Visual Medicine: Part Two – Advanced Topics in Visual Medicine

Fast Tagged Multi-resolution Volume Rendering

Florian Link

MeVis Research
Bremen, Germany

link@mevis.de

Overview

- Challenges
- Multi-resolution Rendering
- Slicing vs. RayCasting
- Per Tag Shading
- Transparent Geometry in Volume Rendering

Challenges

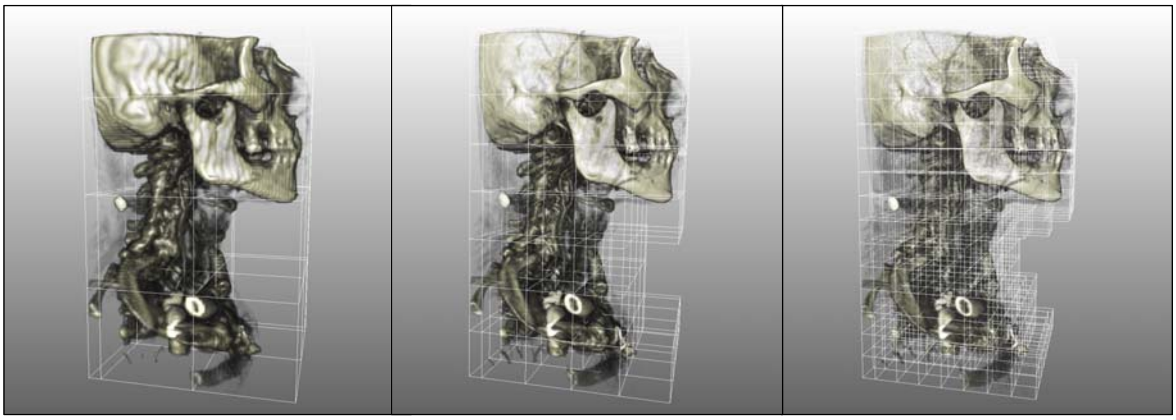
- medical datasets are getting huge
- e.g. CT 512 x 512 x 2000 Slices (12Bit) is normal as of today
- the user wants Volume Rendering without waiting
- labeling/tagging of objects in the volume data helps highlighting important structures, but adds an additional (potentially large) tag volume

Octree Approach

- Create an Octree structure [LaMar 1999]
- Node size e.g. 32x32x32 or 64x64x64
- Resampling of parent nodes
  - Rank filter
  - Average filter (for RGBA data)
- Each parent node contains the data of its 8 children in half resolution
- Cache file (approx 1.4 x original dataset)
- Trilinear interpolation requires 1 voxel overlap
- On-the-fly gradients require 3 voxel overlap

Octree Rendering

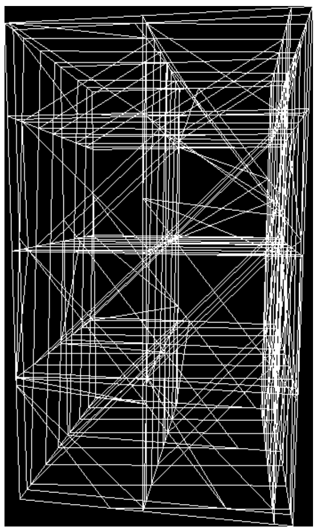
- Octree nodes are rendered back-to-front
- Node rendering order defined by viewing direction [Fang1996]
- Use different resolutions for interactive/static rendering



Node Slicing

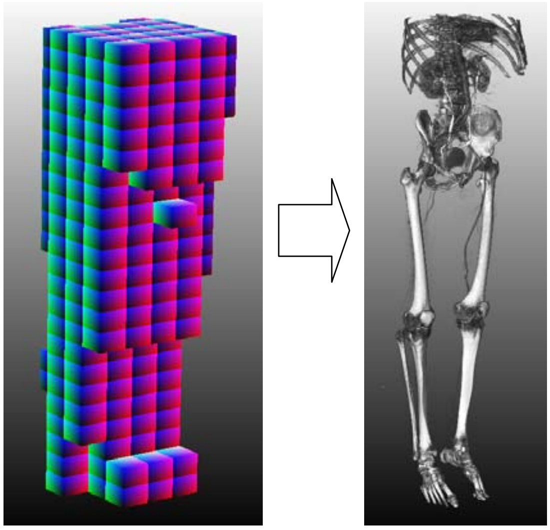
- Load nodes as 3D Textures
- Render view-aligned slices (proxy geometry) for each node [LiShen2002]
- Fast node slicing is required
  - Marching cube slicing [Ben2005]

  or
  - Slicing with Vertex Program [Salama2005]



Node Ray Casting

- Alternative to Slicing:
  Individual nodes are rendered with GPU raycasting, blended to final image
  - Requires only 6 faces to be rendered per node instead of multiple slices
  - Single pass ray casting avoids FBO switch for each node [Stegmaier2005]
  - Other approach: All nodes are stored in one large texture, requires additional texture lookup [Hadwiger2005]

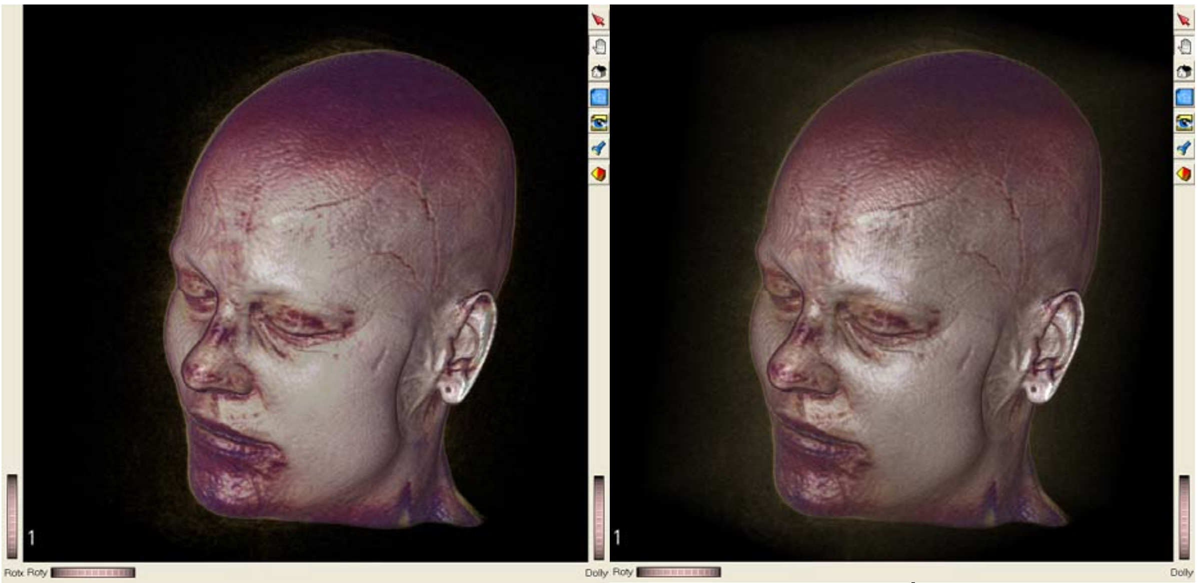


Node Ray Casting

- Advantage:
  - better image quality because of floating point blending
  - correct endoscopic rendering
  - straight-forward to implement [Stegmaier2005]
- Disadvantage:
  - requires Shader Model 3.0 hardware
  - and is still slower than traditional slicing

Slicing vs. RayCasting

- Example with transfer function that has highly transparent regions

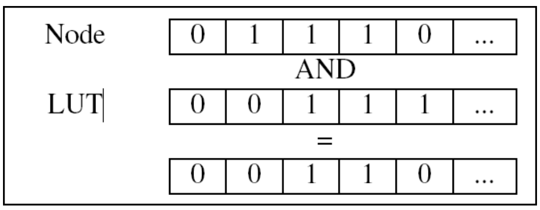


Slicing (8bit blending) RayCasting

Culling

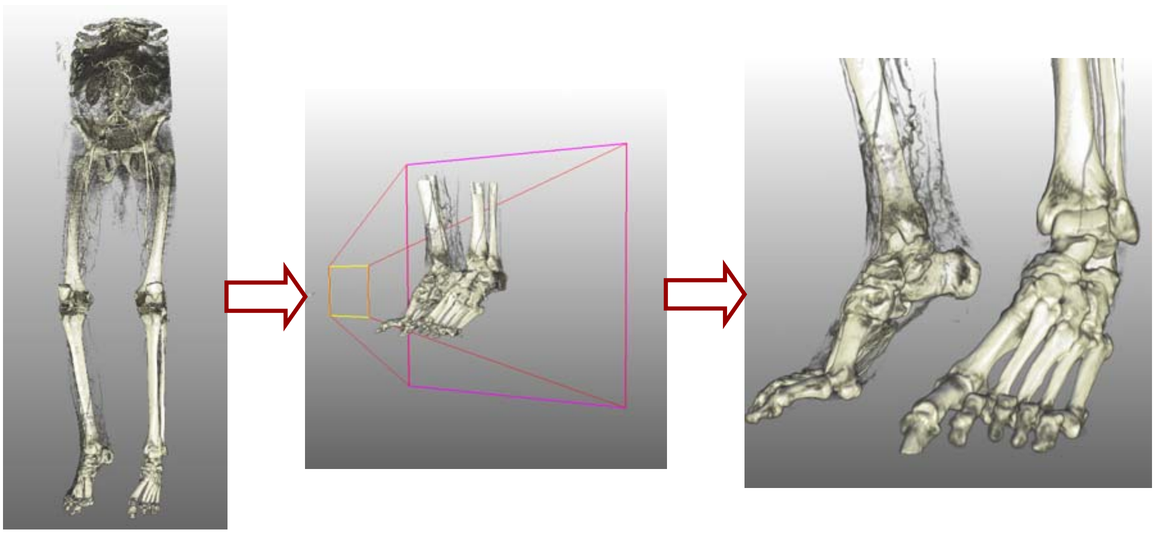
Clips nodes to:

- Selected Sub Volume
- Clipping planes
- Camera Frustum
- Visibility



  - Combine binary histogram of each node with histogram of current LUT, result != 0 means node is visible

Culling Example



Caching

Volume Data Cache

- Hierarchical Cache based on node height
- Uses MRU strategy

Texture Memory Cache

- Manages amount of texture memory used
- Uses MRU strategy

All Volume Renderer modules share the caches

Avoids that multiple renderers use too much resources

Shading

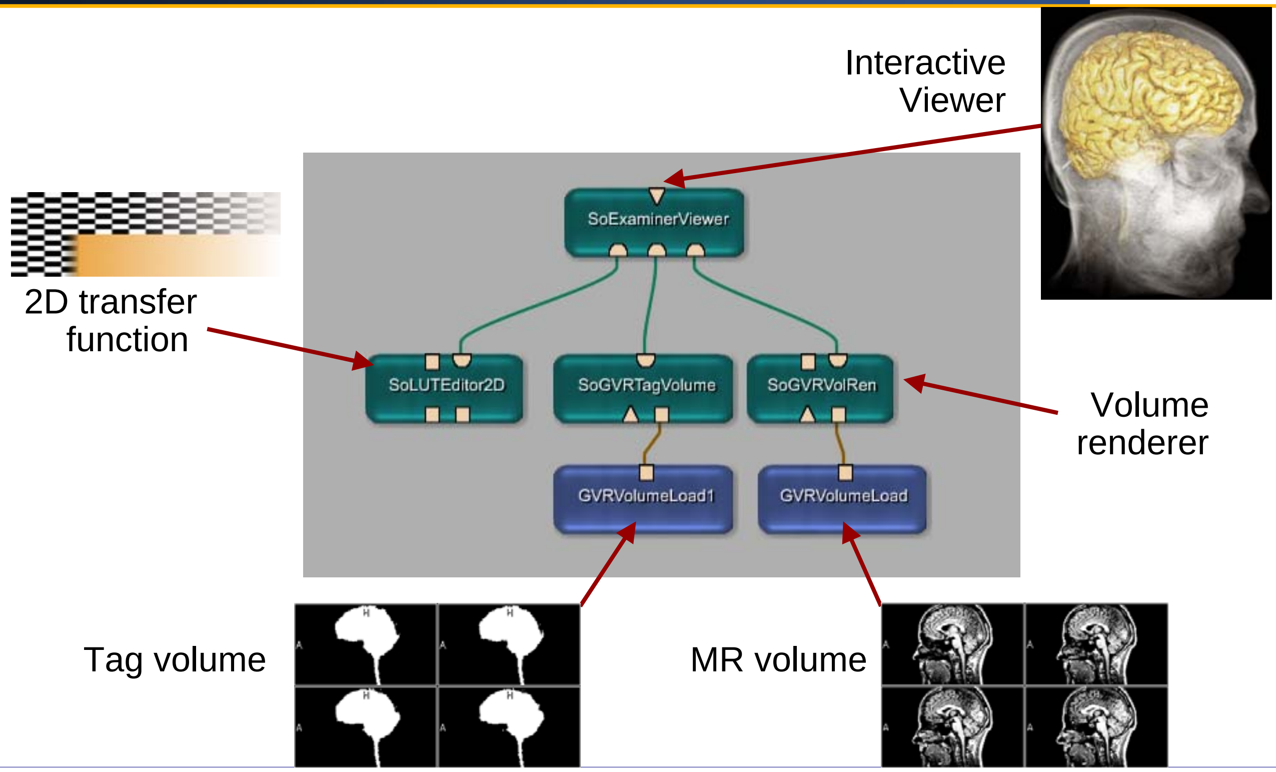
- Blinn-Phong Lights
- Tone Shading [Gooch1998]
- Silhouette/Boundary Enhancement [Ebert2000]
- Shaders written in GL Shading Language (GLSL) or ARB_FRAGMENT_PROGRAM
- Gradients either precalculated as RGBA texture or on-the-fly central difference in shader

Tag Volume

Defines an additional (sub)volume

- Same voxel size as original volume
- Tag volume labels segmented objects
- Tag range is 0-255
- per-tag LUTs [Kniss2001] and per-tag shading
- Optimization: Tag Volume Octree can use shared node images for nodes that contain only a single tag value (very efficient for sparse tag volumes)

MeVisLab Example Network



Per Tag Shading

- In addition to a per tag transfer function, select different shading modes for each individual tag
- Original idea by [Hadwiger2003], applied to multi-resolution rendering
- Use 1D RGBA texture to store 4 weights for each tag [Link2006]
  - Light1
  - Light2
  - Tone Shading
  - Boundary/Silhouette enhancement

Per Tag Shading

- Optimization: Avoid branching in the shaders and select specialized shader on each octree nodes, using binary histograms of tag nodes [Link2006]

Id	0	1	2	3	4	...
Per object rendering mode weights texture T_E						
R	1	0	0	1	0	...
G	0	0	1	1	0	...
B	0	.5	1	1	0	...
A	0	1	1	1	0	...
Node histogram S_n						
	0	1	0	0	0	...
Rendering mode histograms E						
R	1	0	0	1	0	...
G	0	0	1	1	0	...
B	0	1	1	1	0	...
A	0	1	1	1	0	...

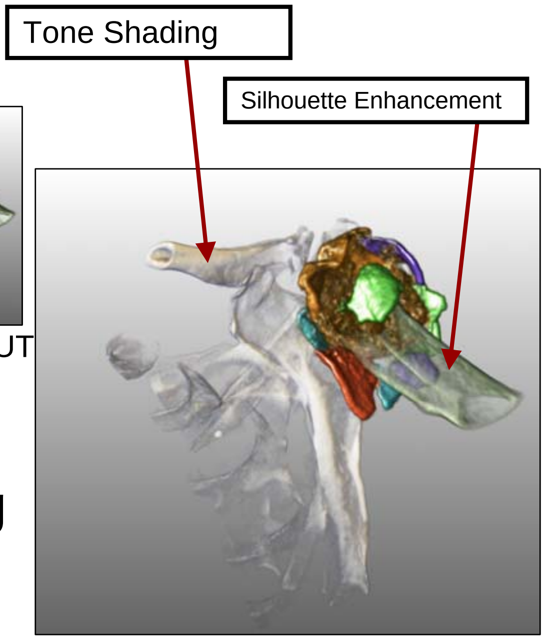
```
shaderID = 0;
for (i = 0; i < 4; i++)
  if (S_n & E[i] != 0)
    shaderID += 1 << i;
```


Per Tag Shading example

DVR

DVR + 2D LUT

Shaded + 2D LUT



Per Tag Shading

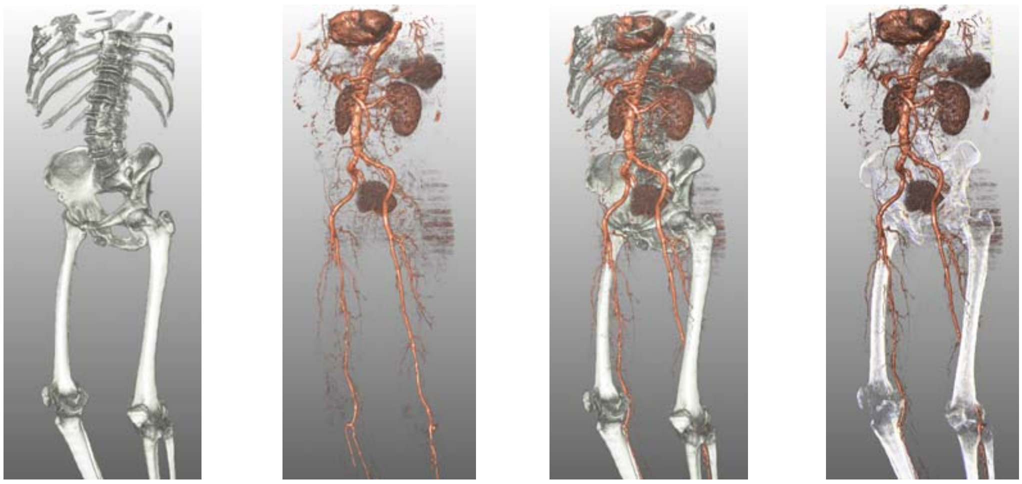
Mask Volume

- Defines an additional volume (no octree!)
- Binary inside/outside decision
- May be positioned/oriented anywhere in the volume
- Affects color and/or tag value
- Fast update, but size limited by texture memory, because it is loaded into one 3D Texture

Bone Removal with Mask Volume

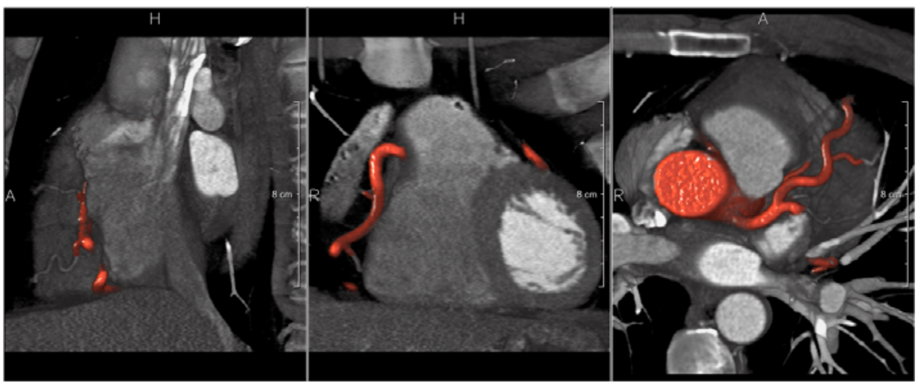
- Bone Removal outputs 256x256x256 mask volume when e.g. segmenting a 512x512x2000 slices dataset [Hahn2005]
- Mask volume is scaled to original volume while rendering
- Reduced mask resolution compared to original image creates artifacts, but allows for interactive segmentation updates

Bone Removal Example



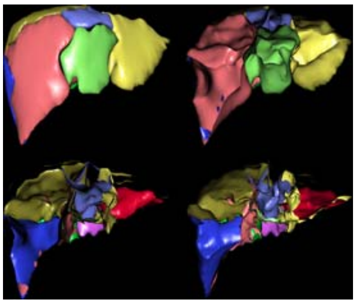
Slab Rendering

- Extends 2D Viewers to show Volume Rendering
- Render slab of slices with orthogonal projection
- All presented render modes are applicable

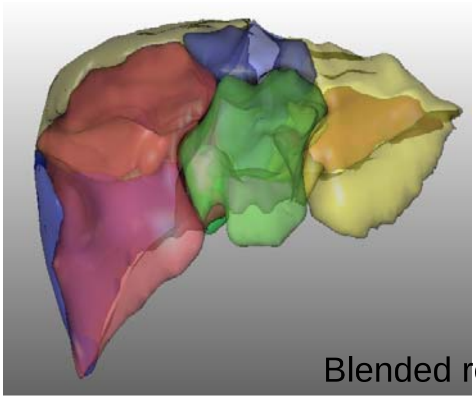


Depth Peeling

- Order Independent Transparency [Everitt2001]
- Image space algorith on GPU
- Emulates dual depth buffer tests
- Replaces traditional triangle sorting on CPU
- Multipass rendering of scene



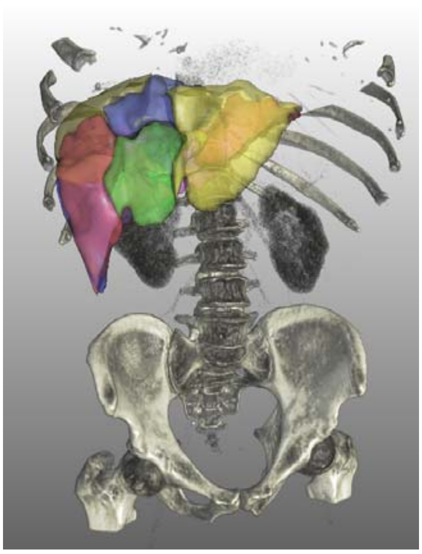
Layers



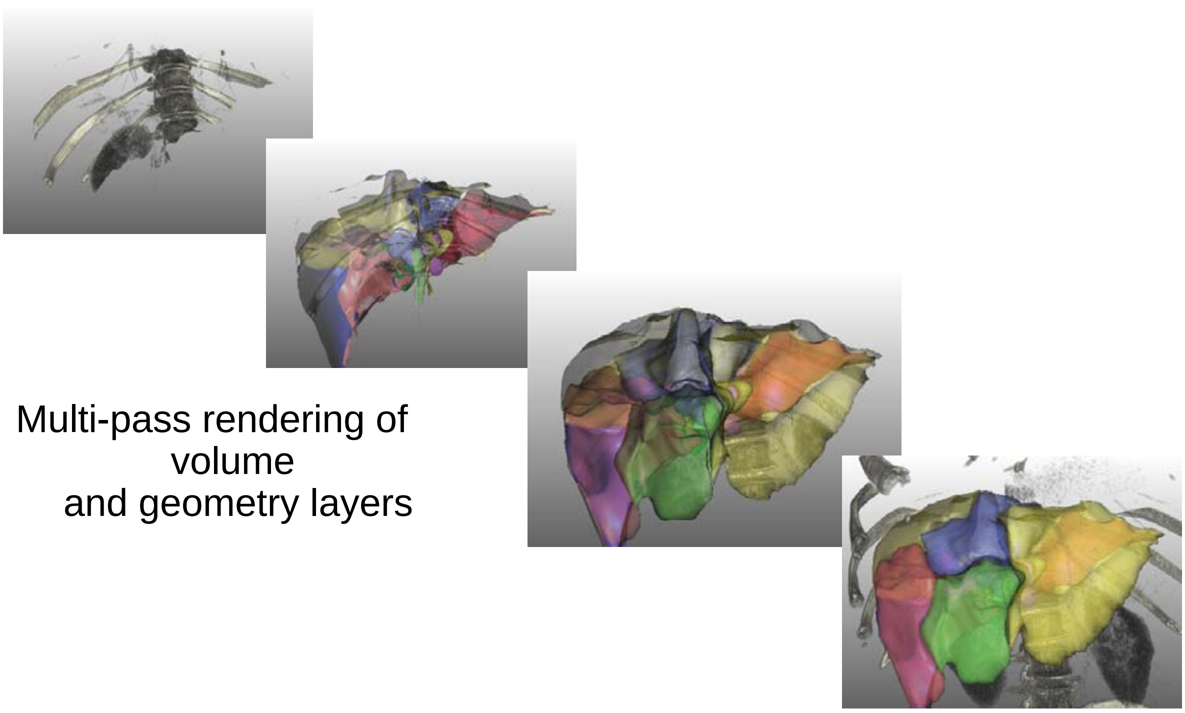
Blended result image

Transparent geometry and Volume Rendering

- Do depth peeling on geometry and keep color and depth layers as textures
- Mixes peeled color layers and volume rendering
- Multipass volume rendering for depth layers
- Can also use depth layers to do depth clipping or modify color/tag values [WEE02] [WEE03]



Transparent geometry and Volume Rendering



Multi-pass rendering of volume and geometry layers

Summary

- Modern graphics hardware allows advanced shading at interactive frame rates
- Presenting segmented objects becomes more important
- Most presented rendering modes are integrated in the **SoGVRVolRen** module and it's extensions, which is part of **MeVisLab Basic**
- Download MeVisLab at www.mevislab.de and check the examples of **SoGVRVolRen**

Acknowledgements

I would like to thank the following people for their contributions:

H. Hahn, T. Boskamp, O. Konrad, F. Ritter (MeVis)

Thanks to Markus Hadwiger (VRVis) for the nice discussions!

References

[Ben2005] A. Benassarou, E. Bittar, N. W. John, and L. Lucas. MC slicing for volume rendering applications. In International Conference on Computational Science (2), 2005.

[Salama2005] C.Rezk-Salama and A.Kolb. Vertex program for efficient box-plane intersection. In Proc. Vision Modeling and Visualization 2005, 2005.

[Ebert2000] D. Ebert and P. Rheingans. Volume illustration: non-photorealistic rendering of volume models. In Proc. IEEE Visualization 2000, 2000.

[Gooch1998] A. Gooch, B. Gooch, P. Shirley, and E. Cohen. A non-photorealistic lighting model for automatic technical illustration. Computer Graphics, 32(Annual Conference Series):447–452,1998.

References

[Grimm2004] S. Grimm, S. Bruckner, A. Kanitsar, and E. Groeller. A refined data addressing and processing scheme to accelerate volume raycasting. Computers and Graphics, 27:719–729, 2004.

[Guthe2004] S. Guthe and W. Strasser. Advanced techniques for high-quality multi-resolution volume rendering. Computers & Graphics, 28:51–58, 2004.

[Hadwiger2003] M. Hadwiger, C. Berger, and H. Hauser. Highquality two-level volume rendering of segmented data sets on consumer graphics hardware. In Proc. IEEE Visualization 2003, 2003.

[Hadwiger2005] M. Hadwiger, C. Sigg, H. Scharsach, K. Bühler, and M. Gross. Real-time raycasting and advanced shading of discreate isosurfaces. In Proc. Eurographics, 2005.

[Hahn2005] H.K. Hahn. Morphological Volumetry – Theory, Concepts, and Application to Quantitative Medical Imaging. PhD thesis, University of Bremen, 2005.

[Higue2005] F. Vega Higuera, P. Hastreiter, R. Naraghi, R. Fahlbsuch, and G. Greiner. Smooth volume rendering of labeled medical data on consumer graphics hardware. In Proc. SPIE Medical Imaging 2005, 2005.

[Kindl1998] G. Kindlmann and J. Durkin. Semi-automatic generation of transfer functions for direct volume rendering. In Proc. IEEE Volume Visualization 1998, 1998.

References

[Kniss2001] J. Kniss, G. Kindlmann, and C. Hansen. Multi-dimensional transfer functions for interactive volume rendering. In Proc. IEEE Visualization 2001, 2001.

[LaMar1999] E. LaMar, B. Hamann, and K.I. Joy. Multiresolution techniques for interactive texturebased volume visualization. In Proc. IEEE Visualization 1999, 1999.

[LiShen2002] X. Li and H.-W. Shen. Time-critical multiresolution volume rendering using 3d texture mapping hardware. In Proc. Volume Vis. 2002, 2002.

[Weiler2000] M. Weiler, R. Westermann, C. Hansen, K. Zimmerman, and T. Ertl. Level–of–detail volume rendering via 3d textures. In Proc. Volume Visualization 2000, 2000.

[You2006] H. Younesy, T.Möller, and H. Carr. Improving the quality of multi-resolution volume rendering. In Proc. Eurographics/IEEE-VGTC Symposium on Visualization 2006, 2006.

[Fang1996] S Fang, R Srinivasan, S Huang, R Raghavan. Deformable Volume Rendering by 3D Texture Mapping and Octree Encoding. In Proc of the 7th conference on Visualization, 1996

References

[Stegmaier2005] S. Stegmaier, M. Strengert, T. Klein, T.Ertl: A Simple and Flexible Volume Rendering Framework for Graphics-Hardware-based Raycasting. Volume Graphics 2005

[Everitt2001] Cass Everitt, Interactive Order-Independent Transparency

http://developer.nvidia.com/object/Interactive_Order_Transparency.html

[WEE02] D. Weiskopf, K. Engel, and T. Ertl.: Volume Clipping via Per-Fragment Operations in Texture-Based Volume Visualization, IEEE Visualization 2002 Proceedings, R. Moorhead, M. Gross, K. I. Joy (eds.), ACM Press, October 2002, 93-100.

[WEE03] D. Weiskopf, K. Engel, and T. Ertl.: Interactive Clipping Techniques for Texture-Based Volume Visualization and Volume Shading. IEEE Transactions on Visualization and Computer Graphics, 2003.

[Link2006] F. Link, M.König, H-O. Peitgen, Multi-Resolution Volume Rendering with per Object Shading. To appear in VMV Proceedings 2006