

# Advanced Visualization of 3D Data

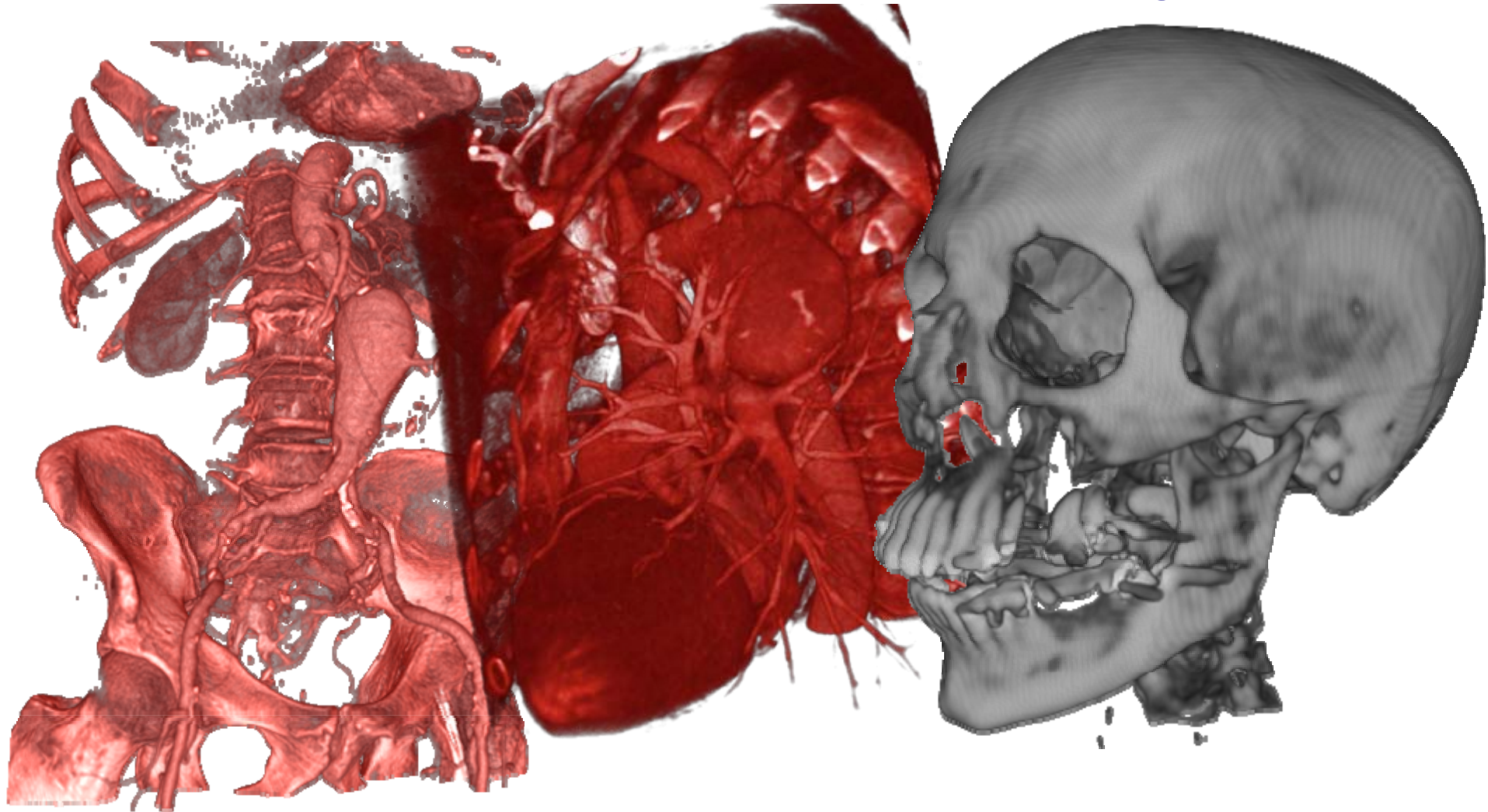
Oliver Kutter and Nassir Navab

12 July 2007

Chair for Computer Aided Medical Procedures & Augmented Reality

Department of Computer Science | Technische Universität München

# Motivation – 3D Vis of Medical ImageData



# Outline

- Volume Visualization Techniques
- Direct Volume Rendering Techniques
  - Theory
  - Volume Rendering Pipeline
  - Common Approaches
  - GPU accelerated Raycasting
  - Optimizing Performance and Quality

# Volume Visualization Techniques

- Polygonal Isosurface Rendering
  - Fast and easy to implement
  - Requires prior surface extraction → No online update
- Direct Volume Rendering
  - No prior extraction of surfaces
  - Various Possibilities for on-the-fly Classification, Shading
  - Direct online update
- Non Polygonal Isosurface Rendering
  - No prior preprocessing
  - Direct online update

# Direct Volume Rendering

- Render Volume without extraction of surfaces (DVR)
- Map scalar values to optical properties (Classification)
- Determine volume rendering integral for rays into the volume



# Volume Rendering Theory

# DVR Theory and Background Material

- Theory and Background: [http://www.cg.informatik.uni-siegen.de/data/Tutorials/EG2006/01\\_Theory.ppt](http://www.cg.informatik.uni-siegen.de/data/Tutorials/EG2006/01_Theory.ppt)
- GPU Programming Basics: [http://www.cg.informatik.uni-siegen.de/data/Tutorials/EG2006/02\\_GPU\\_Programming.ppt](http://www.cg.informatik.uni-siegen.de/data/Tutorials/EG2006/02_GPU_Programming.ppt)  
(only for the interested, not mandatory for the course)

# Volume Rendering Approaches

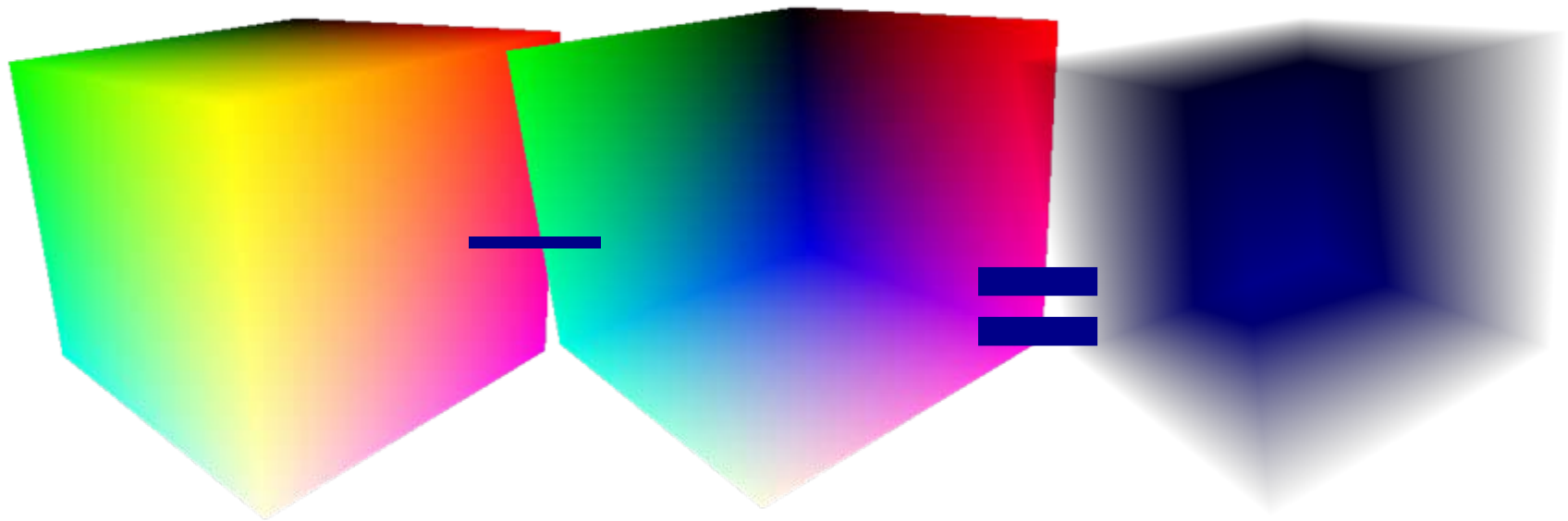
# DVR Approaches Material

- Texture-Based VR: [http://www.cg.informatik.uni-siegen.de/data/Tutorials/EG2006/03\\_VolumeRendering.ppt](http://www.cg.informatik.uni-siegen.de/data/Tutorials/EG2006/03_VolumeRendering.ppt)
- GPU Raycasting: [http://www.cg.informatik.uni-siegen.de/data/Tutorials/EG2006/04\\_Raycasting.ppt](http://www.cg.informatik.uni-siegen.de/data/Tutorials/EG2006/04_Raycasting.ppt)

# GPU Accelerated Raycasting

- Raycasting yields superior images over Proxy Geometry based DVR approaches
- Works correct for any perspective projection
- 32bit float precision rendering possible, when implemented in single loop in fragment shader
- Raycasting in SW is slow → exploit computational power of GPU for raycasting
- Easy to implement, highly optimizable and adaptable
- Real time capable

# GPU Accelerated Raycasting



**Front Faces =  
Ray Entries**

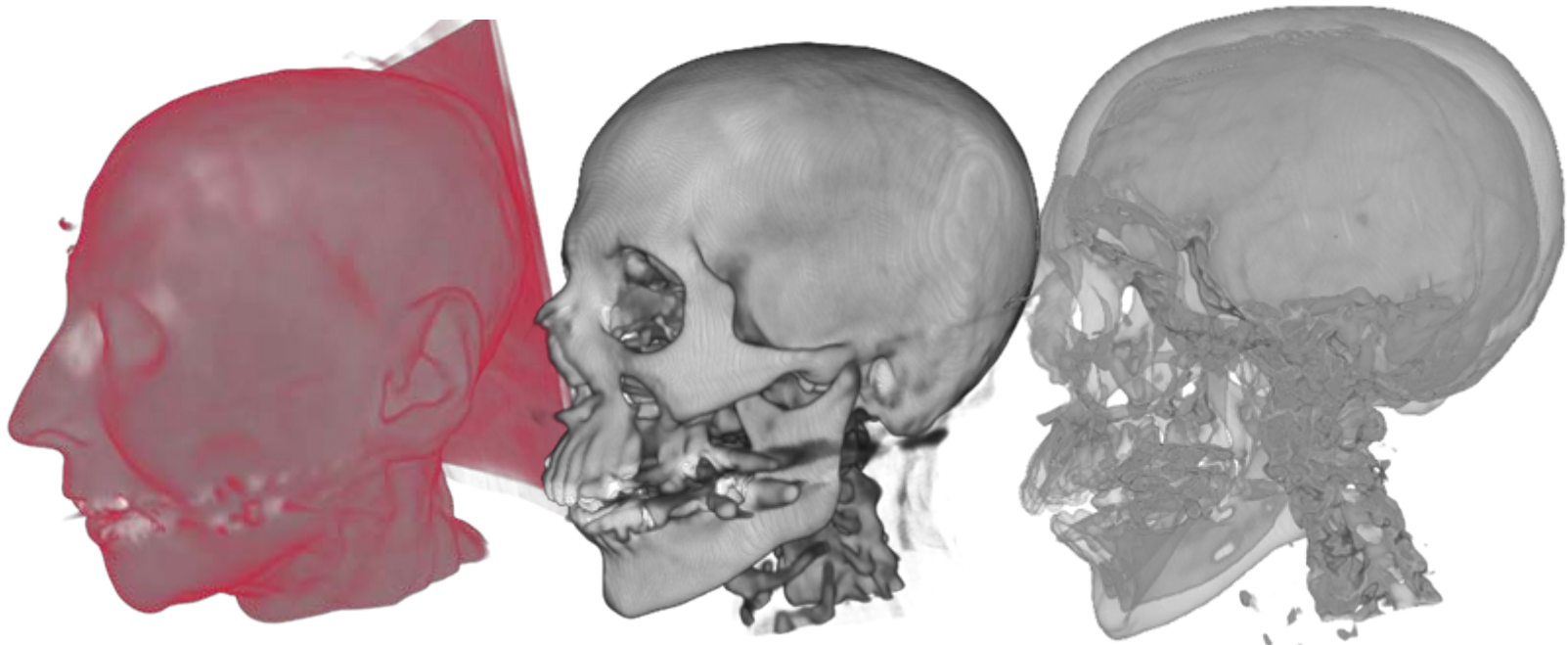
**Back Faces =  
Ray Exit**

**Ray Directions  
and Lengths**

```
1  uniform sampler1D  transfer_function;
2  uniform sampler2D  ray_start;
3  uniform sampler2D  ray_dirTex;
4  uniform sampler3D  volume;
5
6  uniform float      sample_distance;
7  uniform int        max_samples;
8
9  void main()
10 {
11     // lookup ray start
12     vec4 rayStart = texture2D(ray_start, gl_TexCoord[0].st);
13     // lookup ray end
14     vec4 ray = texture2D(ray_dirTex, gl_TexCoord[0].st);
15     vec3 rayDir = ray.xyz;
16     float rayLength = ray.w;
17     vec3 rayStep = normalize(rayDir) * sample_distance;
18     float rayStepLength = length(rayStep);
19     vec3 rayPos = rayStart.xyz;
20
21     float length_acc = 0.0;
22     vec4 dst = vec4(0.0);
23     vec4 src = vec4(0.0);
24
25     for(int i = 0; i < max_samples; ++i)
26     {
27         src = texture3D(volume, rayPos);
28         src.rgb *= src.a;
29         dst = (1.0 - dst.a) * src + dst;
30         rayPos += rayStep;
31         length_acc += rayStepLength;
32
33         if(length_acc > rayLength)
34             break;
35     }
36
37     gl_FragColor = dst;
38 }
```

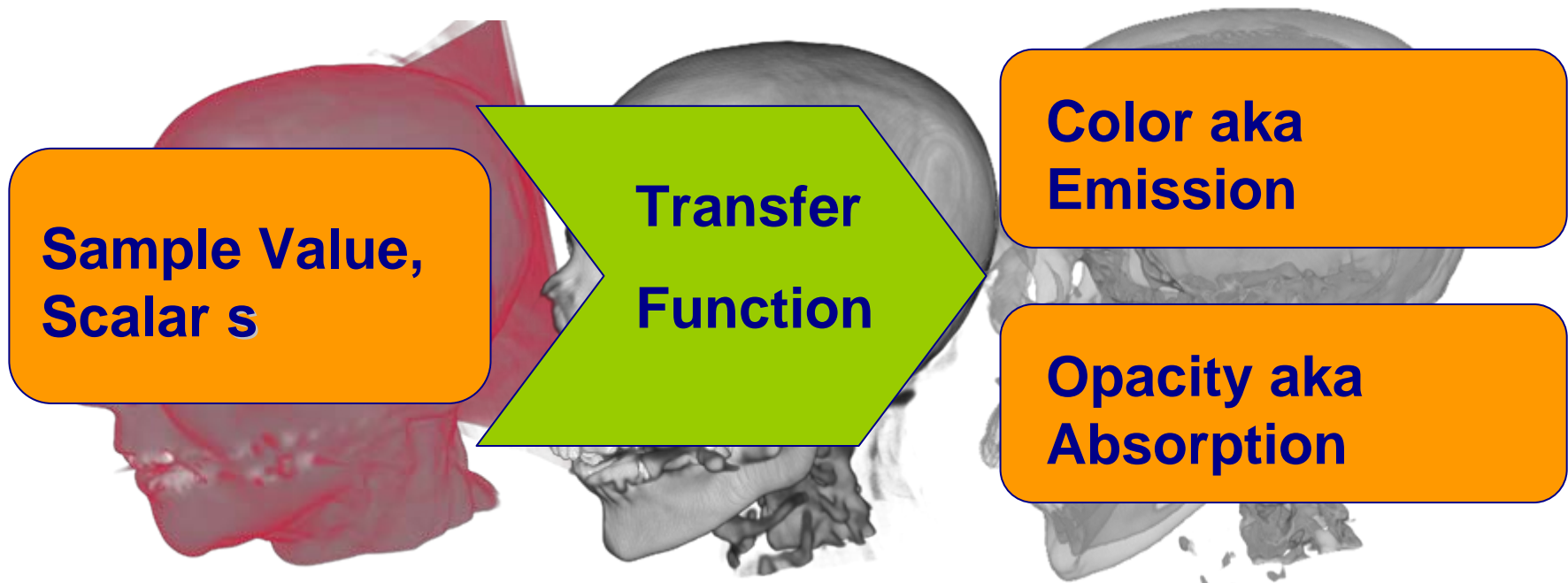
## Compute VR Integral in Fragment Shader in final pass

# Classification



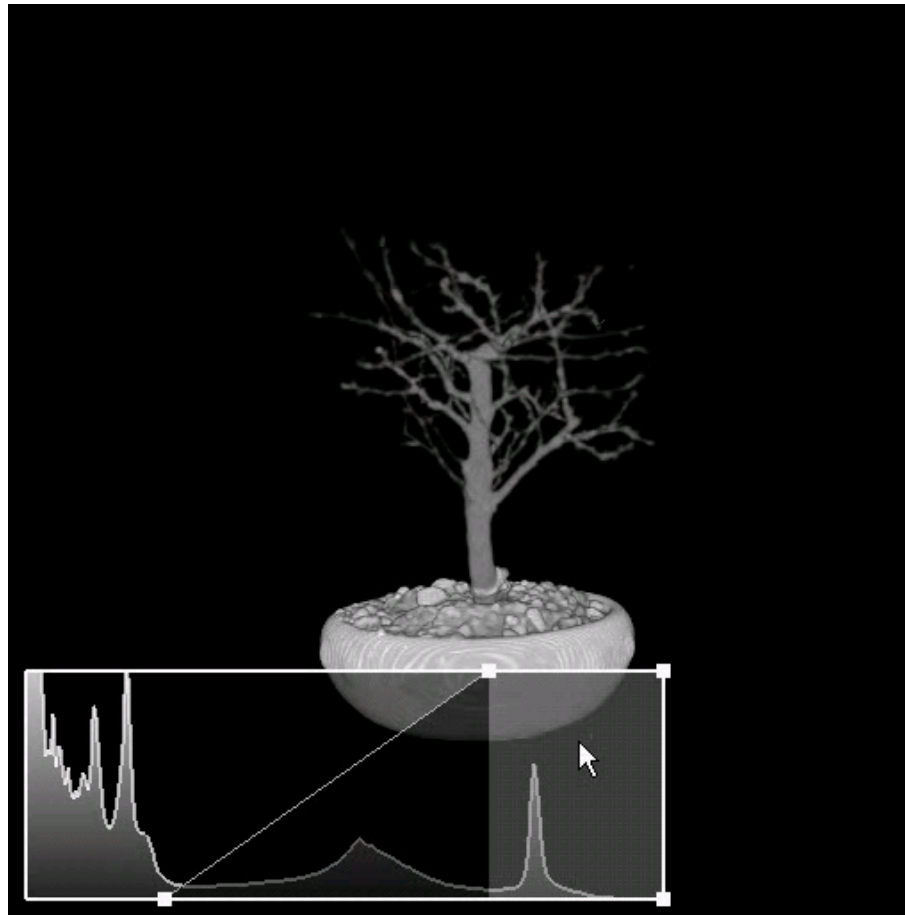
- By Classification the user defines „**Appearance**“ of the volume
  - Which parts are transparent?
  - Which colors are assigned to which parts of the volume?

# Classification

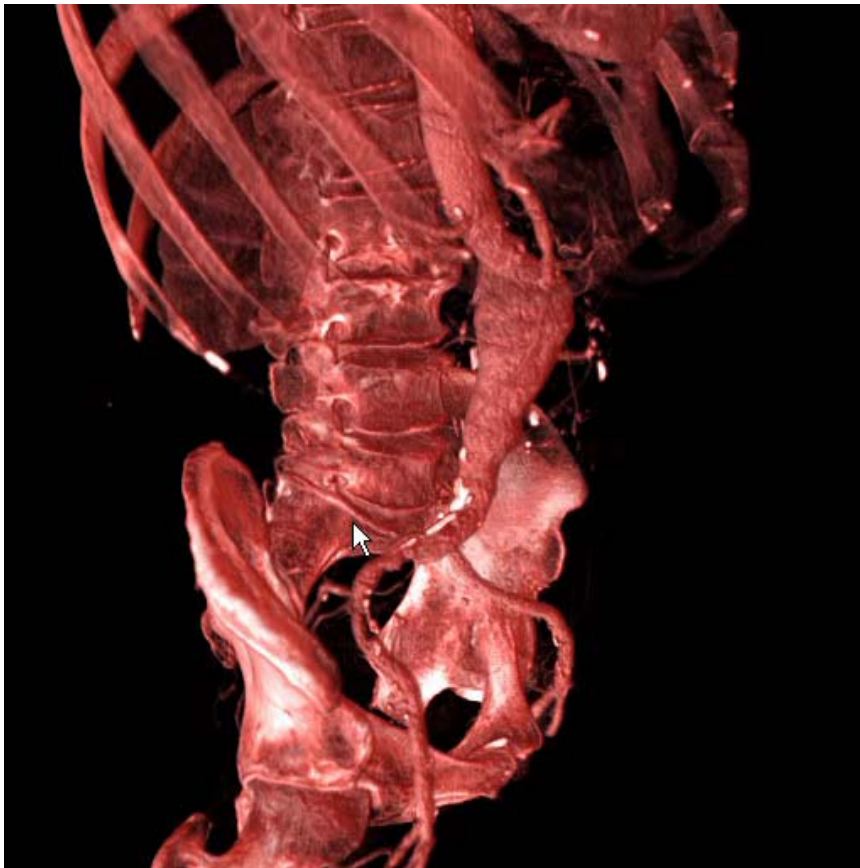


- By Classification the user defines „**Appearance**“ of the volume
  - Which parts are transparent?
  - Which colors are assigned to which parts of the volume?

# 1D TransferFunction Demo

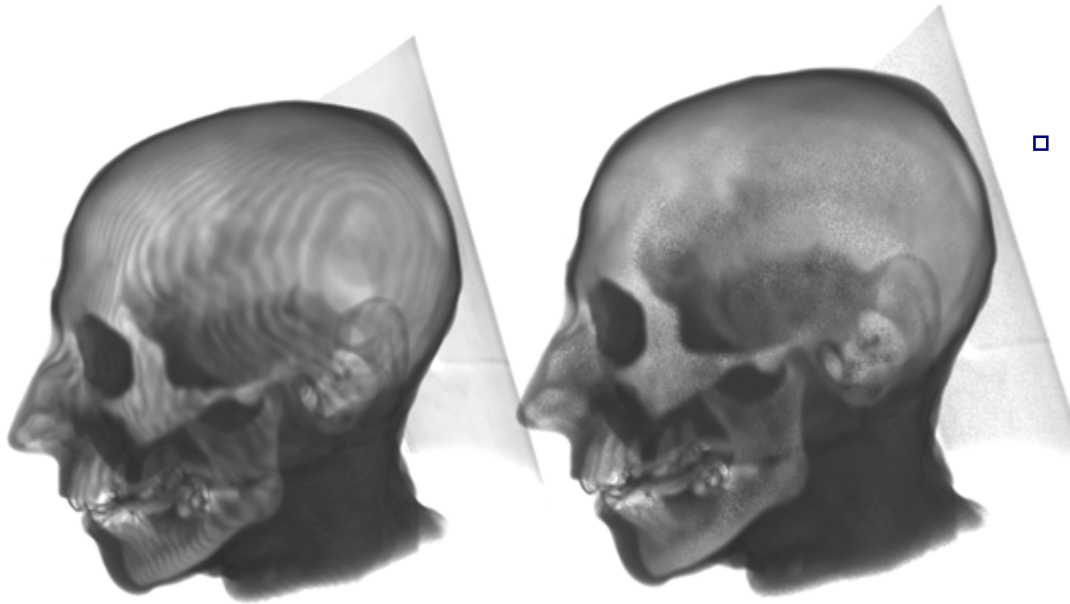


# Illumination for VR



- Why
  - Improved depth perception
  - Improved visualization of surface structures
- How
  - Local illumination by gradients and local illumination model e.g. Phong
  - Global illumination possible but not scope of this lecture

## Improving Quality



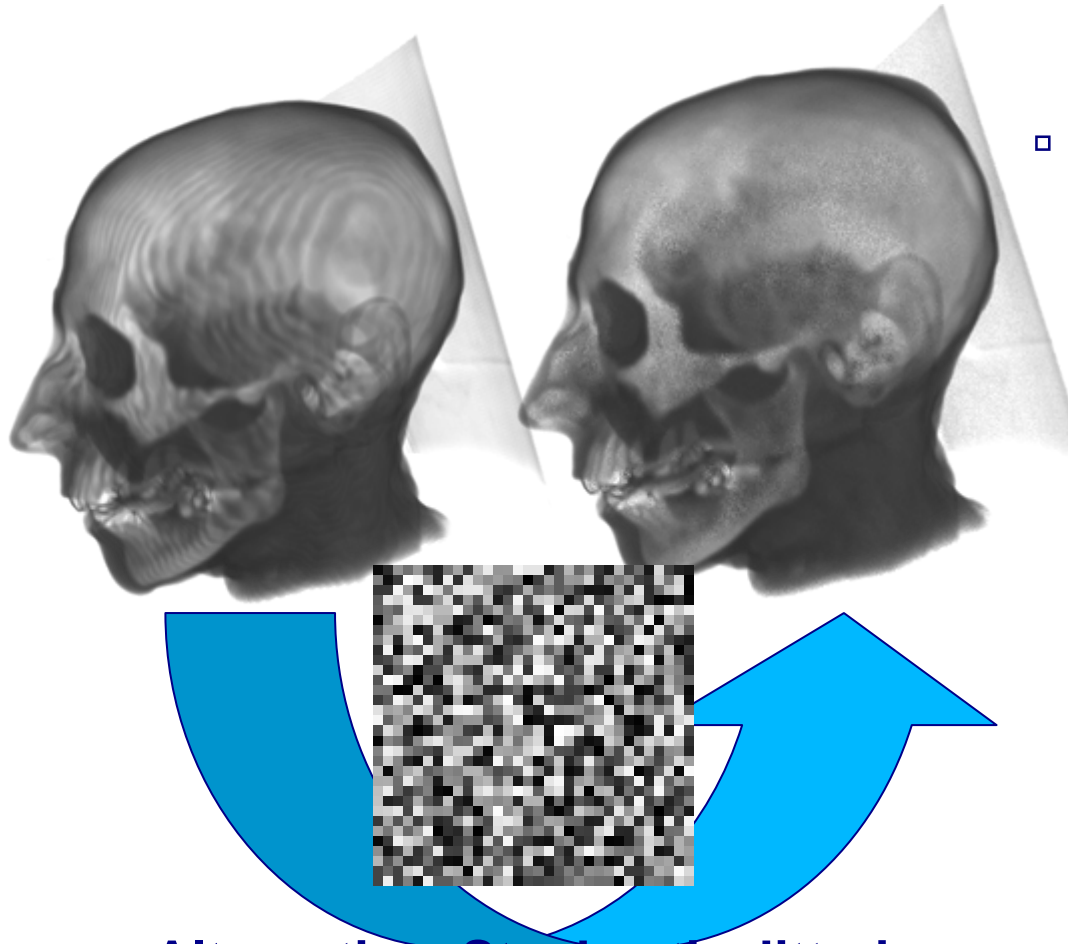
- Sources of Errors
  - Low sampling rate

**Increase Sampling Rate**



**Directly decreases performance**

# Improving Quality



- Sources of Errors
  - Low sampling rate

**Alternative: Stochastic Jittering**  
**Add noise to start positions**

# Improving Quality

**Use Higher Order Filters  
implemented in Shader on GPU**

**Use higher res texture formats,  
however this decreases available  
memory!**

- Sources of Errors
  - Low sampling rate
  - Filtering Artifacts

# Improving Quality

- Sources of Errors
  - Low sampling rate
  - Filtering Artifacts
  - Classification Artifacts

**Do not apply (pre-) classification of data before filtering, apply (post-) classification after filtering**

**Use Preintegrated Transfer Functions!**

# Improving Quality

**Compute Centralized Gradients on  
the fly**

**Use High Quality Gradient  
Computation Kernels**

- Sources of Errors
  - Low sampling rate
  - Filtering Artifacts
  - Classification Artifacts
  - Shading Artifacts

# Improving Quality

**Use full precision offscreen  
render targets**

**Blend in Fragment program with  
full GPU precision**

- Sources of Errors
  - Low sampling rate
  - Filtering Artifacts
  - Classification Artifacts
  - Shading Artifacts
  - Blending Artifacts

# Improving Performance

Render only visible parts of volume!

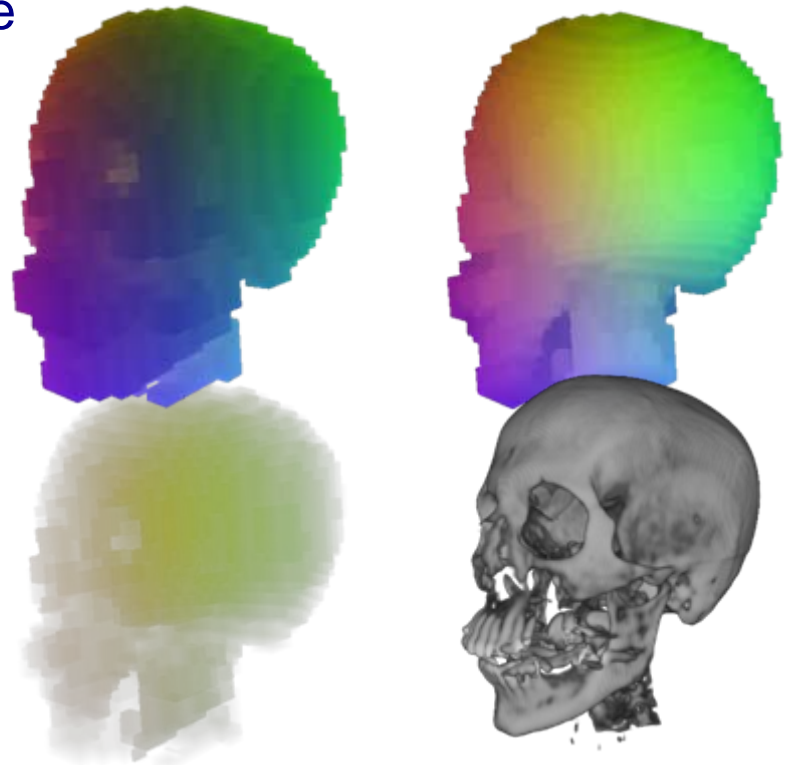
- Cull against TransferFunction, IsoValue
- Cull against View Frustum
- Use Octrees, Bounding Volumes
- Leap over Empty Space in Volume

Stop rendering as early as possible!

- Early ray termination

Be as lazy as possible!

- Employ Deferred Shading Techniques



# Visualization Modes Examples

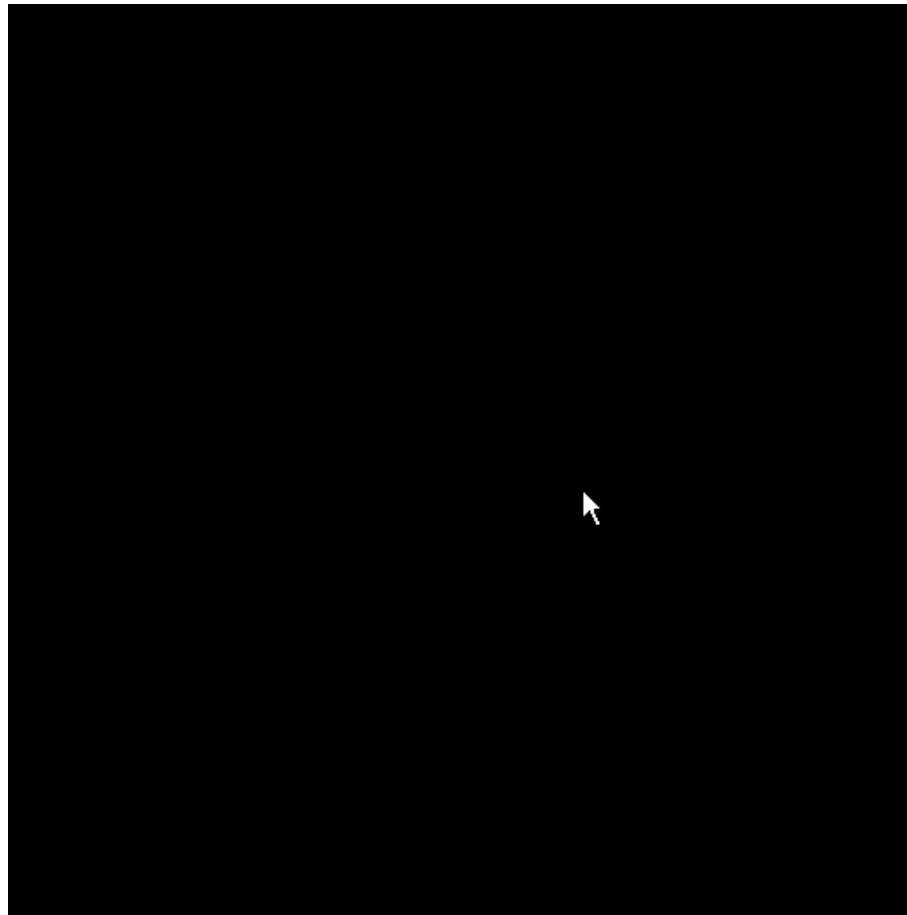
# Maximum Intensity Projection (MIP)



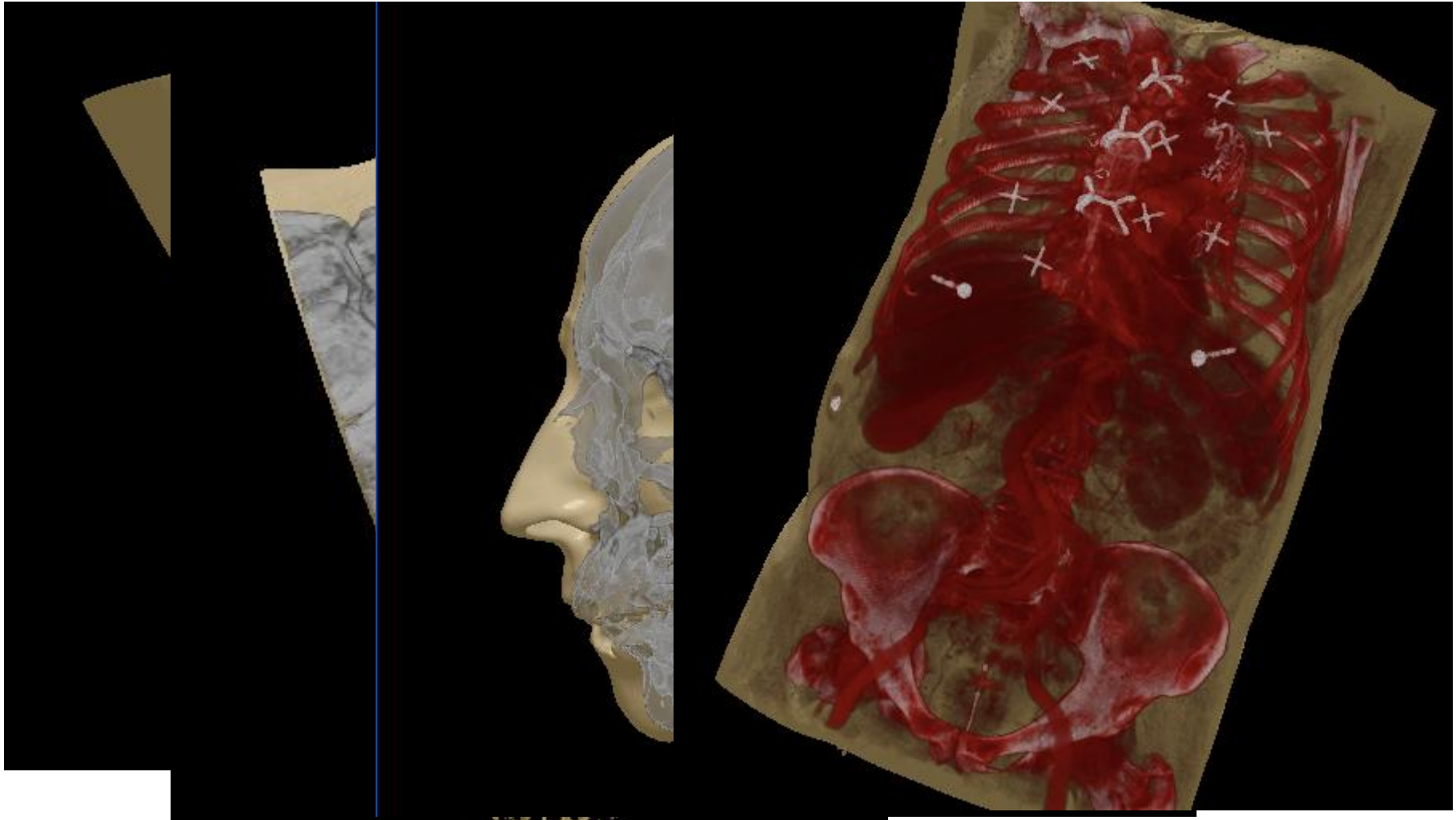
# Digitally Reconstructed Radiographs



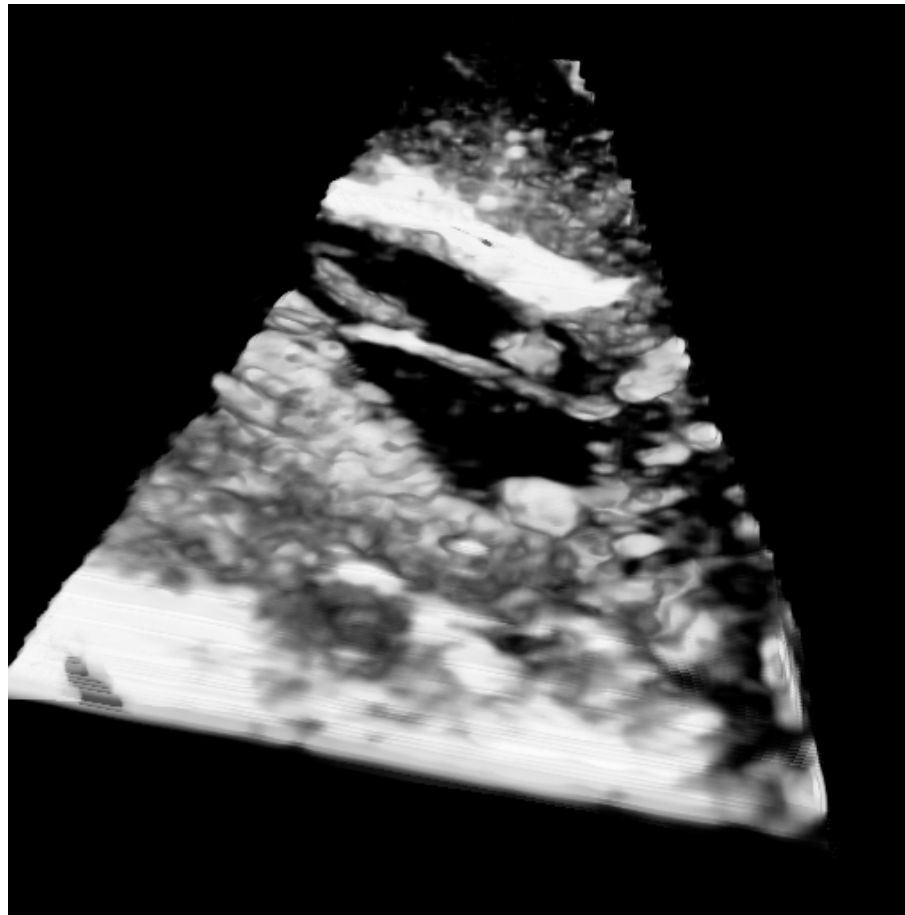
# Non Polygonal Isosurfaces



# Hybrid DVR



# Visualization of 4D Datasets



# References

For specific papers please refer to the references given in

- <http://www.real-time-volume-graphics.org>
- <http://www.cg.in.tum.de>
- [http://www.vrvis.at/research/publications\\_de.html](http://www.vrvis.at/research/publications_de.html)
- [http://www.vis.uni-stuttgart.de/vis03\\_tutorial/](http://www.vis.uni-stuttgart.de/vis03_tutorial/)
- <http://www.vis.uni-stuttgart.de/eng/research/fields/current/spvolren/>
- <http://www.cs.utah.edu/~jmk/simian/>

# Acknowledgements

Dr.-Ing. Christof Rezk-Salama

Computer Graphics and Multimedia Group,

University of Siegen, Germany

<http://www.cg.informatik.uni-siegen.de/People/Rezk>

# Open Student Projects / Open Hiwi Positions



Interested?

**mailto:** [kutter@in.tum.de](mailto:kutter@in.tum.de)

Thanks for your attention!