Efficient Shadows for GPU-based Volume Raycasting

Timo Ropinski Jens Kasten Klaus Hinrichs

Visualization and Computer Graphics Research Group
University of Münster, Germany

WSCG 2008

Motivation

We wanted to find out ...

- ... which shadow algorithms can be combined with GPU-based volume raycasting.
- ... which algorithm is most efficient and effective.

Overview

1 Introduction and Related Work

2 Implementation of Shadow Techniques
- Shadow Rays
- Shadow Mapping
- Deep Shadow Maps

3 Performance and Results

4 Conclusions and Future Work

Introduction and Related Work

Object-based

- shadow volumes by Crow (1977)

Introduction and Related Work

Object-based

- shadow volumes by Crow (1977)

Image-based

- shadow mapping by Williams (1978)
- opacity shadow maps by Kim/Neumann (2001)
- deep shadow maps by Lokovich/Veach (2000)

Introduction and Related Work

Object-based

- shadow volumes by Crow (1977)

Image-based

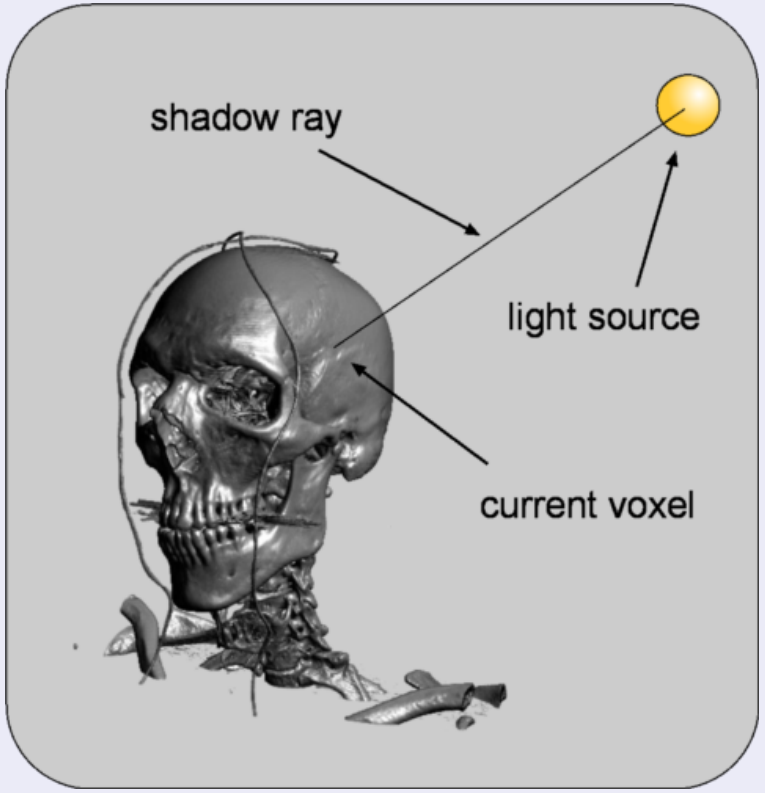
- shadow mapping by Williams (1978)
- opacity shadow maps by Kim/Neumann (2001)
- deep shadow maps by Lokovich/Veach (2000)

Compared approaches

- Shadow Rays
- Shadow Mapping
- Deep Shadow Maps

Shadow Rays

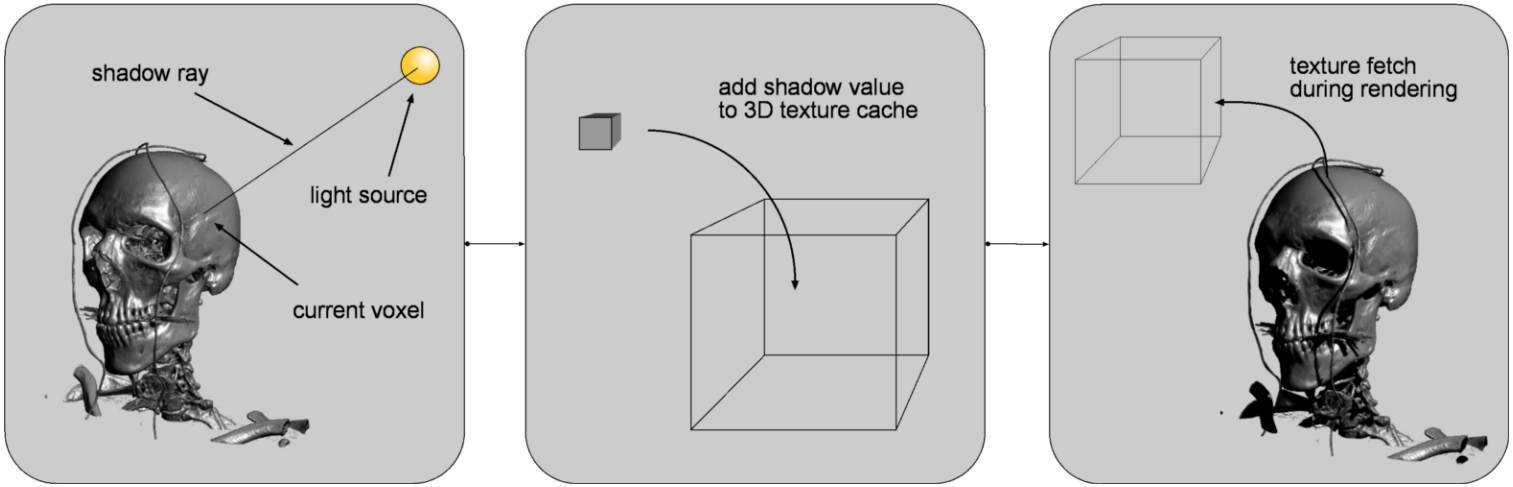
- similar to shadow rays in raytracing
- 3 versions implemented
  - opaque occluders (similar to first order intersections)
  - alpha raycasting
  - 3D textures for result caching



3D-Texture Rendering

Performance gain with 3D textures

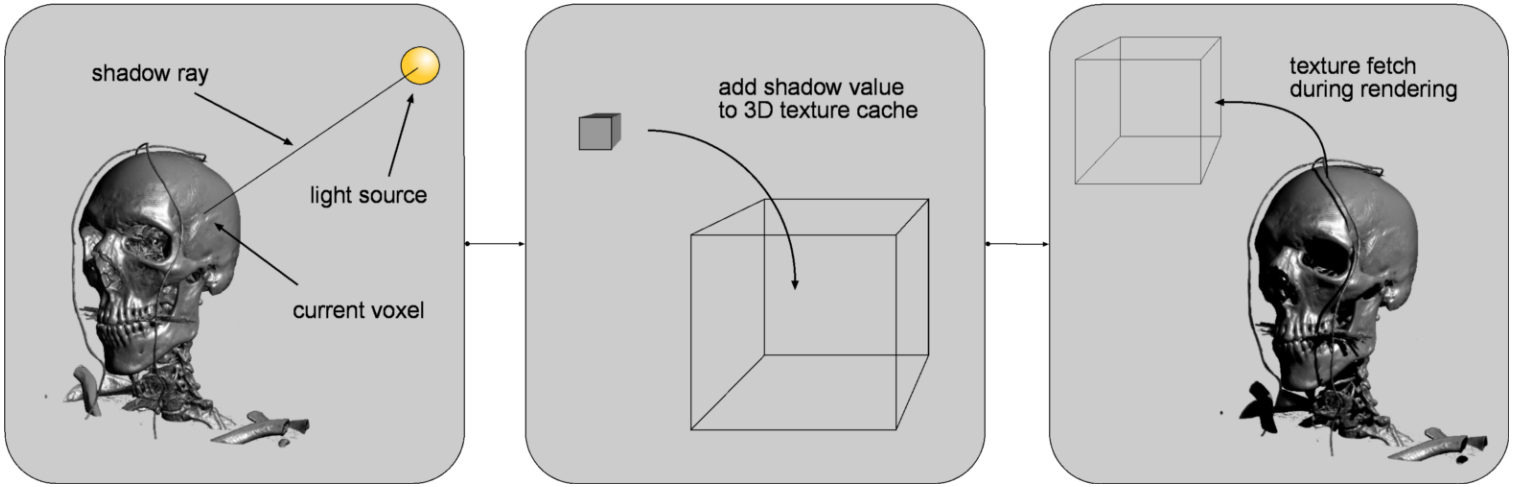
- only one 3D texture lookup when preprocessing shadow feelers
- needs to be recomputed when changing light source position or transfer function



3D-Texture Rendering

Performance gain with 3D textures

- only one 3D texture lookup when preprocessing shadow feelers
- needs to be recomputed when changing light source position or transfer function

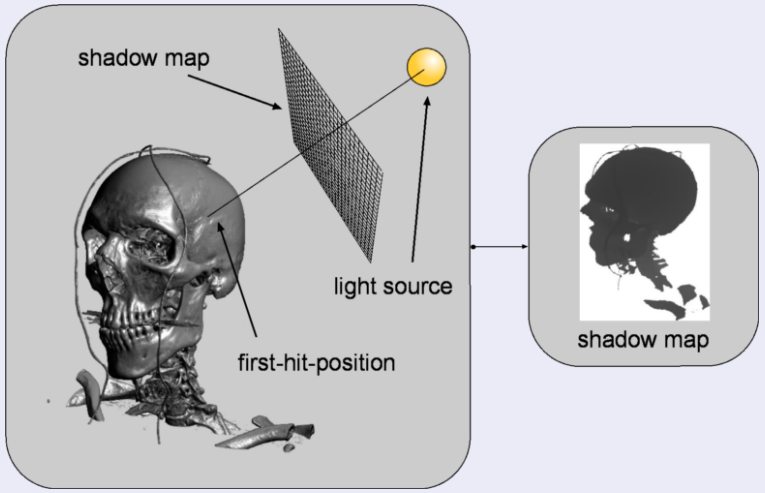


Implementation details

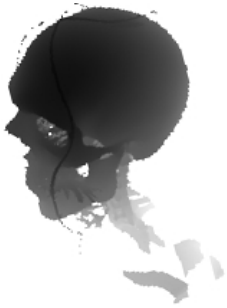
- using framebuffer objects
- render to texture slice by slice

Shadow Mapping

- map saves depth values of first hit points seen from the light source
- depth comparison during rendering gives binary decision for shadowing
- shadow threshold marks intensity limit
- supporting opaque occluders only

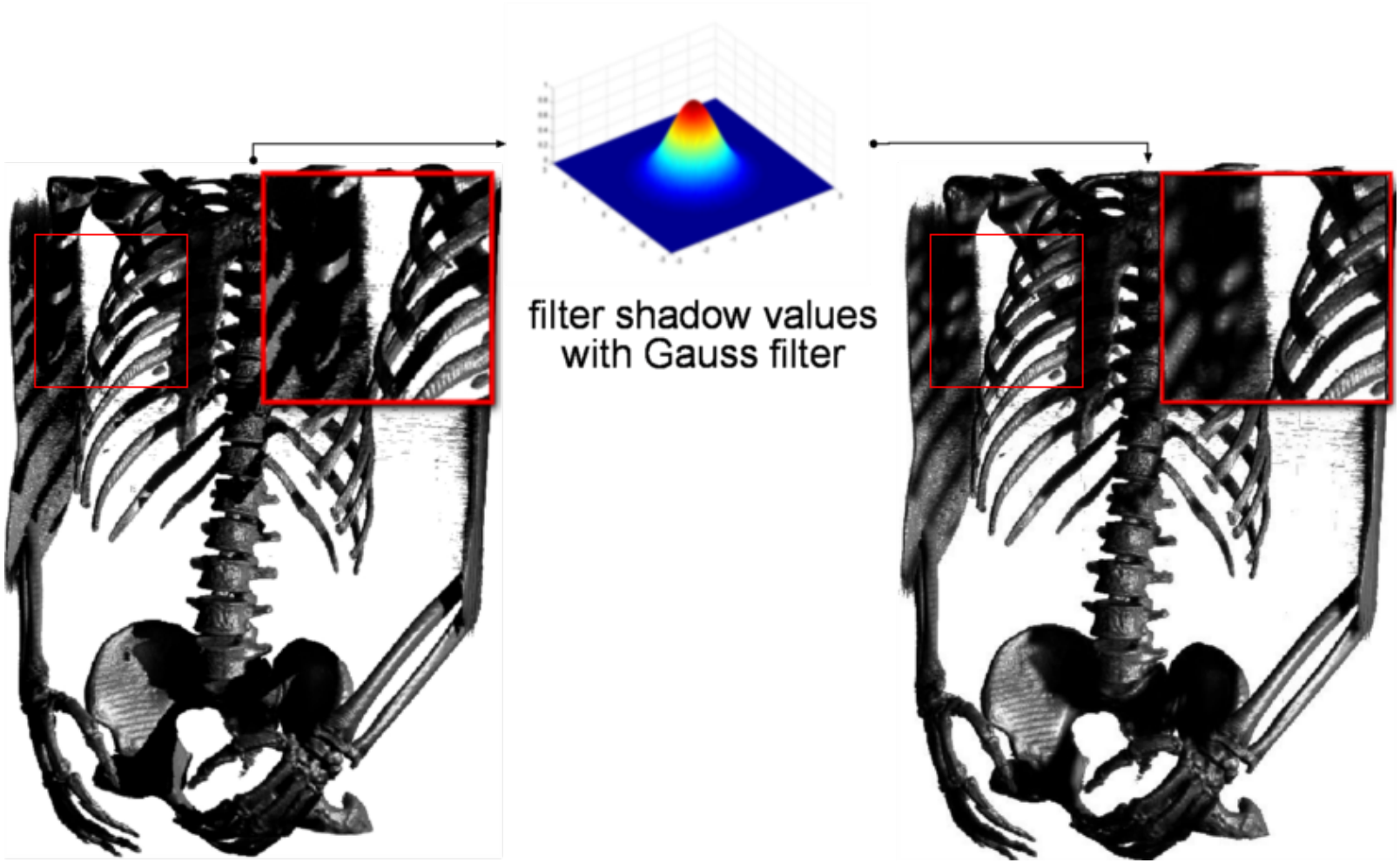


first hit image



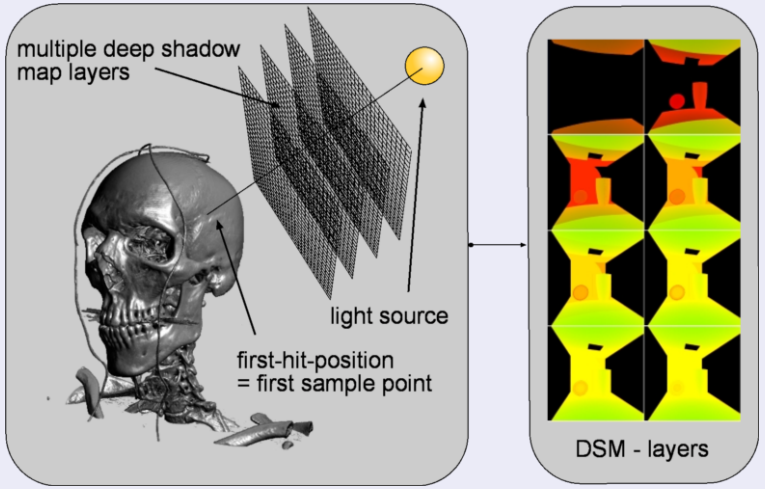
shadow map

Shadow Mapping



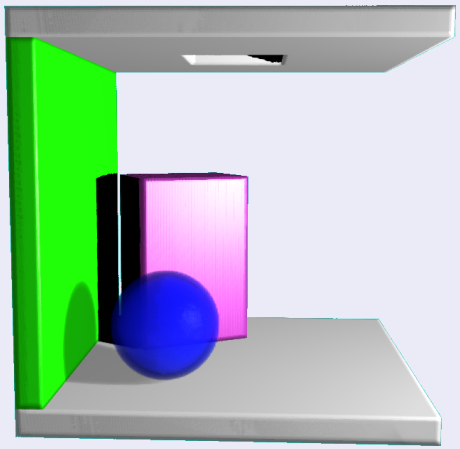
Deep Shadow Maps

- supports semi-transparent occluders
- shadow map with eight layers (pair of depth and transparency value per layer)
- entire map is rendered on the fly
- deep shadow maps are combined with 3D textures

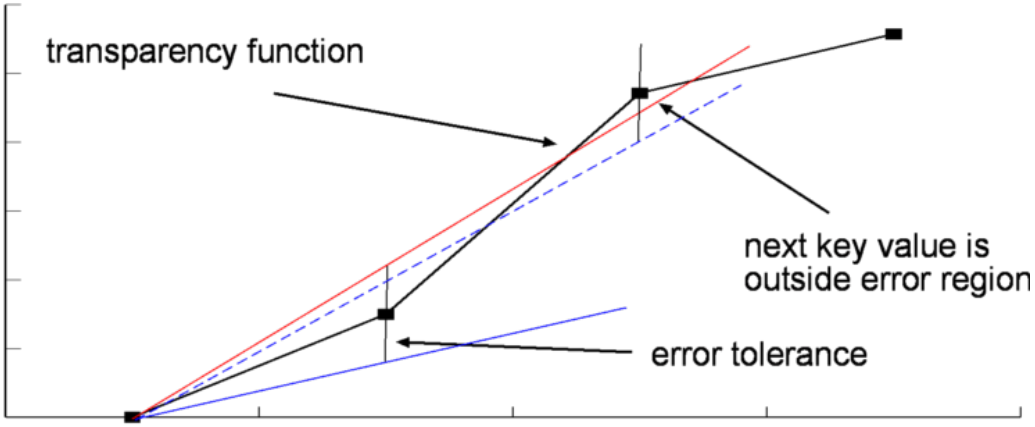


Deep Shadow Map

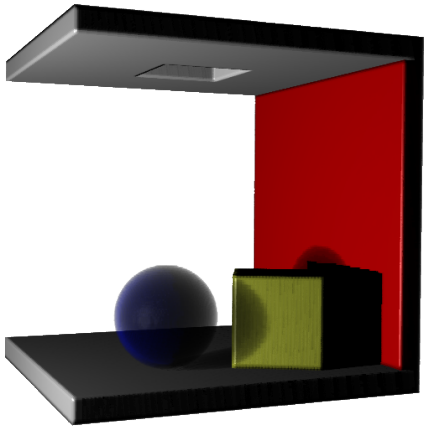
Construction of the DSM



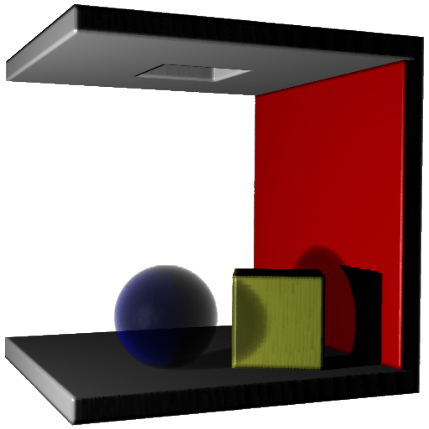
- save first key value at the first hit point
- error value defines constraining functions
- refinement of functions
- linear function differs from transparency function by this error value (maximum variation)



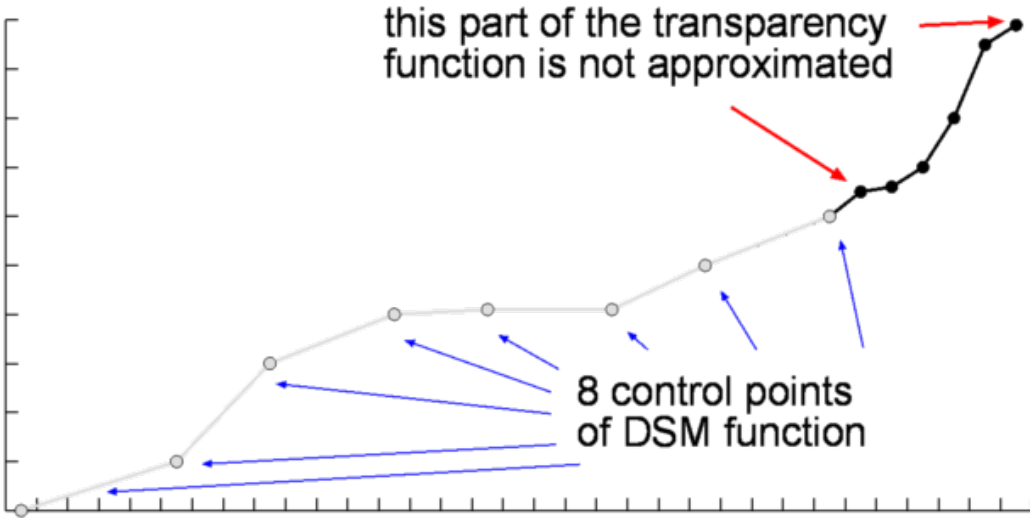
Deep Shadow Map



error value of 0.01



error value of 0.00005



Performance

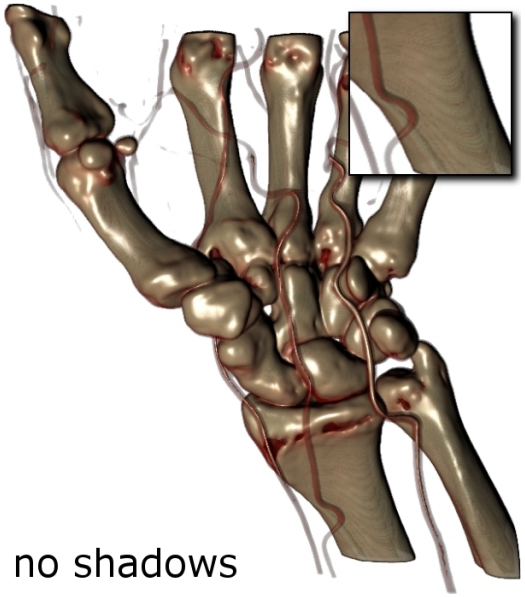
Results

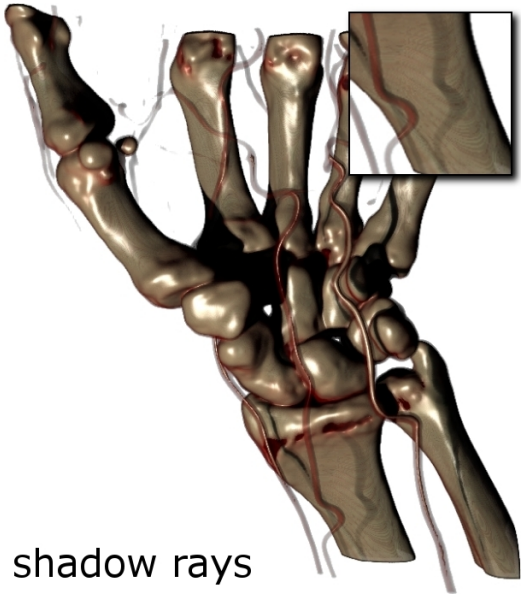
rendered on an Intel Core2 6600 (2.4GHz), 2GB RAM and an nVidia GeForce 8800GTX (values in fps)

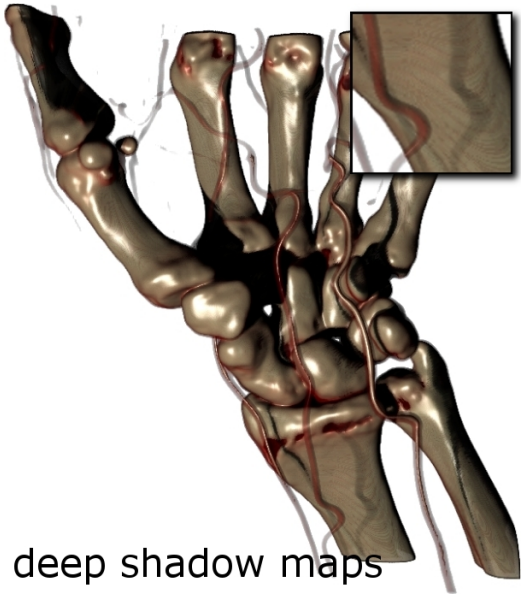
Shadow Mode	Recomputation?	
	yes	no
ShadowRay (B)	10.03	10.03
ShadowRay (A)	10.00	10.00
ShadowRay (B + PP)	5.59	46.00
ShadowMap (B)	29.08	45.50
DeepShadowMap (A)	15.76	34.50
DeepShadowMap (A + PP)	13.06	45.20

B = binary shadows (only opaque occluders)
A = shadowing includes semi-transparent occluders
PP = preprocessing

Visual Comparison



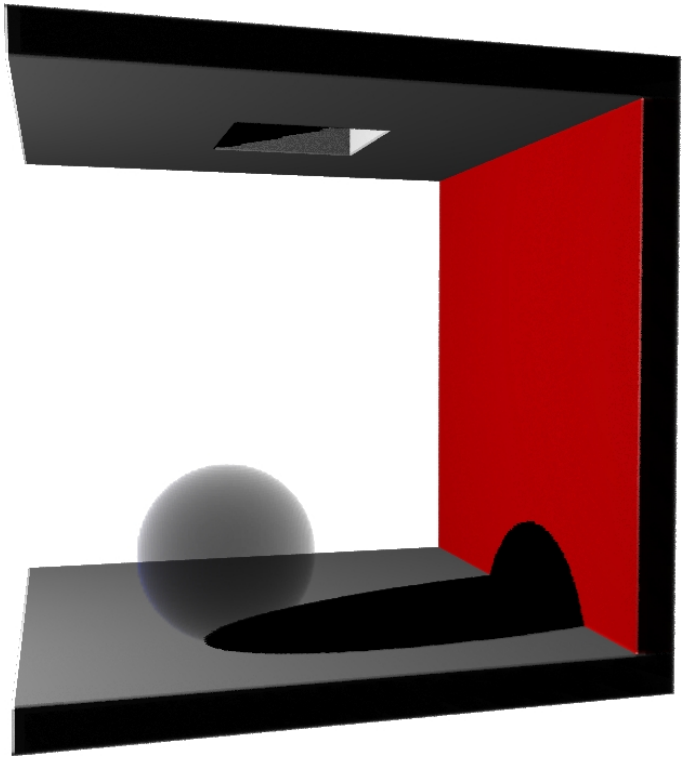




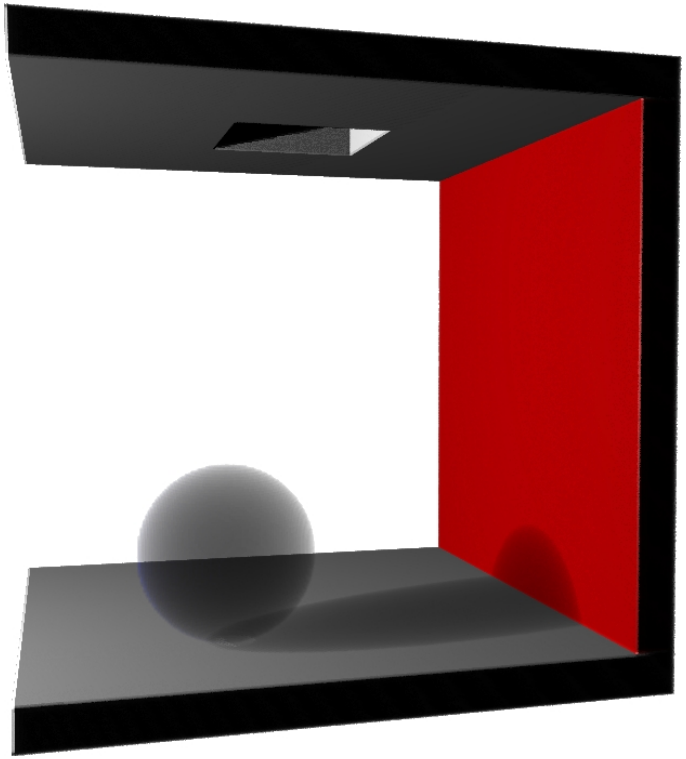
Visual Comparison

Interaction Video

Visual Comparison



Shadow Map



Deep Shadow Map

Conclusions

Results

- shadow rays technique is inefficient
- deep shadow map is a good compromise

Conclusions

Results

- shadow rays technique is inefficient
- deep shadow map is a good compromise

Future Work

- support of
  - soft shadows
  - colored shadows
- combination of shadow approaches

Questions?

http://www.voreen.org