FAKULTÄT FÜR !NFORMATIK

Faculty of Informatics

Diplomarbeitspräsentation

Interactive 3D Reconstruction and BRDF Estimation for Mixed Reality Environments

Masterstudium:
Computergraphik & Digitale Bildverarbeitung

Georg Tanzmeister

Technische Universität Wien
Institut für Computergraphik und Algorithmen
Arbeitsbereich Computergraphik
Betreuerung: Associate Prof. Dipl.-Ing Dipl.-Ing Dr. Michael Wimmer
Mitwirkung: Dipl.-Ing. Dr. Christoph Traxler
Dipl.-Ing. Mag. Martin Knecht

Scope and Motivation

typical mixed reality system

- Requires to have **geometry**, **material** and **lighting** information about the real world scene
- All of these can change dynamically in mixed reality
- Estimations need to be done in real time since interaction should be possible

desired mixed reality system

Contribution

We developed a BRDF estimation algorithm that

- runs at interactive frame rates
- can handle dynamic scenes like moving objects, insertion or removal of objects and lighting changes
- does not need any pre-processing

One reason for the significant speed-up was

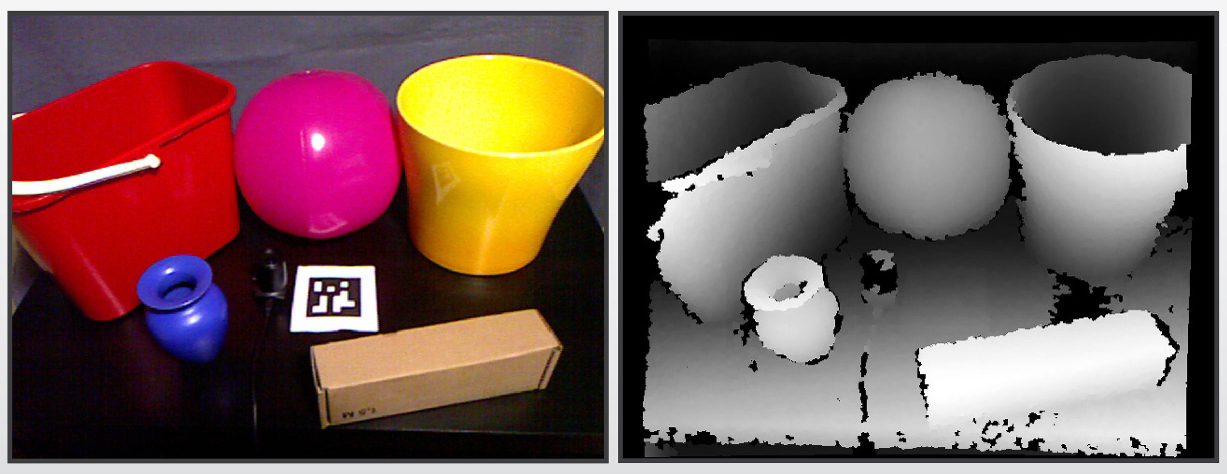
- a novel GPU K-Means implementation using MIP maps

Algorithm

Based on the Phong reflection model as stated below, the goal is to estimate the diffuse (k_d) and specular (k_s, n_s) characteristics of all the objects in the scene:

$$I = k_a I_a + \sum_{l=1}^{n} I_l [\overset{?}{k_d}(N \cdot L_l) + \overset{?}{k_s}(V \cdot R_l)^{\overset{?}{n_s}}]$$

Data Acquisition



Using the Microsoft Kinect sensor color (left) and depth (right) data is acquired.

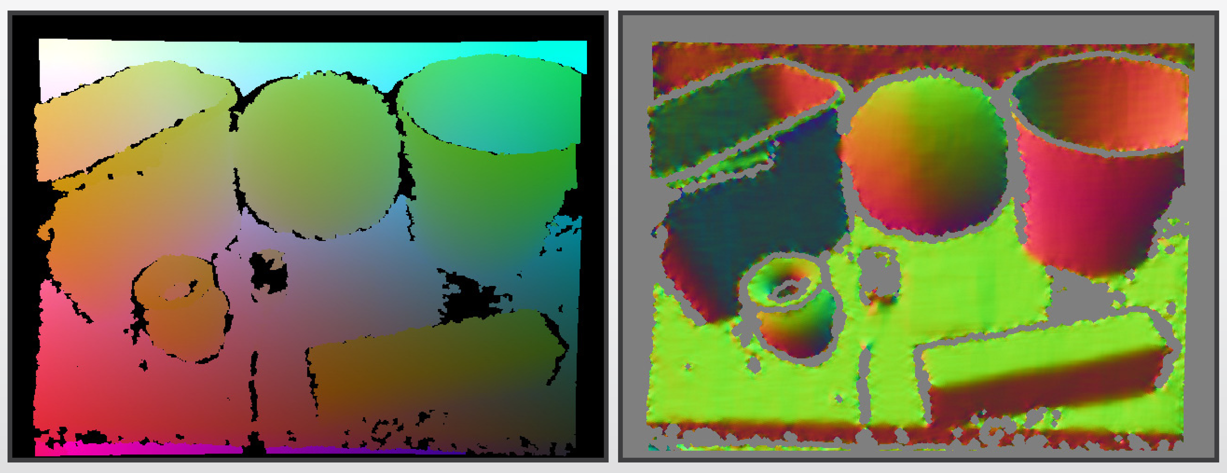
The environment is observed using a fish-eye lens camera.

Results

We achieve ~2 estimations per second*, which is a huge speed-up compared to 20 minutes for a single one and we thus have proved that real-time estimations are possible. Additionally, our approach is able to handle changes of the lighting and the geometry.
Left is a Phong rendering of the estimated parameters and right an integration of the technique into a mixed reality system. Note the red color bleeding on the virtual Buddha and the real objects and that the pocket lamp is in fact turned off. Apart from the Buddha nothing was modeled, so all the geometry and color information needed for the rendering was estimated by the algorithm.

* on GTX 580 and Core i7-2600

Data Extraction



The depth data is used to calculate 3D point locations (left) and point normals (right).

Importance sampling of the environment map is used to place point lights on a hemisphere.

Specular and Diffuse Estimation

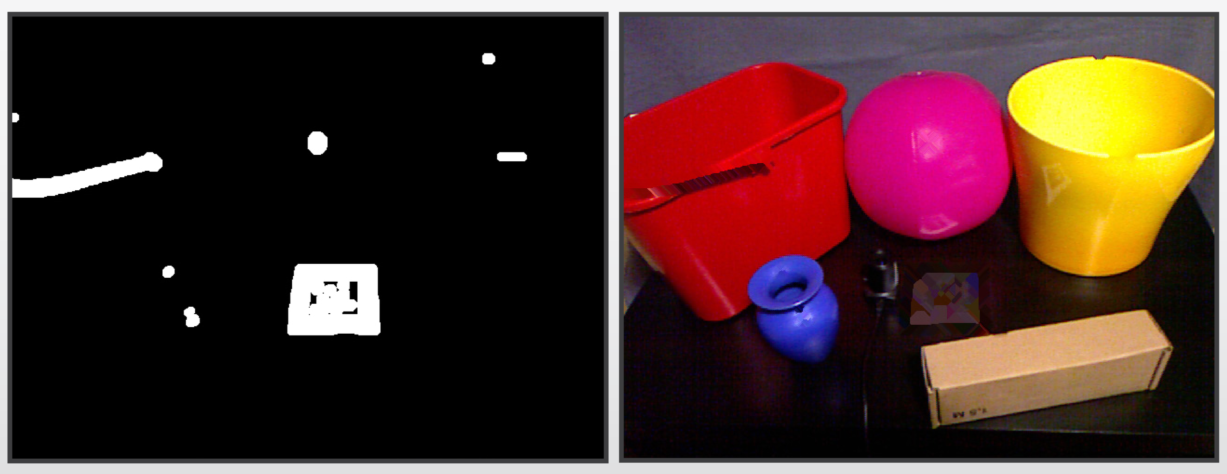
Specular estimation using non-linear least squares per cluster.

$$Fj = \sum_i [I_{i\,KINECT} - I_{i\,RENDERING}]^2$$

Diffuse estimation per pixel.

$$k_d = \frac{(I - k_a I_a - \sum_{l=1}^{n} I_l k_s (V \cdot R_l)^{ns}}{\sum_{l=1}^{n} I_l (N \cdot L_l)}$$

Highlight Removal



Highlight pixels are detected as those having high brightness and low saturation values yielding a highlight mask (left). This mask is used to apply a simple inpainting technique to obtain a specular free image (right).

Inverse Diffuse Shading

Inverse diffuse shading is used to flatten objects yielding a more uniform color distribution.

Clustering

Similar colors are assumed to belong to similar materials and are therefore clustered together in order to estimate common specular reflection characteristics.

Clustering is done using a novel GPU K-Means implementation. In contrast to almost all other implementations of K-Means our method runs entirely on the GPU.

Kontakt: georg.tanzmeister@tuwien.ac.at