MPEG-guided Global Illumination Rendering

Robert Herzog,
Shinichi Kinuwaki,
Karol Myszkowski

Motivation

- Global illumination (GI) rendering: main bottleneck in particular for animations
- Exploit lossy MPEG compression metric to accelerate rendering (i.e. no need to compute lighting details that are invisible after compression)
- GI rendering controlled by MPEG has not been investigated so far (though for static images by means of JPEG compression [Bolin SIGGRAPH95])

Planned Contributions

- Temporally coherent global illumination
  - Lightcut with Instant global illumination algorithm extended into temporal domain
  - (or: some hierarchical photon density estimation)
  - Number of VPLs controlled by the compression level
    - → Interactive frame-rates with dynamic scenes and camera
- Video compression side
  - Perception-motivated compression
  - Exact and fast motion estimation based on pixel flow
- Video Streaming at Client/Server Platforms
  - Thin clients may have limited computational capabilities
  - No need for rendering software and storage of (huge) 3D data on the client side
  - Simple control over up-to-date confidential data

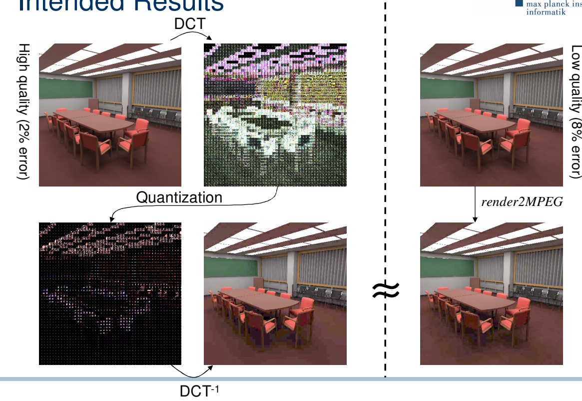
Applications

- Remote access to 3D data
  - Maintenance and repair purposes
- Industrial and architectural design
- 3D navigation and tourism
- Interactive online (mobile) entertainment
- Medical applications
- Commercial products (so far separate rendering and compression side with high latency):
  - RealityServer from mental images
  - Visage CS Thin Client/Server from Mercury Comp. Systems

Selling points

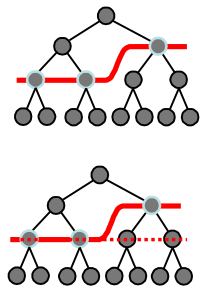
- Rendering: rendering up to the quality allowed by lossy MPEG compression (perceptual image quality metric)
  - (May also exploit frequency space (DCT) for efficient noise filtering as a post-process in image space)
- Compression: geometric information available and better temporal coherence → more efficient compression

Intended Results



Extensions to Lightcuts

- Refine light partition (light cut) according to perceptual error metric, i.e. stop refining if light cut error < quantization error in MPEG
- Requires mapping to discrete cosine transform (DCT) and evaluating difference of quantized DCT coefficients at each refinement step
- Not scientifically sound but DCT transformation on an 8x8 block level is cheap! (two 8x8 matrix multiplications)
- According to MPEG we compute lightcuts at a block level of 8x8 pixels
- Reuse lightcuts from previous frame by reprojection of blocks (possibly on a per pixel level)



Open Questions

- How to map the proposed algorithm to GPU?
  - DCT can easily be implemented on GPU! What about the visibility computation (shadow maps?)
  - (GPU GI algorithm with *local* quality control possible?)
- Directly computing error bounds (i.e. relative upper bound lighting differences) in frequency space (without DCT <-> IDCT transforms)?
  - Similar to orthogonal series density estimation?