$$w_3 = \frac{\|(\mathbf{v}_1 - \mathbf{p}) \times (\mathbf{v}_2 - \mathbf{v}_1)\|}{\|(\mathbf{v}_2 - \mathbf{v}_1) \times (\mathbf{v}_3 - \mathbf{v}_2)\|}$$

- If w_1, w_2 and w_3 are all greater than zero, $\mathbf{p}$ is strictly inside the triangle.
- If $w_i = 0$ and the other two coordinates are positive, $\mathbf{p}$ lies on the edge opposite $\mathbf{v}_i$.
- If $w_i = 0$ and $w_j = 0$, $\mathbf{p}$ lies on $\mathbf{v}_k$.
- If $w_i < 0$, $\mathbf{p}$ lies outside the edge opposite $\mathbf{v}_i$.

Measured Directions Interpolation

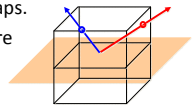


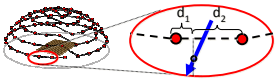
Measured Directions Interpolation



Measured Directions Interpolation

- Interpolation ⇨ for each pixel (fragment) of rendered object,
- 3 closest directions indices & interpolation weights can be precomputed and stored, e.g. in cube-maps.
- Easy implementation in graphics hardware
- For elevation angles higher than those measured, a modification is required:





BRDF Rendering

- Total material radiance ⇨ emitted + reflected:

$$L_r(x,y,\omega_v) = \cancel{L_{em}(x,y,\omega_v)} + L_{rf}(x,y,\omega_v)$$

- General rendering equation

$$L_r(x,y,\omega_v) = \int_{\Omega_i} BRDF(\omega_i,\omega_v) L_i(x,y,\omega_i) \cos(\theta_i) d\omega_i$$

Surface reflectance

Illum./view direction

Illumination contribution

Illuminated area foreshortening

BTF Rendering

- Reflected light integration over hemisphere
  - Foreshortening term comprised in BTF data

$$L_r(x,y,\omega_v) = \int_{\Omega_i} BTF(x,y,\omega_i,\omega_v) L_i(x,y,\omega_i) d\omega_i$$

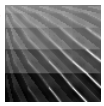
  - Impossible to solve analytically ⇨
- Rendering for a single light ⇨ unrealistic

$$L_r(x,y,\omega_v) = BTF(x,y,\omega_i,\omega_v) L_i(x,y,\omega_i)$$

- Approximation by a finite number of lights

$$L_r(x,y,\omega_v) = \sum_i^{n_i} BTF(x,y,\omega_i,\omega_v) L_i(x,y,\omega_i)$$

Mean pixel luminance – leather BTF



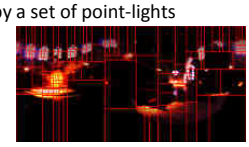
Point-Light vs. Environment Illumination

- Point-light
  - Idealized, non-realistic
  - Fast computation
- Environment Illumination
  - Realistic
  - More demanding comp.

Environment Illumination

[Debevec 05, Median Cut]

- Environment approximated by a set of point-lights



- Smoother images - convolution of pixel with lights
- Slower rendering - optimization techniques are available

Environment Illumination

Implementation of BTF Visualization

- Some processing steps can (has) to be precomputed:
  - Texture mapping
  - Local texture space (LTS) on 3D object
  - Measured directions Barycentric interpolation
- Other have to be computed per-pixel on the fly:
  - Conversion illumination and view directions to LTS
  - LTS interpolation across triangles (HW supported)
  - BTF model evaluation
  - Environment illumination
- Benefit from parallelized graphics hardware
  - Shading languages: OpenGL+GL Shading Language, CG,
  - Higher level programming: CUDA, OpenCL

GPU Online Shading Example

- Part of car interior

  (interactive ~14 fps)

GPU Online Shading Example

Interpolated BTF Visualization - Example

References

Barycentric coordinates

- H. S. M. Coxeter. *Introduction to Geometry*. New York: Wiley, 1969.
- J. Fauvel, R. Flood, and R. J. Eds. Wilson. *Mobius and his Band: Mathematics and Astronomy in Nineteenth-Century Germany*. Oxford University Press, Oxford, England, 1993.