GPU-Based Volume Ray-Casting with Advanced Illumination

Timo Ropinski*
Visualization and Computer Graphics Research Group, University of Münster

Christof Rezk-Salama†
Computer Graphics Group, University of Siegen

Markus Hadwiger‡
VRVis Research Center for Virtual Reality and Visualization

Patric Ljung§
Department of Imaging and Visualization, Siemens Corporate Research

Abstract

GPU-based ray-casting techniques are becoming more and more important for the visualization of volume data in medicine and engineering. Thanks to their flexibility and accuracy, they will likely replace existing slice-based techniques in the near future. This tutorial targets the growing number of developers and scientific researchers who work with specialized volume visualization algorithms on state-of-the-art graphics hardware.

Starting with a brief introduction to the concepts behind GPU-based ray-casting, we will review existing techniques capable to accelerate the rendering performance. These acceleration techniques are the key issue for supporting advanced illumination models, since these models usually consume more rendering time. In contrast to commonly used local illumination models, advanced illumination models allow to incorporate the light interactions between neighboring structures. Such effects include soft and hard shadows as well as translucency and multiple scattering. The tutorial focuses strongly upon those effects, which support improved spatial comprehension and are thus relevant for scientific visualization from a perceptual point of view, but it also covers topics more related to visual arts.

Index Terms: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Subjects: Color, shading, shadowing, and texture

1 Tutorial Duration

1/2 day tutorial.

2 Level of the Tutorial

Intermediate to advanced. The tutorial is aimed at scientific researchers and developers of visualization tools. Course participants should have basic programming skills and should be familiar with graphics hardware and shading languages. We will assume a basic knowledge regarding volume data as well as interactive volume rendering techniques. Furthermore, a basic understanding of GPU-based rendering techniques is required.

3 Tutorial Website

Updated versions of the slides and the tutorial notes can be found at the following website: `http://www.voreen.org/vis08-tutorial`

4 Tutorial Description

We will present an in-depth tutorial on GPU-based ray-casting approaches to volume visualization, including acceleration techniques, efficient memory management and advanced illumination models. Participants will learn how to leverage the new features of modern commodity graphics hardware to implement advanced illumination techniques for high-quality volume rendering applications supporting improved spatial comprehension. Thus they should be able to target new research directions towards visualization techniques exploiting novel shading models as well as to improve the existing advanced shading models.

The tutorial starts with an introduction to the basic principles of GPU-based volume ray-casting, including useful optimization techniques highly benefitial for advanced illumination models. After these areas have been covered existing interactive illumination models for GPU-based volume ray-casting are explained in detail, by considering their contribution to the image understanding process.

Thus, besides the basic implementation details, the following topics are covered:

- Memory Management,
- Space Leaping and Early Ray Termination,
- Soft and Hard Shadows, Semi-Transparent Shadows,
- Diffuse and Specular Light Interactions,
- Translucency and Multiple Scattering Effects, and
- Monte-Carlo based Illumination Techniques.

Participants are provided with implementation details often omitted in scientific publications and should thus be able to conduct their own research in this area in the near future. We feel that it is important to also demonstrate these ideas since the reality of implementation can be quite different from the theory presented in the scientific literature. Since the content of the tutorial can be quite challenging, we intend to answer questions on the fly.

5 Instructors Background and Contact Information

Markus Hadwiger
VRVis Research Center for Virtual Reality and Visualization
Donau-City-Straße 1
A-1220 Vienna, Austria
Email: `msh@vrvis.at`

Markus Hadwiger is a senior researcher at the VRVis Research Center in Vienna, Austria. He received his Ph.D. in computer science from the Vienna University of Technology in 2004, and has been a researcher at VRVis since 2000, working in the Basic Research on Visualization group and the Medical Visualization group (since 2004). He has been involved in several courses and tutorials about volume rendering and visualization at ACM SIGGRAPH, IEEE Visualization, and Eurographics. He is a co-author of the book *Real-Time Volume Graphics* published by A K Peters.

*e-mail: ropinski@math.uni-muenster.de
†e-mail: rezk@fb12.uni-siegen.de
‡e-mail: msh@vrvis.at
§e-mail: patric.ljung@siemens.com

Christof Rezk-Salama
Computer Graphics Group
University of Siegen
Hoelderlinstr. 3
57066 Siegen, Germany
Email: `rezk@fb12.uni-siegen.de`

Christof Rezk-Salama has received a PhD from the University of Erlangen-Nuremberg as a scholarship holder of the graduate college *3D Image Analysis and Synthesis*. He has worked as a research engineer for the R&D department of Siemens Medical Solutions. Since October 2003 he is working as an assistant professor at the Computer Graphics Group of the University of Siegen, Germany.

The results of his research have been presented at international conferences, including ACM SIGGRAPH, IEEE Visualization, Eurographics, MICCAI and Graphics Hardware. He is regularly holding lectures, courses and seminars on computer graphics, scientific visualization, character animation and graphics programming.

He has gained practical experience in applying computer graphics to several scientific projects in medicine, geology and archaeology. Christof Rezk-Salama has released the award winning open-source project *OpenQVis* and is a co-author of the book *Real-Time Volume Graphics*.

Timo Ropinski
Visualization and Computer Graphics Research Group (VisCG)
University of Münster
Einsteinstr. 62
48149 Münster, Germany
Email: `ropinski@math.uni-muenster.de`

Timo Ropinski is a postdoctoral researcher working in the field of medical volume visualization. After receiving his PhD in 2004 from the University of Münster, he became a project leader within the collaborative research center SFB 656, a cooperation between researchers from medicine, mathematics, chemistry, physics and computer science. His research is focused on interactive aspects in medical volume visualization with the goal to make these techniques more accessible. He is regularly holding lectures and seminars on computer graphics and scientific visualization, and is the initiator of the *Voreen* open source project, in which a flexible volume rendering framework is developed. The results of his scientific work have been published in various international conferences including Eurographics, IEEE VR, VMV and others.

Patric Ljung
Department of Imaging and Visualization
Siemens Corporate Research
755 College Road East
Princeton, NJ 08540, U.S.A.
Email: `patric.ljung@siemens.com`

Patric Ljung joined in 2007 Siemens Corporate Research in Princeton, NJ, where he works as a Research Scientist in the Imaging Architectures group. He received 2006 his PhD in Scientific Visualization from Linköping University, Sweden and graduated with honors in 2000 his MS in Information Technology from Linköping University. Between 1989 and 1995 he worked as a software engineer with embedded and telecom systems involving software architectures, graphical user interfaces, voice-mail systems, communication protocols, network and interprocess communication, compilers.

Dr. Ljung has published several papers in international conferences and journals including IEEE Visualization, Eurographics conferences, IEEE TVCG and others, on volume rendering of large medical data sets, GPU-based ray-casting of multiresolution data sets. One important focus area has been Virtual Autopsies for forensic pathology. His current research interest is in advanced illumination and shading techniques, software architectures for extensible graphics, and management and rendering of large medical data sets.

6 Contents and Schedule

The half-day tutorial will consist of four blocks, each of about 45 minute length. From a didactic point of view, each block will loosely build upon the information provided in previous blocks with growing complexity and increasing level of difficulty. **8:30 –**

9.15: Introduction and Basics

Speaker: Markus Hadwiger

- Application Areas for Volume Rendering
- Benefits and Drawbacks of Ray-Casting
- GPU-based Volume Ray-Casting
- Space Leaping and Early Ray Termination
- Memory Management
- Multiresolution LOD and Adaptive sampling

GPU-based ray-casting is a direct volume rendering technique with a steadily growing popularity in scientific visualization. The first part of the tutorial gives an introduction to GPU-based ray-casting discussing the benefits and drawbacks compared to previous real-time volume rendering approaches like 3D texture slicing. After this introductory part, the attendees should have a thorough understanding of how GPU-based ray-casting approaches work, and how they must be employed in practice to achieve optimal performance. This section covers optimization techniques such as empty space leaping, level-of-detail and adaptive sampling. The implementation of spatial partitioning techniques, hierarchical data structures and memory management strategies for the handling of large volumes are explained in detail.

9:15 – 10.00: Light Interaction

Speaker: Timo Ropinski

- Light Transport and Illumination Models
- Local Volume Illumination
- Specular Reflections through Ray-Tracing
- Soft vs. Hard Shadows
- Semi-Transparent Shadows with Deep Shadow Maps
- Simulation of Color Bleeding

The second part gives an overview to physically-based light transport introducing the phase function as a volumetric equivalent to the bidirectional reflectance distribution function (BRDF) known from surface lighting. This part covers basic local illumination techniques with respect to GPU-based ray-casting, including run-time gradient estimation, as well as more advanced local illumination techniques for volume data and isosurfaces. For scientific visualization, the inclusion of shadows into volume rendering systems significantly increases the perception of spatial structures and depth relationships. This part covers state-of-the art techniques for shadow computation in semi-transparent volume renditions, including soft and hard shadows. We will show how hard shadows can be computed interactively using efficient GPU-based approaches, such as deep shadow maps.

10:30 – 11.15: Ambient Occlusion

Speaker: Patric Ljung

- Ambient Occlusion for Isosurfaces
- Local Ambient Occlusion (DVR)
- Dynamic Ambient Occlusion (DVR)

Light interactions may be also approximated using ambient occlusion techniques for semi-transparent volumes and dynamic illumination scenarios.

11:15 – 12.00: Scattering

Speaker: Christof Rezk-Salama

- Monte-Carlo Integration
- Single versus Multiple Scattering
- Color Bleeding
- Translucency
- Monte-Carlo Scattering
- First-order Multiple Scattering
- Scattering with Deep Shadow Maps

Scattering effects account for photons which are reflected multiple times before they reach the eye. For volumetric data, scattering effects are usually included to capture the visual appearance of participating media. Such effects are important for modeling of natural phenomena such as smoke and clouds in visual arts. For scientific visualization, volume scattering may be used to convey tissue properties in a more realistic way. Translucency is an important visual property of soft tissue. This part of the tutorial covers interactive implementations of both single and multiple scattering effects in volumes. Compared to visual arts, where the optical properties inside the volume are mostly assumed homogeneous, in order to implement scattering and translucency in scientific data, e. g., tomographic scans, slightly different approaches must be chosen to optimally convey the structures. Several state-of-the-art approaches to achieve multiple scattering effects including color bleeding and translucency for interactive volume graphics are explained.

REFERENCES

[1] K. M. Beason, J. Grant, D. C. Banks, B. Futch, and M. Y. Hussaini. Pre-computed illumination for isosurfaces. In *VDA '94: Proceedings of the conference on Visualization and Data Analysis '06 (SPIE Vol. 6060)*, pages 1–11, 2006.

[2] U. Behrens and R. Ratering. Adding shadows to a texture-based volume renderer. In *VVS '98: Proceedings of the 1998 IEEE symposium on Volume visualization*, pages 39–46. ACM Press, 1998.

[3] I. Boada, I. Navazo, and R. Scopigno. Multiresolution volume visualization with a texture-based octree. *The Visual Computer*, 17:185–197, 2001.

[4] P. Desgranges and K. Engel. US patent application 2007/0013696 A1: Fast ambient occlusion for direct volume rendering, 2007.

[5] P. Desgranges, K. Engel, and G. Paladini. Gradient-free shading: A new method for realistic interactive volume rendering. In *VMV '05: Proceedings of the international fall workshop on Vision, Modeling, and Visualization*, pages 209–216, 2005.

[6] K. Engel, M. Hadwiger, J. Kniss, C. Rezk-Salama, and D. Weiskopf. *Real-Time Volume Graphics*. AK Peters, 2006.

[7] S. Guthe and W. Strasser. Advanced techniques for high quality multiresolution volume rendering. In *Computers & Graphics*, volume 28, pages 51–58. Elsevier Science, February 2004.

[8] S. Guthe, M. Wand, J. Gonser, and W. Straer. Interactive rendering of large volume data sets. In *Proceedings IEEE Visualization 2002*, pages 53–60, 2002.

[9] M. Hadwiger, A. Kratz, C. Sigg, and K. Bühler. Gpu-accelerated deep shadow maps for direct volume rendering. In *GH '06: Proceedings of the 21st ACM SIGGRAPH/Eurographics symposium on Graphics hardware*, pages 49–52, New York, NY, USA, 2006. ACM Press.

[10] F. Hernell, P. Ljung, and A. Ynnerman. Efficient ambient and emissive tissue illumination using local occlusion in multiresolution volume rendering. In *Proceedings Eurographics/IEEE Workshop on Volume Graphics 2007*, pages 1–8,129. IEEE, 2007.

[11] R. Khler, J. Wise, T. Abel, and H.-C. Hege. Gpu-assisted raycasting for cosmological adaptive mesg refinement simulations. In *Proceedings Eurographics/IEEE Workshop on Volume Graphics 2006*, pages 103–110,144, 2006.

[12] T. Klein, M. Strengert, S. Stegmaier, and T. Ertl. Exploiting frame-to-frame coherence for accelerating high-quality volume raycasting on graphics hardware. In *Proceedings IEEE Visualization 2005*, pages 223–230, 2005.

[13] J. Kniss, S. Premoze, C. Hansen, and D. Ebert. Interactive translucent volume rendering and procedural modeling. In *VIS '02: Proceedings of the conference on Visualization '02*, pages 109–116. IEEE Computer Society, 2002.

[14] J. Kniss, S. Premoze, C. Hansen, P. Shirley, and A. McPherson. A model for volume lighting and modeling. *IEEE Transactions on Visualization and Computer Graphics*, 9(2):150–162, 2003.

[15] J. Krüger and R. Westermann. Acceleration techniques for GPU-based volume rendering. In *Proceedings IEEE Visualization 2003*, 2003.

[16] E. C. LaMar, B. Hamann, and K. I. Joy. Multiresolution techniques for interactive texture-based volume visualization. In *Proceedings IEEE Visualization 1999*, pages 355–362, 1999.

[17] M. S. Langer and H. H. Bülthoff. Depth discrimination from shading under diffuse lighting. *Perception*, 29(6):649–660, 2000.

[18] P. Ljung. Adaptive sampling in single pass, GPU-based raycasting of multiresolution volumes. In *Proceedings Eurographics/IEEE Workshop on Volume Graphics 2006*, pages 39–46,134, 2006.

[19] P. Ljung, C. Lundström, and A. Ynnerman. Multiresolution interblock interpolation in direct volume rendering. In *Proceedings Eurographics/IEEE Symposium on Visualization 2006*, pages 259–266, 2006.

[20] P. Ljung, C. Lundström, A. Ynnerman, and K. Museth. Transfer function based adaptive decompression for volume rendering of large medical data sets. In *Proceedings IEEE Volume Visualization and Graphics Symposium 2004*, pages 25–32, 2004.

[21] P. Ljung, C. Winskog, A. Perssson, C. Lundström, and A. Ynnerman. Full body virtual autopsies using a state-of-the-art volume rendering pipeline. *IEEE Transactions on Visualization and Computer Graphics (Proceedings Visualization/Information Visualization 2006)*, 12:869–876, 2006.

[22] T. Lokovic and E. Veach. Deep shadow maps. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 385–392, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[23] G. Marmitt, H. Friedrich, and P. Slusallek. Interactive Volume Rendering with Ray Tracing. In *Eurographics State of the Art Reports*, 2006.

[24] N. Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, 1995.

[25] S. Parker, M. Parker, Y. Livnat, P.-P. Sloan, C. Hansen, and P. Shirley. Interactive ray tracing for volume visualization. *IEEE Transactions on Visualization and Computer Graphics*, 5(3):238–250, 1999.

[26] S. Roettger, S. Guthe, D. Weiskopf, and T. Ertl. Smart hardware-accelerated volume rendering. In *Procceedings of EG/IEEE TCVG Symposium on Visualization VisSym '03*, pages 231–238, 2003.

[27] T. Ropinski, J. Kasten, and K. H. Hinrichs. Efficient Shadows for GPU-based Volume Raycasting. In *Proceedings of the 16th International Conference in Central Europe on Computer Graphics, Visualization (WSCG08)*, pages 17–24, 2008. (Annahmequote: 29.6%).

[28] T. Ropinski, J. Meyer-Spradow, S. Diepenbrock, J. Mensmann, and K. H. Hinrichs. Interactive Volume Rendering with Dynamic Ambient Occlusion and Color Bleeding. *Computer Graphics Forum (Eurographics 2008)*, 27(2):567–576, 2008. (Annahmequote: 19.3%).

[29] C. R. Salama. GPU-Based Monte-Carlo Volume Raycasting. In *Proc. Pacific Graphics*, 2007.

[30] M. Sattler, R. Sarlette, T. Mücken, and R. Klein. Exploitation of human shadow perception for fast shadow rendering. In *APGV '05: Proceedings of the 2nd symposium on Applied perception in graphics and visualization*, pages 131–134. ACM Press, 2005.

[31] L. M. Sobierajski and A. E. Kaufman. Volumetric ray tracing. In *VVS '94: Proceedings of the 1994 symposium on Volume Visualization '94*, pages 11–18. ACM Press, 1994.

[32] S. Stegmaier, M. Strengert, T. Klein, and T. Ertl. A simple and flexible volume rendering framework for graphics-hardware–based raycasting. In *Eurographics/IEEE Volume Graphics Symposium*. Eurographics, 2005.

[33] A. J. Stewart. Vicinity shading for enhanced perception of volumetric data. In *VIS '03: Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, page 47. IEEE Computer Society, 2003.

[34] J. E. Vollrath, T. Schafhitzel, and T. Ertl. Employing complex GPU data structures for the interactive visualization of adaptive mesh refinement data. In *Proceedings Eurographics/IEEE Workshop on Volume Graphics 2006*, pages 55–58,136, 2006.

[35] M. Weiler, R. Westermann, C. Hansen, K. Zimmerman, and T. Ertl. Level–of–detail volume rendering via 3d textures. In *Proceedings IEEE Volume Visualization and Graphics Symposium 2000*, pages 7–13. ACM Press, 2000.

[36] C. Wyman, S. Parker, C. Hansen, and P. Shirley. Interactive display of isosurfaces with global illumination. *IEEE Transactions on Visualization and Computer Graphics*, 12(2):186–196, 2006.

[37] C. Zhang and R. Crawfis. Shadows and soft shadows with participating media using splatting. *IEEE Transactions on Visualization and Computer Graphics*, 9(2):139–149, 2003.

Advanced Illumination Techniques for GPU-Based Volume Raycasting

Markus Hadwiger
VR VIS Research Center
Vienna, Austria

Patric Ljung
Siemens Corporate Research
Princeton, NJ, USA

Christof Rezk Salama
Computer Graphics Group
Institute for Vision and Graphics
University of Siegen, Germany

Timo Ropinski
Visualization and Computer
Graphics Research Group,
University of Münster, Germany

Ray-Casting Basics

Markus Hadwiger
VR VIS Research Center
Vienna, Austria

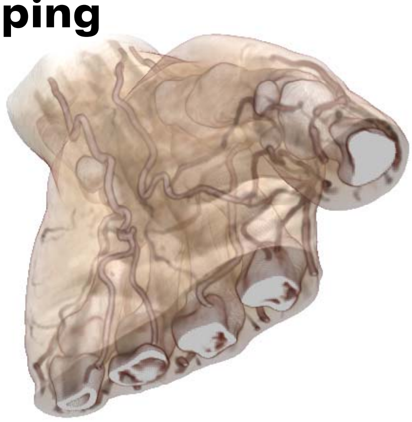
Patric Ljung
Siemens Corporate Research
Princeton, NJ, USA

Christof Rezk Salama
Computer Graphics Group
Institute for Vision and Graphics
University of Siegen, Germany

Timo Ropinski
Visualization and Computer
Graphics Research Group,
University of Münster, Germany

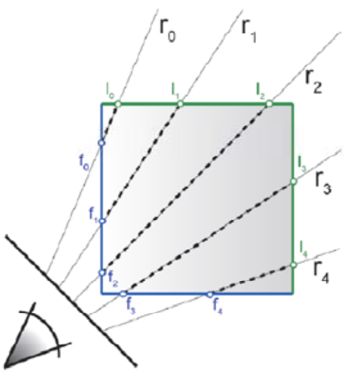
Talk Outline

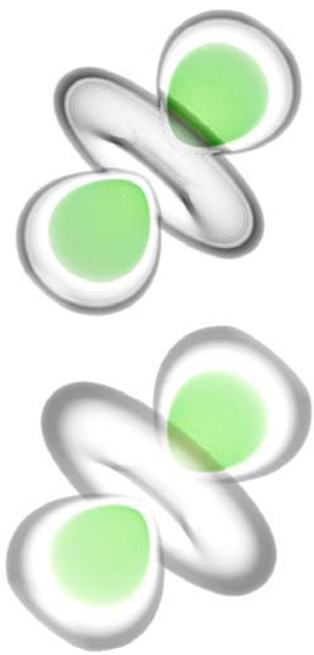
- **Why use ray-casting instead of slicing?**
- **Ray-casting of rectilinear (structured) grids**
  - **Basic approaches on GPUs**
  - **Basic acceleration methods**
  - **Object-order empty space skipping**
  - **Isosurface ray-casting**
  - **Endoscopic ray-casting**
- **Memory management**
- **LOD selection**



Why Ray-Casting on GPUs?

- **Most GPU rendering is object-order (rasterization)**
- **Image-order is more "CPU-like"**
  - **Recent fragment shader advances**
  - **Simpler to implement**
  - **Very flexible (e.g., adaptive sampling)**
  - **Correct perspective projection**
  - **Can be implemented in single pass!**
  - **Native 32-bit compositing**

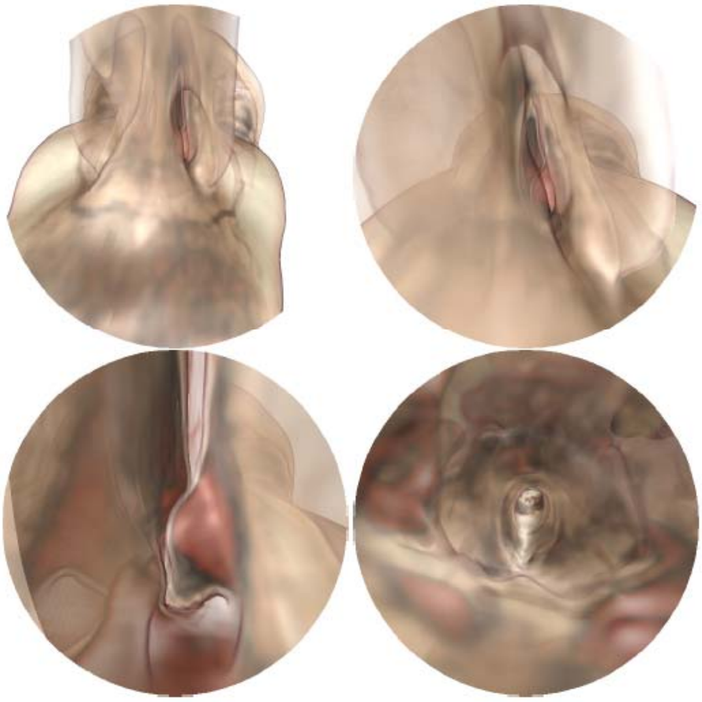




Correct Perspective

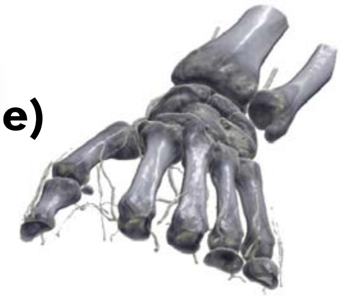
- Entering the volume
- Wide field of view

- Fly-throughs
- Virtual endoscopy
- Integration into perspective scenes, e.g., games

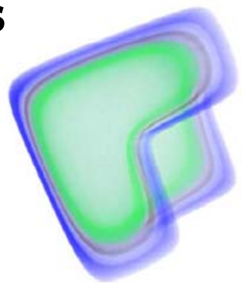


Recent GPU Approaches

- Rectilinear grids
  - [Krüger and Westermann, 2003]
  - [Röttger et al., 2003]
  - [Green, 2004] (NVIDIA SDK Example)
  - [Stegmaier et al., 2005]
  - [Scharsach et al., 2006]



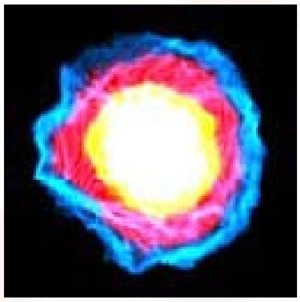
- Unstructured (tetrahedral) grids
  - [Weiler et al., 2002, 2003, 2004]
  - [Bernardon, 2004]



Single-Pass Ray-Casting

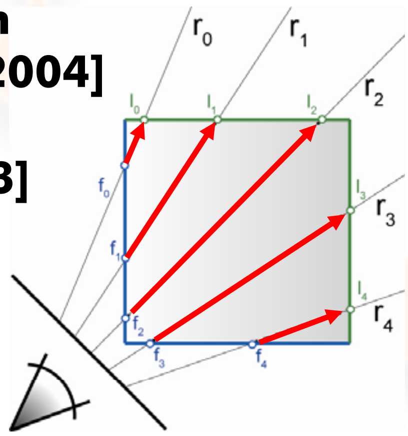
- **Enabled by conditional loops in fragment shaders (Shader Model 3; e.g., Geforce 8800, ATI X2800)**
- **Substitute multiple passes and early-z testing by single loop and early loop exit**
- **No compositing buffer: full 32-bit precision!**

- **NVIDIA example: compute ray intersections with bounding box, march along rays and composite**



Basic Ray Setup / Termination

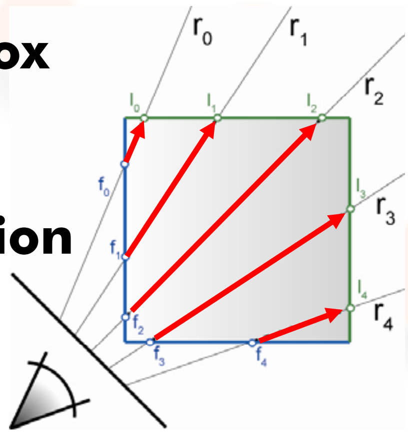
- **Two main approaches:**
  - **Procedural ray/box intersection [Röttger et al., 2003], [Green, 2004]**
  - **Rasterize bounding box [Krüger and Westermann, 2003]**



- **Some possibilities**
  - **Ray start position and exit check**
  - **Ray start position and exit position**
  - **Ray start position and direction vector**

Procedural Ray Setup/Term.

- Everything handled in the fragment shader
- Procedural ray / bounding box intersection



- Ray is given by camera position and volume entry position
- Exit criterion needed

- Pro: simple and self-contained
- Con: full load on the fragment shader

Fragment Shader

- Rasterize front faces of volume bounding box

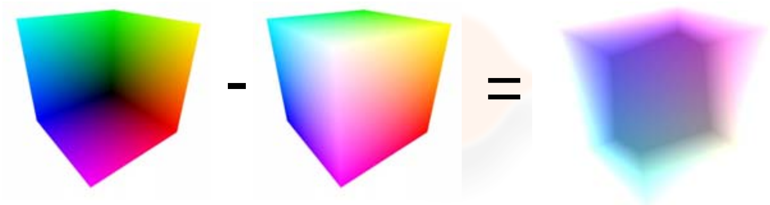
- Texcoords are volume position in [0,1]
- Subtract camera position

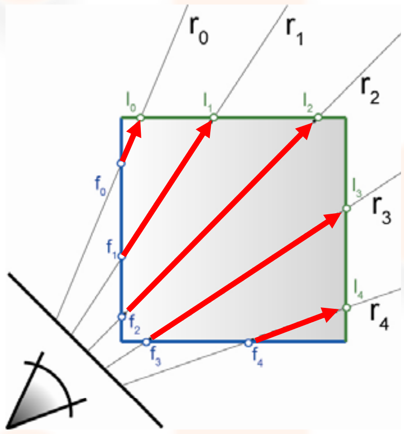
- Repeatedly check for exit of bounding box

```
// Cg fragment shader code for single-pass ray casting
float4 main(VS_OUTPUT IN, float4 TexCoord0 : TEXCOORD0,
            uniform sampler3D SamplerDataVolume,
            uniform sampler1D SamplerTransferFunction,
            uniform float3 camera,
            uniform float stepsize,
            uniform float3 volExtentMin,
            uniform float3 volExtentMax
            ) : COLOR
{
    float4 value;
    float scalar;
    // Initialize accumulated color and opacity
    float4 dst = float4(0,0,0,0);
    // Determine volume entry position
    float3 position = TexCoord0.xyz;
    // Compute ray direction
    float3 direction = TexCoord0.xyz - camera;
    direction = normalize(direction);
    // Loop for ray traversal
    for (int i = 0; i < 200; i++)  // Some large number
    {
        // Data access to scalar value in 3D volume texture
        value = tex3D(SamplerDataVolume, position);
        scalar = value.a;
        // Apply transfer function
        float4 src = tex1D(SamplerTransferFunction, scalar);
        // Front-to-back compositing
        dst = (1.0-dst.a) * src + dst;
        // Advance ray position along ray direction
        position = position + direction * stepsize;
        // Ray termination: Test if outside volume ...
        float3 temp1 = sign(position - volExtentMin);
        float3 temp2 = sign(volExtentMax - position);
        float inside = dot(temp1, temp2);
        // ... and exit loop
        if (inside < 3.0)
            break;
    }
    return dst;
}
```

"Image-Based" Ray Setup/Term.

- Rasterize bounding box front faces and back faces
  [Krüger and Westermann, 2003]
- Ray start position: front faces
- Direction vector: back−front faces





- Independent of projection (orthogonal/perspective)

Standard Ray-Casting Opt. (1)

Early ray termination

- Isosurfaces: stop when surface hit
- Direct volume rendering:
  stop when opacity >= threshold

- Several possibilities
  - Older GPUs: multi-pass rendering with early-z test
  - Shader model 3: break out of ray-casting loop
  - Current GPUs: early loop exit not optimal but good

Standard Ray-Casting Opt. (2)

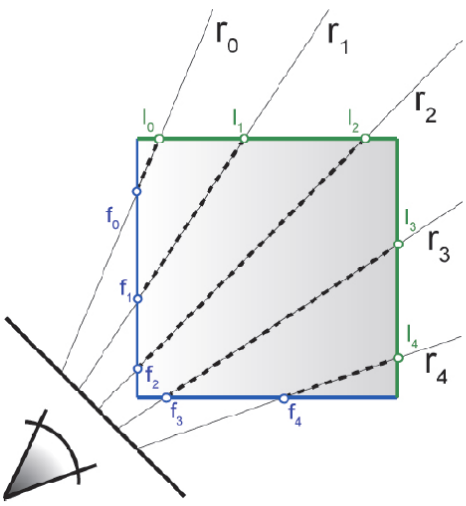
Empty space skipping

- Skip transparent samples
- Depends on transfer function
- Start casting close to first hit

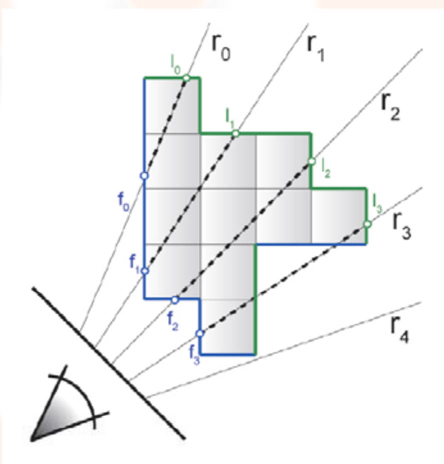
- Several possibilities
  - Per-sample check of opacity (expensive)
  - Traverse hierarchy (e.g., octree) or regular grid
  - These are image-order: what about object-order?

Object-Order Empty Space Skip. (1)

- Modify initial rasterization step



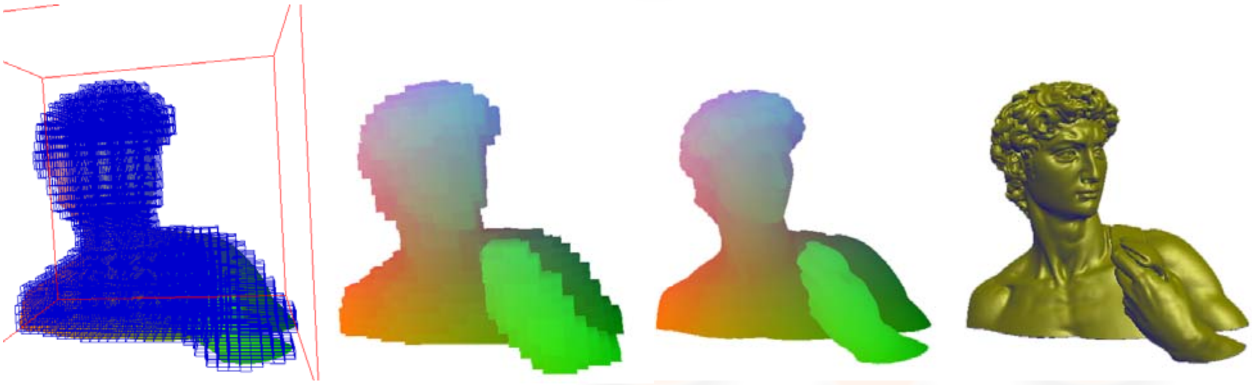
rasterize bounding box



rasterize “tight" bounding geometry

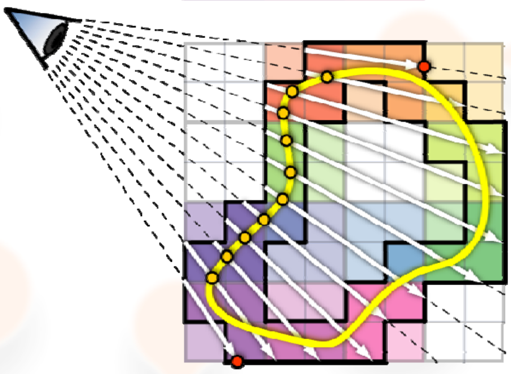
Object-Order Empty Space Skip. (2)

- Store min-max values of volume bricks
- Cull bricks against isovalue or transfer function
- Rasterize front and back faces of active bricks



Object-Order Empty Space Skip. (3)

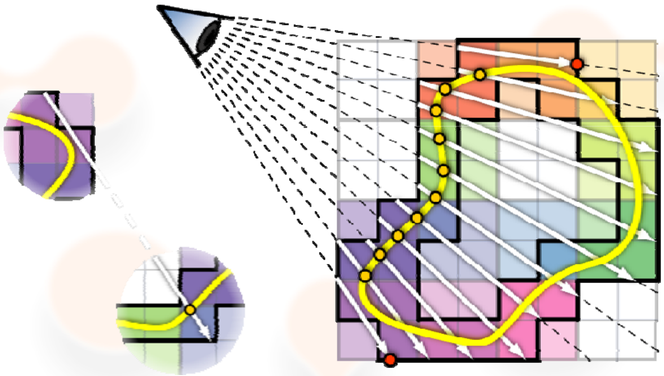
- Rasterize front and back faces of active min-max bricks
- Start rays on brick front faces
- Terminate when
  - Full opacity reached, or
  - Back face reached



Object-Order Empty Space Skip. (3)

- Rasterize front and back faces of active min-max bricks
- Start rays on brick front faces
- Terminate when
  - Full opacity reached, or
  - Back face reached

- Not all empty space is skipped



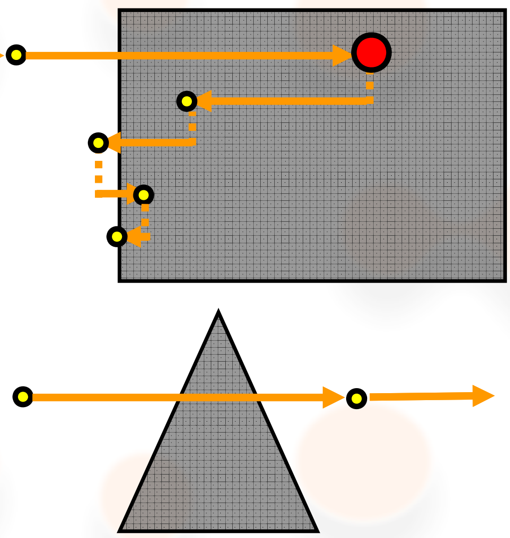
Isosurface Ray-Casting

- Isosurfaces/Level Sets
  - scanned data
  - distance fields
  - CSG operations
  - level sets: surface editing, simulation, segmentation, ...

Intersection Refinement (1)

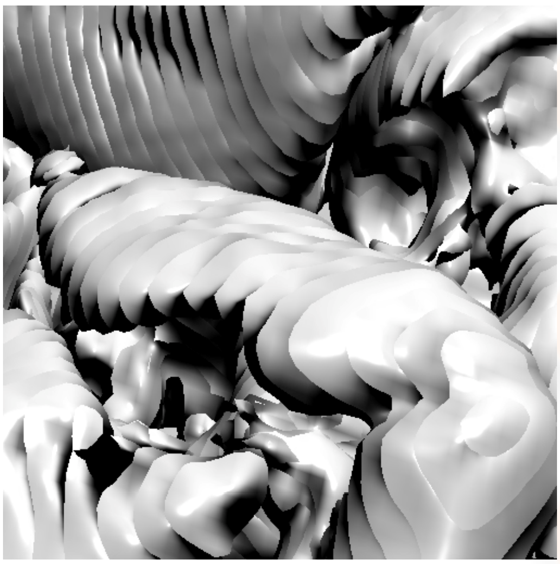
- Fixed number of bisection or binary search steps
- Virtually no impact on performance

- Refine already detected intersection
- Handle problems with small features / at silhouettes with adaptive sampling



Intersection Refinement (2)

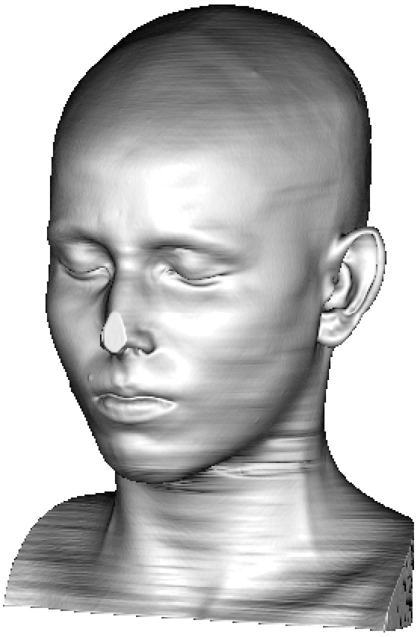
without refinement



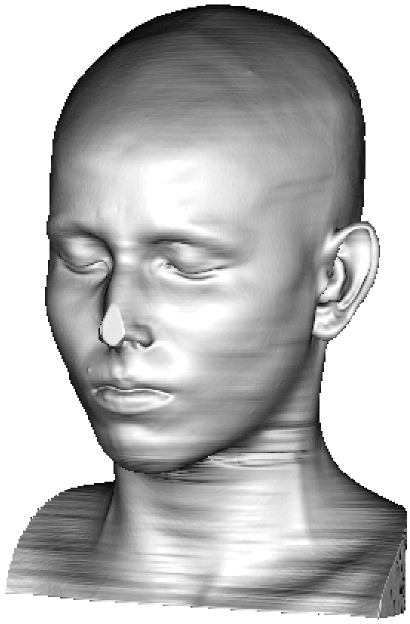
with refinement

sampling rate 1/5 voxel (no adaptive sampling)

Intersection Refinement (3)



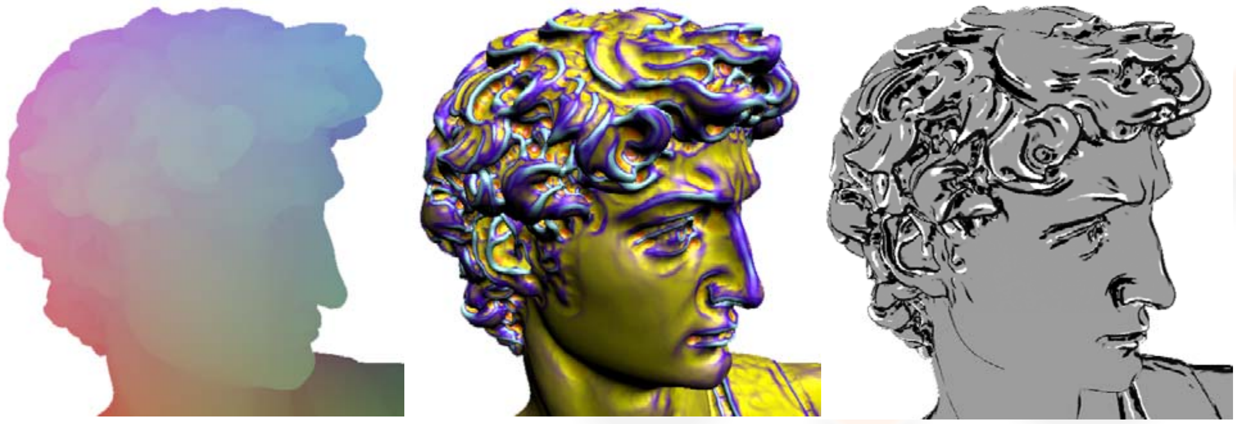
Sampling distance 1.0, 24 fps



Sampling distance 5.0, 66 fps

Deferred Isosurface Shading

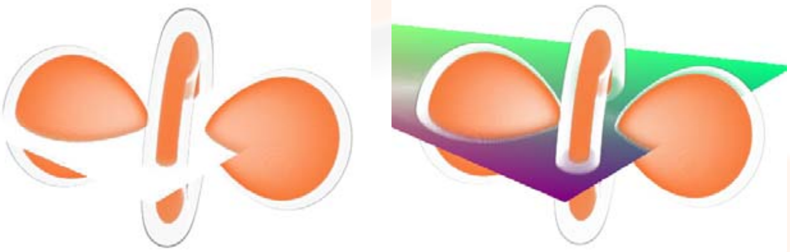
- Shading is expensive
  - Gradient computation; conditional execution not free
- Ray-casting step computes only intersection image



Enhancements (1)

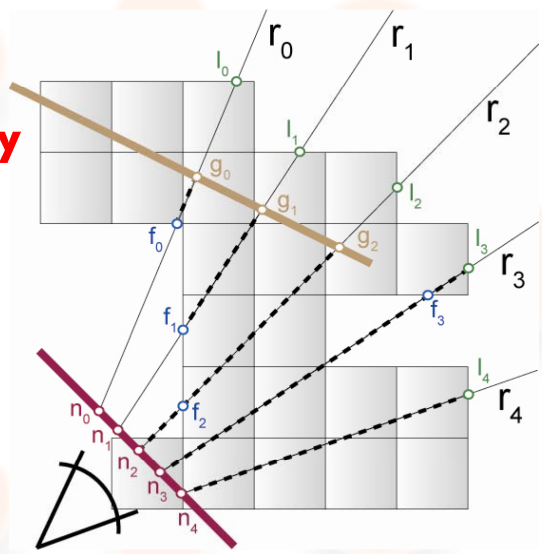
- Build on image-based ray setup
- Allow viewpoint inside the volume

- Intersect polygonal geometry



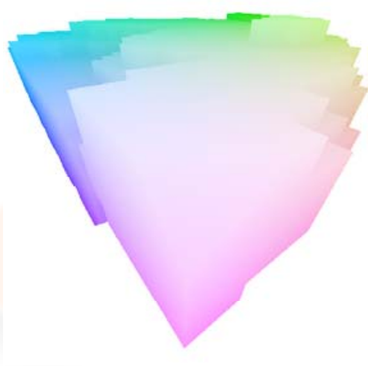
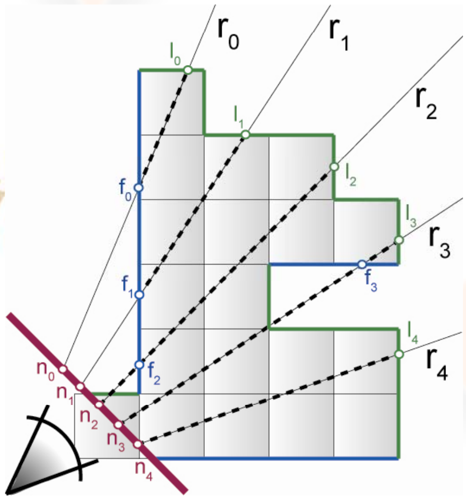
Enhancements (2)

1. **Starting position computation**
   ⇨ **Ray start position image**
2. **Ray length computation**
   ⇨ **Ray length image**
3. **Render polygonal geometry**
   ⇨ **Modified ray length image**
4. **Raycasting**
   ⇨ **Compositing buffer**
5. **Blending**
   ⇨ **Final image**



Moving Into The Volume (1)

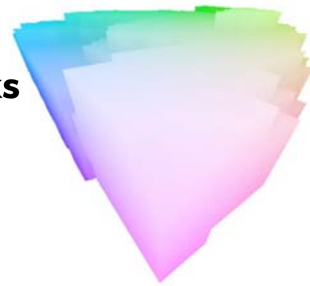
- Near clipping plane clips into front faces

- Fill in holes with near clipping plane
- Can use depth buffer [Scharsach et al., 2006]

Moving Into The Volume (2)

1. Rasterize near clipping plane
   - Disable depth buffer, enable color buffer
   - Rasterize entire near clipping plane
2. Rasterize nearest back faces
   - Enable depth buffer, disable color buffer
   - Rasterize *nearest back faces* of active bricks
3. Rasterize nearest front faces
   - Enable depth buffer, enable color buffer
   - Rasterize *nearest front faces* of active bricks

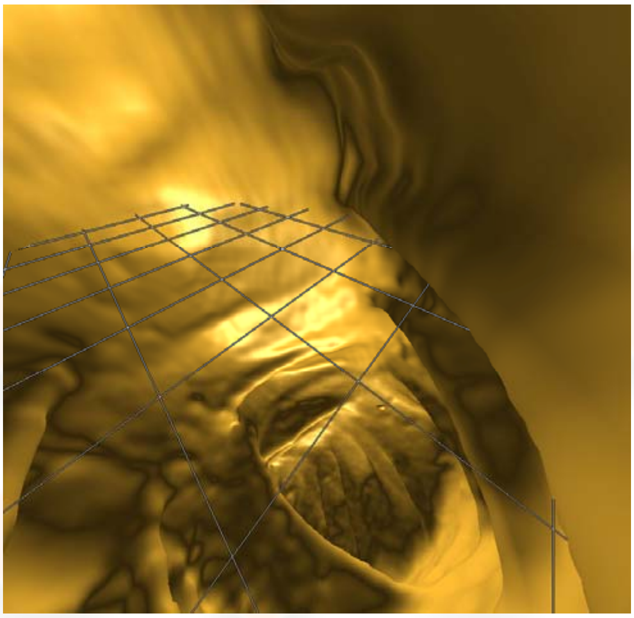
Virtual Endoscopy

- Viewpoint inside the volume with wide field of view
- E.g.: virtual colonoscopy

- Hybrid isosurface rendering / direct volume rendering
- E.g.: colon wall and structures behind

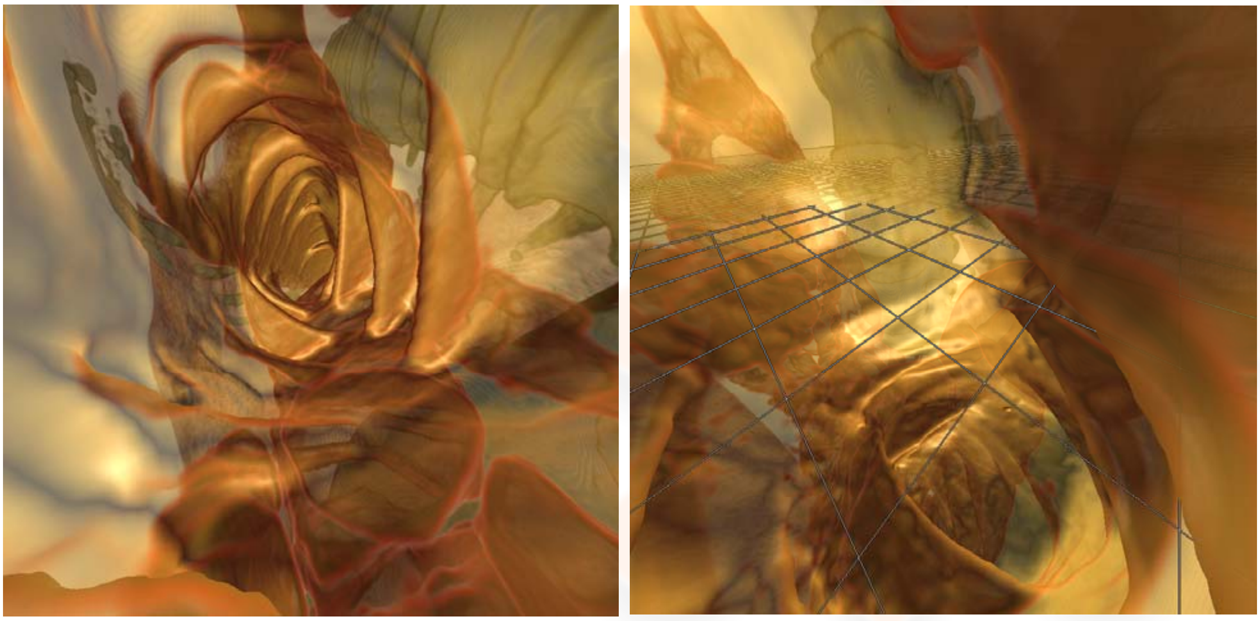
Virtual Colonoscopy

- First find isosurface; then continue with DVR



Virtual Colonoscopy

- First find isosurface; then continue with DVR



Hybrid Ray-Casting (1)

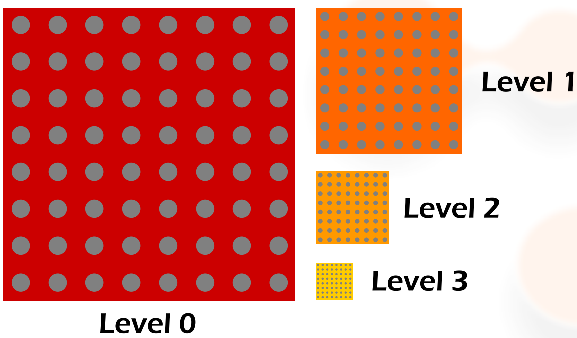
- Isosurface rendering
  - Find isosurface first
  - Semi-transparent shading provides surface information
- Additional unshaded DVR
  - Render volume behind the surface with unshaded DVR
  - Isosurface is starting position
  - Start with (1.0-iso_opacity)

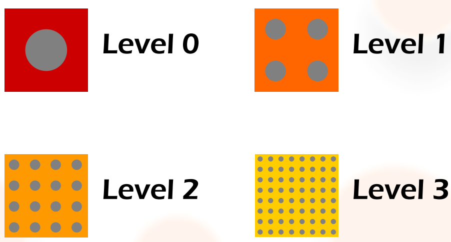
Hybrid Ray-Casting (2)

- Hiding sampling artifacts (similar to interleaved sampling, [Heidrich and Keller, 2001])

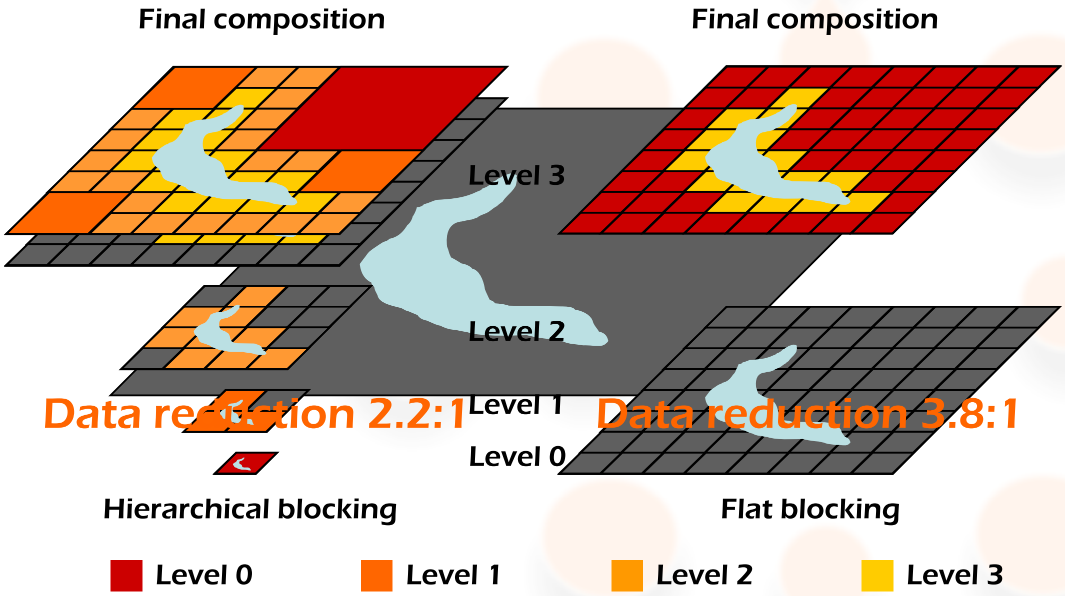
Multiresolution Volumes

- Hierarchical blocking
  - Constant data size
  - Lower resolution blocks cover increasing spatial size
- Flat blocking structure
  - Constant spatial size
  - Lower resolution blocks have fewer samples





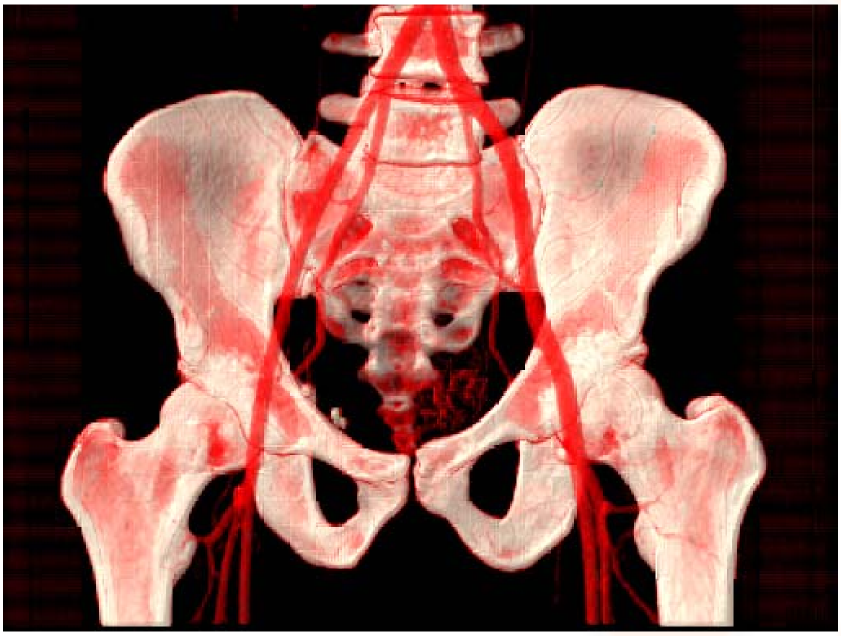
Hierarchical vs Flat Blocking



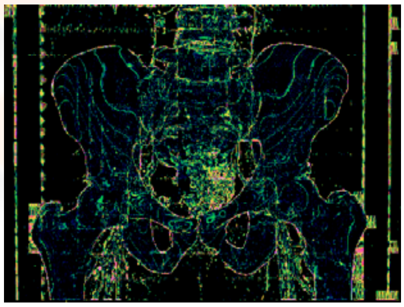
TF Based Level-of-Detail

- **Block resolution derived by TF content**
  - **Content of a block in the current TF domain**
  - **Empty: All voxels are transparent**
  - **Homogeneous: Non-transparent voxels, similar color**
  - **Varying: Varying transparency, varying color**
- **Block significance based on visual error**
  - **Color difference from CIE L*u*v*, ΔE**
  - **Perceptually uniform**
- **Optimize LOD to minimize visual error for a given memory limit**

TF Based LOD Data Reduction



144 MB, 512x512x384



CIE L*u*v* ΔE error

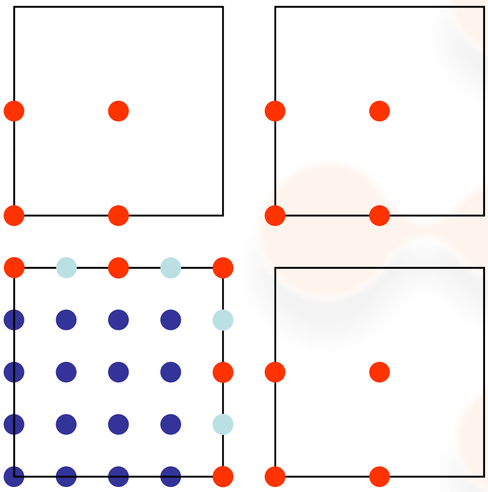
Data Reduction:

64:1

1.6 % (2.25 MB)

Sample replication and interpolation

- **Single sided replication**
  - **Weiler et al. ’00, Guthe et al. ’02**



Four-block neighborhood

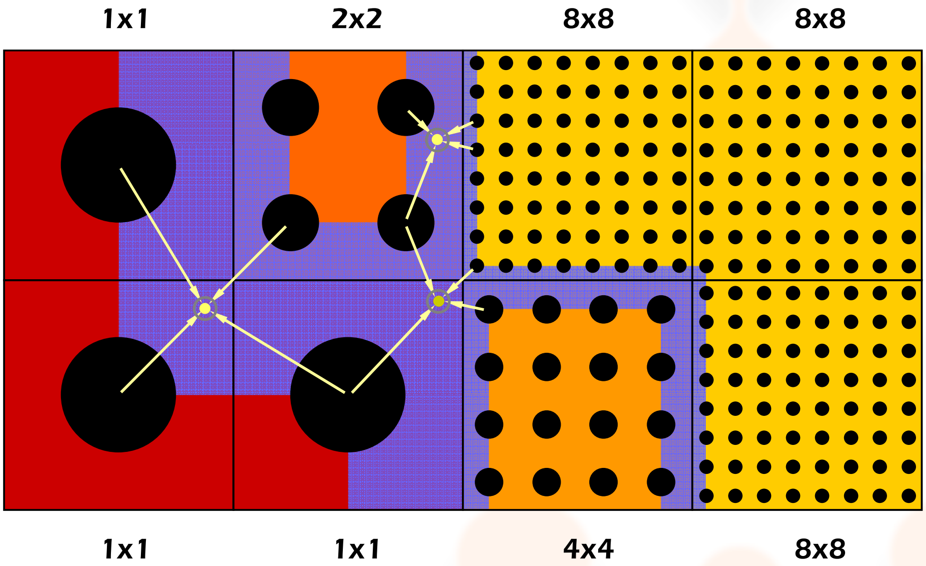
Replicate samples from neighbors

Interpolate missing samples

Each block is self-contained, with varied overhead [Kraus & Ertl ’02]

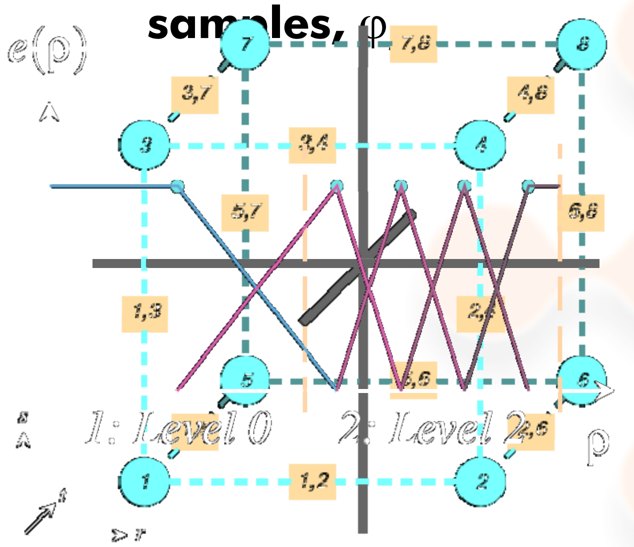
Double sided replication for improved continuity support [Weiler et al. ’00]

Multiresolution Interblock Interpolation



Interblock interpolation

- **A normalized sum, φ, of bounded block samples, φ**

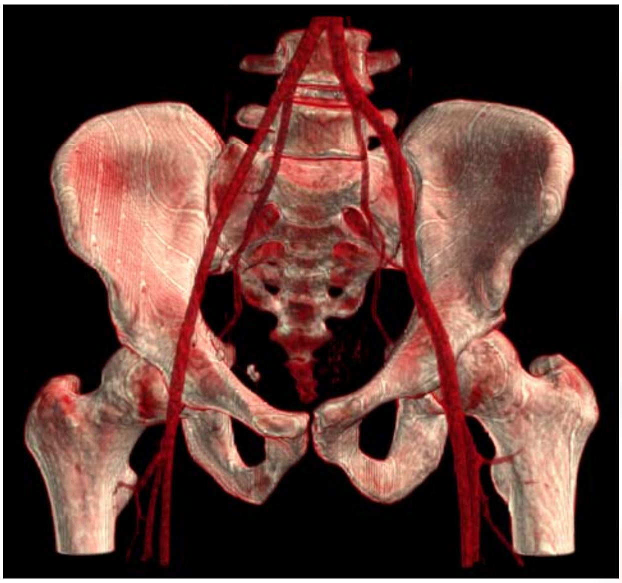


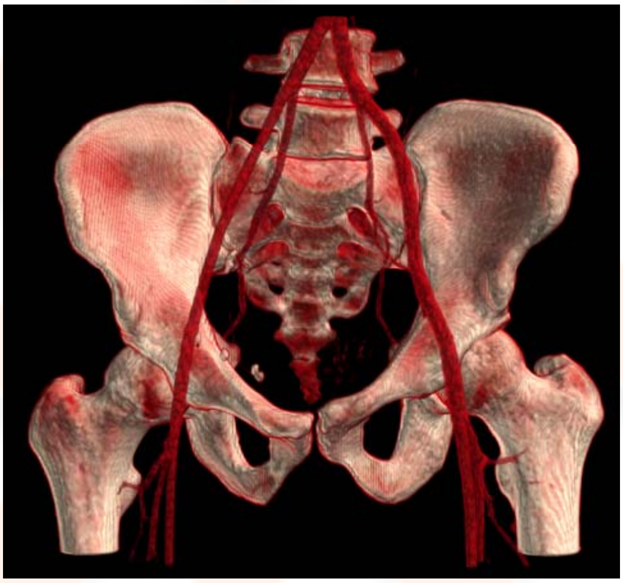
$$\omega_1 = (1-e_{1,2})\cdot(1-e_{1,3})\cdot(1-e_{1,5})$$
$$\omega_2 = e_{1,2}\cdot(1-e_{2,4})\cdot(1-e_{2,6})$$
$$\omega_3 = (1-e_{3,4})\cdot e_{1,3}\cdot(1-e_{3,7})$$
$$\omega_4 = e_{3,4}\cdot e_{2,4}\cdot(1-e_{4,8})$$
$$\omega_5 = (1-e_{5,6})\cdot(1-e_{5,7})\cdot e_{1,5}$$
$$\omega_6 = e_{5,6}\cdot(1-e_{6,8})\cdot e_{2,6}$$
$$\omega_7 = (1-e_{7,8})\cdot e_{5,7}\cdot e_{3,7}$$
$$\omega_8 = e_{7,8}\cdot e_{6,8}\cdot e_{4,8}$$

$$\varphi = \frac{\sum_{b=1}^{8} \omega_b \varphi_b}{\sum_{b=1}^{8} \omega_b}$$

Interblock Interpolation

Data reduction 80:1 (1.25% of original data size)

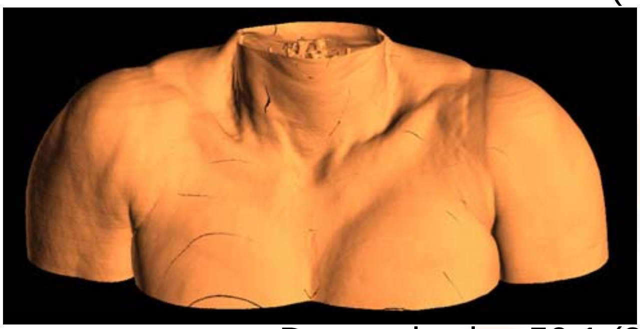




Without Interblock Interpolation With Interblock Interpolation

Interblock Interpolation

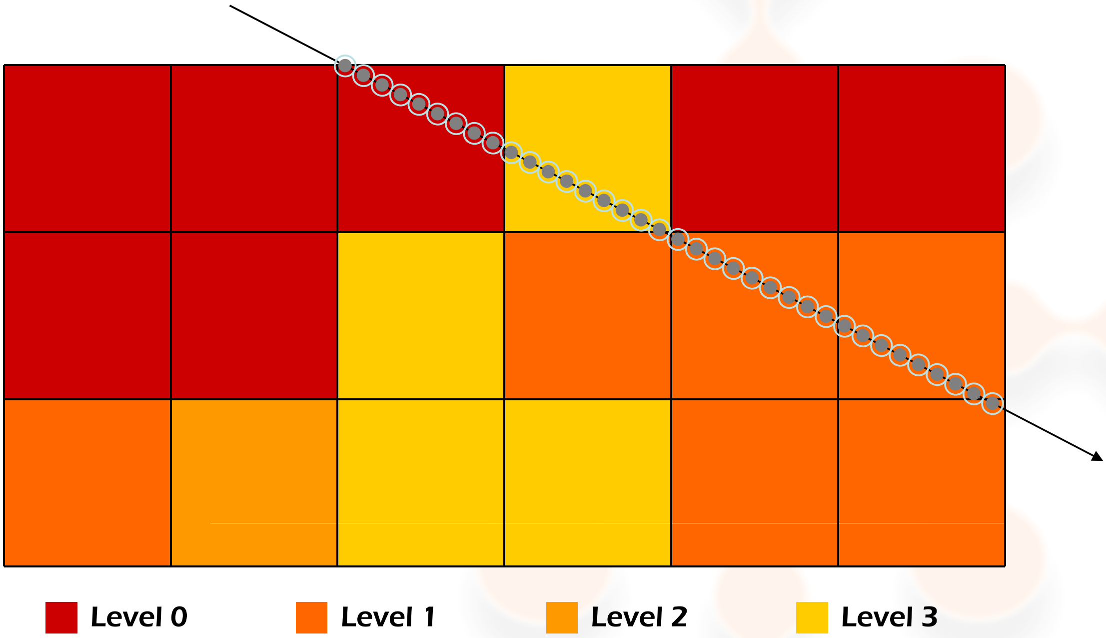
Data reduction 40:1 (2% of original data size)



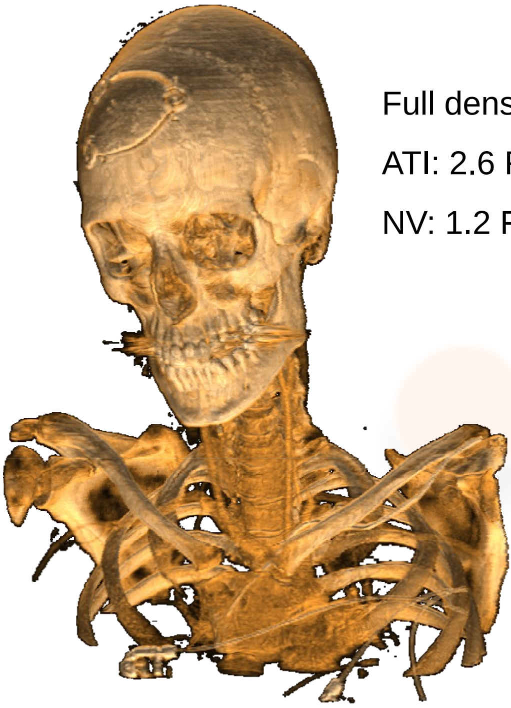
Data reduction 50:1 (2% of original data size)

Without Interblock Interpolation With Interblock Interpolation

Adaptive Volume Sampling



Adaptive Volume Sampling



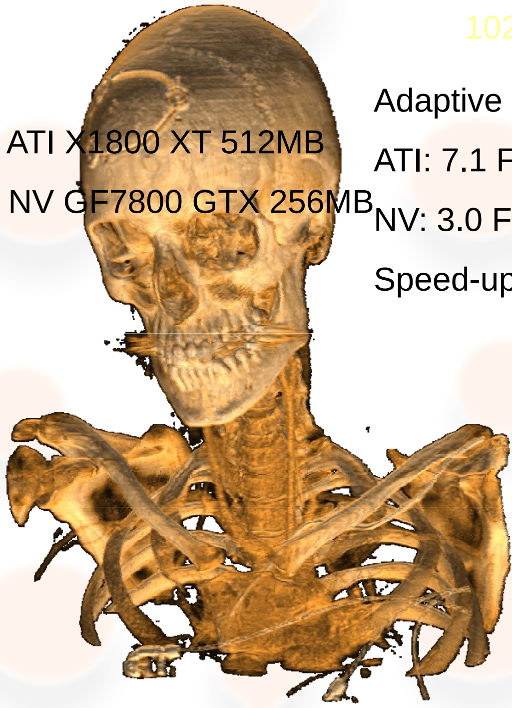
Full density

ATI: 2.6 FPS

NV: 1.2 FPS

- ATI X1800 XT 512MB
- NV GF7800 GTX 256MB

1024x1024



Adaptive density

ATI: 7.1 FPS

NV: 3.0 FPS

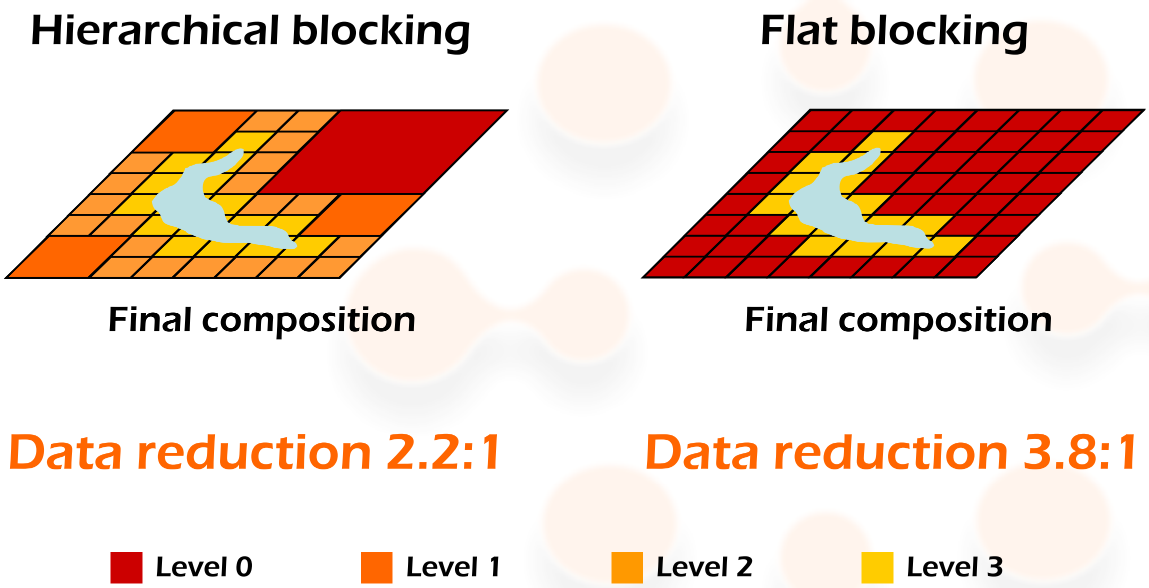
Speed-up: 2.6-2.8

Out-of-Core Data Management

- Disk performance
  - Data transfer rates, about 60 MB/s
  - Access density performance is still poor!!!
  - Random access of many small blocks is bad
- Group of blocks – GOBs
  - Large pages: 24 – 192 kb
  - Spatial coherence
  - Minimizes disk read requests
- Uncompressed storage
- Precomputed gradients

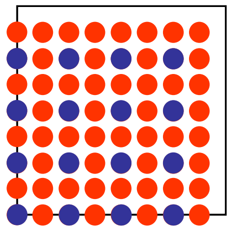
Out-of-Core Data Management

Hierarchical vs Flat Blocking



Intrablock sampling

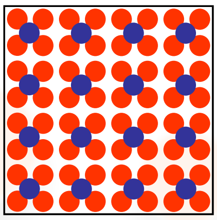
- **Algorithm outline**
  - **Determine closest block (nearest block sampling)**
  - **Lookup block size and location in index texture**
  - **Clamp sampling position to the block's sample boundary**
  - **Lookup interpolated sample in the packed texture**



Boundary aligned

Sample replication

Nearest neighbor downsampling



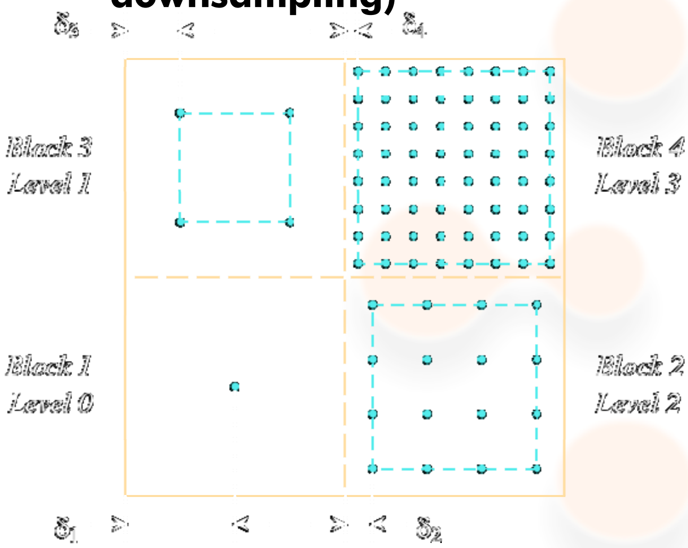
Centered

Average value downsampling

Wavelets

Nearest block sampling

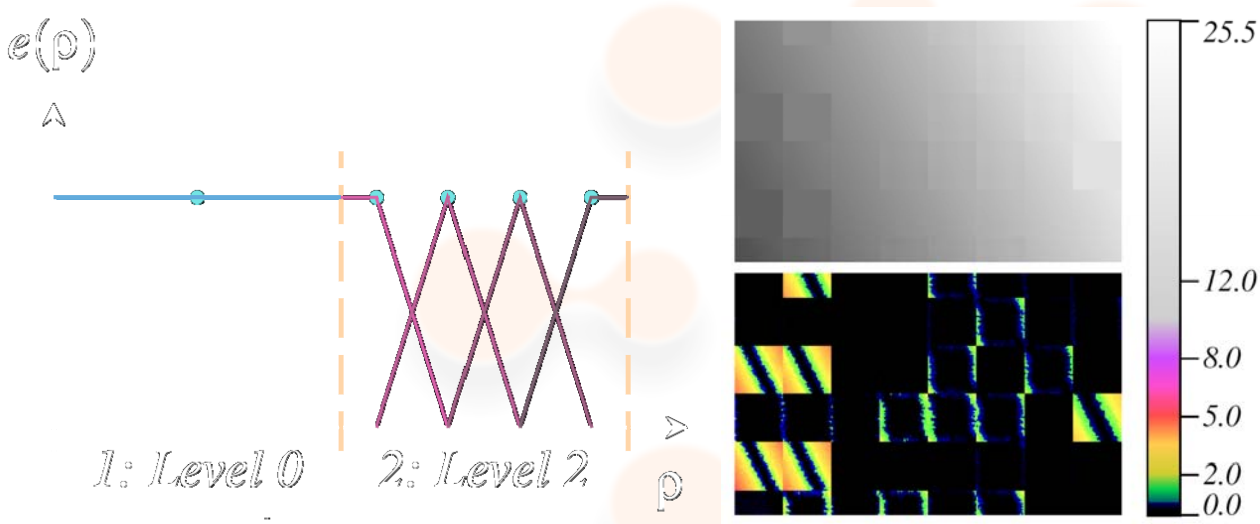
- **Block sample boundary**
  - **An inset, δ, from block boundary (mean-value downsampling)**



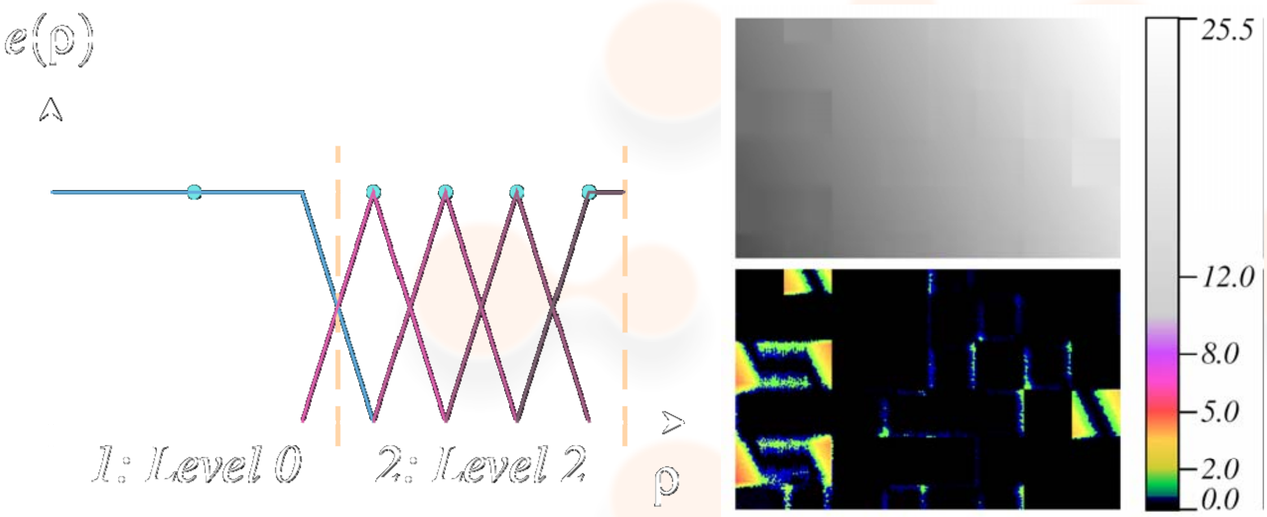
$$\delta(l) = \frac{1}{2^{1+l}}$$

$$u = C_{\delta}^{1-\delta}(u')$$

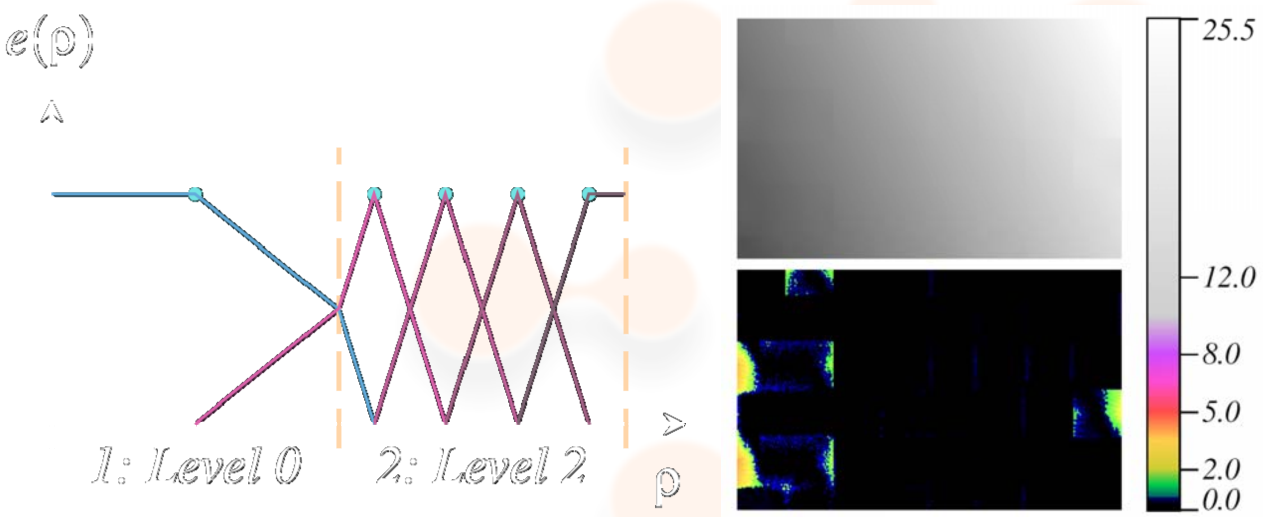
No block interpolation



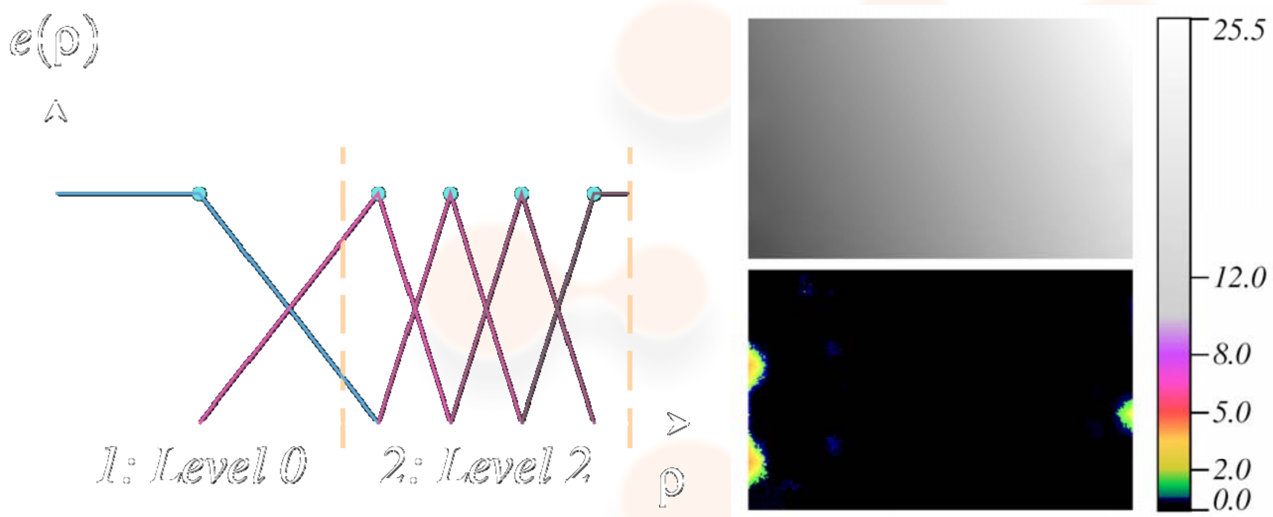
Minimum distance edge weight



Boundary split edge weight



Maximum distance edge weight



Conclusions

- **GPU ray-casting is an attractive alternative**
- **Very flexible and easy to implement**
- **Fragment shader conditionals are very powerful; performance pitfalls very likely to go away**
- **Mixing image-order and object-order well suited to GPUs (vertex and fragment processing!)**
- **Deferred shading allows complex filtering and shading at high frame rates**

Thank You!

Acknowledgments

- **Patric Ljung, Henning Scharsach, Christian Sigg, Daniel Weiskopf**

Light Interactions

Markus Hadwiger
VR VIS Research Center
Vienna, Austria

Patric Ljung
Siemens Corporate Research
Princeton, NJ, USA

Christof Rezk Salama
Computer Graphics Group
Institute for Vision and Graphics
University of Siegen, Germany

Timo Ropinski
Visualization and Computer
Graphics Research Group,
University of Münster, Germany

Specular Reflections Through Ray-Tracing

Markus Hadwiger
VR VIS Research Center
Vienna, Austria

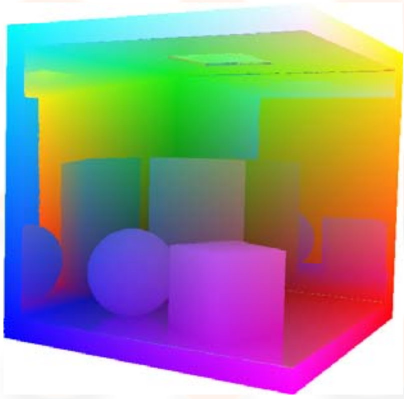
Patric Ljung
Siemens Corporate Research
Princeton, NJ, USA

Christof Rezk Salama
Computer Graphics Group
Institute for Vision and Graphics
University of Siegen, Germany

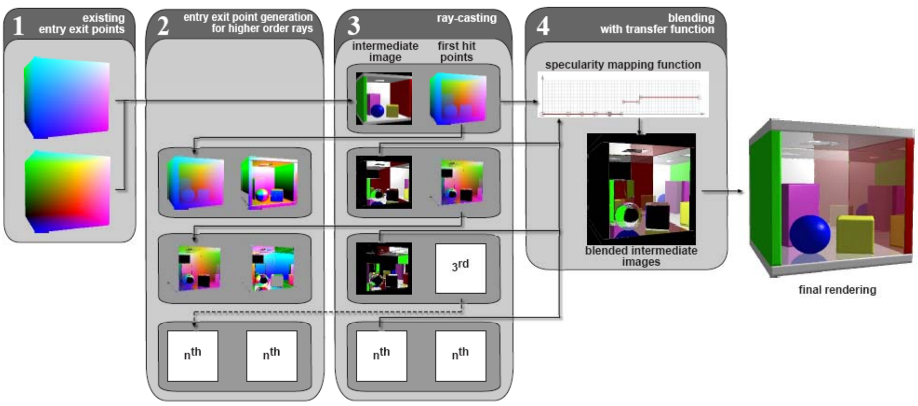
Timo Ropinski
Visualization and Computer
Graphics Research Group,
University of Münster, Germany

Higher Order Rays

- Entry parameter texture can be modified for tracing higher order rays

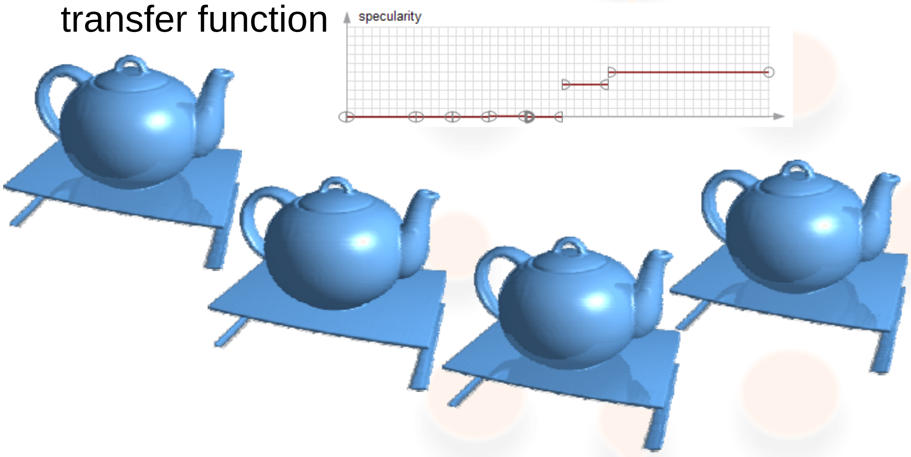


Higher Order Rays



Specular Reflections

- Degree of specularity can be controlled with a transfer function



Soft vs. Hard Shadows

Markus Hadwiger
VR VIS Research Center
Vienna, Austria

Patric Ljung
Siemens Corporate Research
Princeton, NJ, USA

Christof Rezk Salama
Computer Graphics Group
Institute for Vision and Graphics
University of Siegen, Germany

Timo Ropinski
Visualization and Computer Graphics Research Group,
University of Münster, Germany

Why adding Shadows?

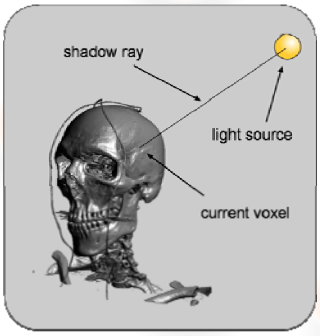
- Adding interactive shadows to volume graphics supports spatial comprehension
- Focus on shadow algorithms integrationable into GPU-based raycasters

Object- vs. Image-Based

- Object-based
  - object-based shadow algorithms like Crow's shadow volumes
  - require polygonal representation of rendered objects
- Image-based
  - representation of shadows in an image
  - shadow mapping by Williams (1978)
  - oppacity shadow maps
  - deep shadow maps (allow transparent objects)

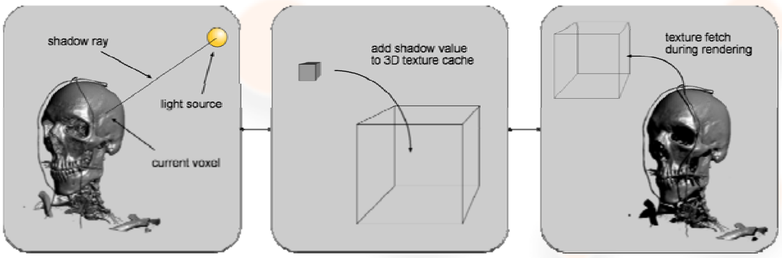
Shadow Rays

- similar to shadow rays in raytracing
  - opaque occluders (similar to first hit raycasting)
  - alpha raycasting
  - 3D-textures can be used for caching results



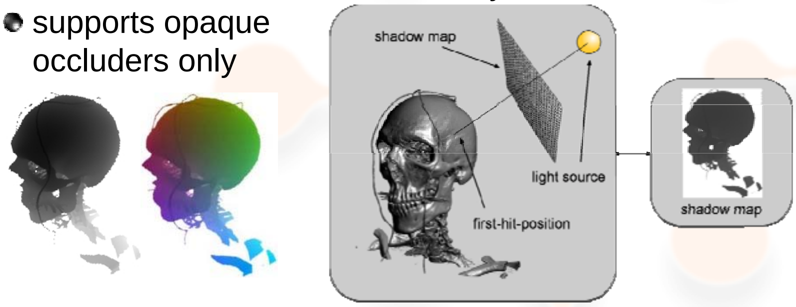
3D Texture Caching

- Shadows can be cached in 3D Textures to gain performance
  - 3D-texture for shadow lookup
  - preprocessing shadow feelers
  - needs to be recomputed on light source or transfer function change



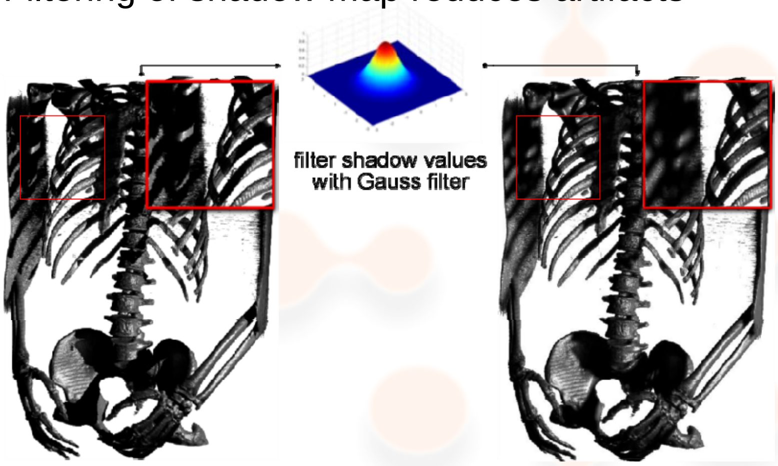
Shadow Mapping

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



Shadow Mapping

- Filtering of shadow map reduces artifacts



Semi-Transparent Shadows

Markus Hadwiger
VR VIS Research Center
Vienna, Austria

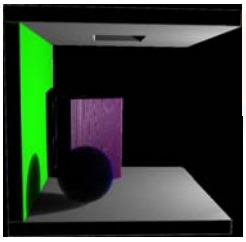
Patric Ljung
Siemens Corporate Research
Princeton, NJ, USA

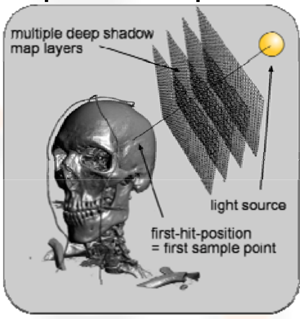
Christof Rezk Salama
Computer Graphics Group
Institute for Vision and Graphics
University of Siegen, Germany

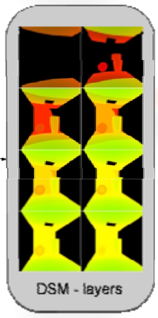
Timo Ropinski
Visualization and Computer
Graphics Research Group,
University of Münster, Germany

Deep Shadow Maps

- Support semi-tranparent occluders by incorporating multiple layers
- Each layer is a pair of depth and transparency
- For each pixel control points of piecewise linear functions are saved

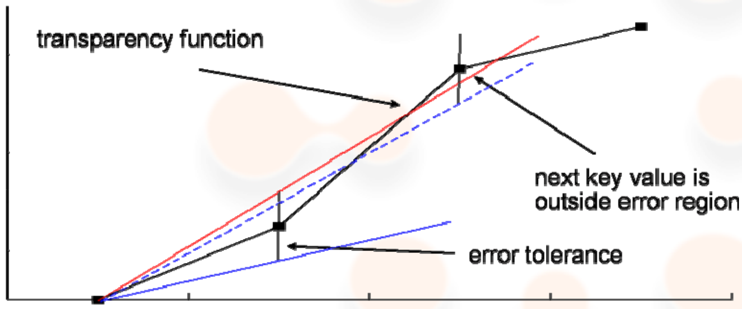






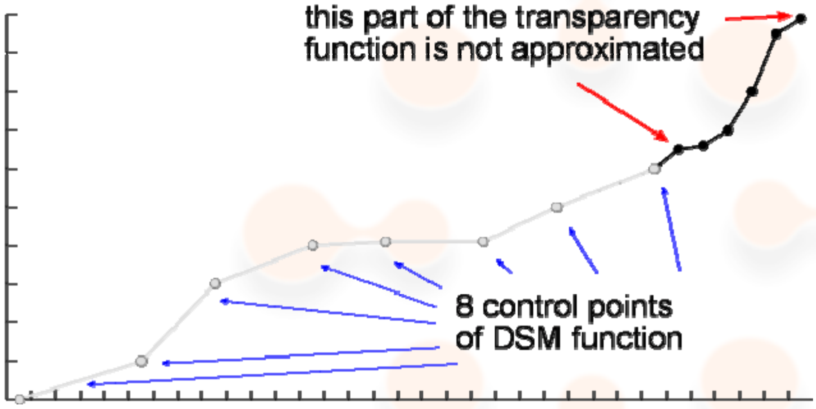
Deep Shadow Maps

- Shadow map constrution
  - At the first hit point, the first key value is saved
  - Based on a error function, further key values are saved

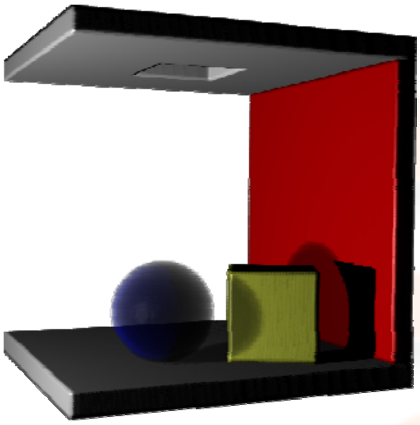


Deep Shadow Maps

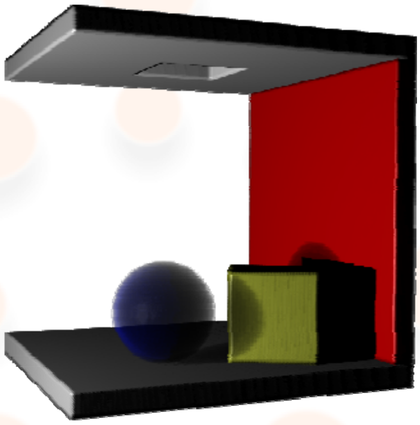
- Original function differs by the choosen error value



Approximation Artifacts



error value of 0.01



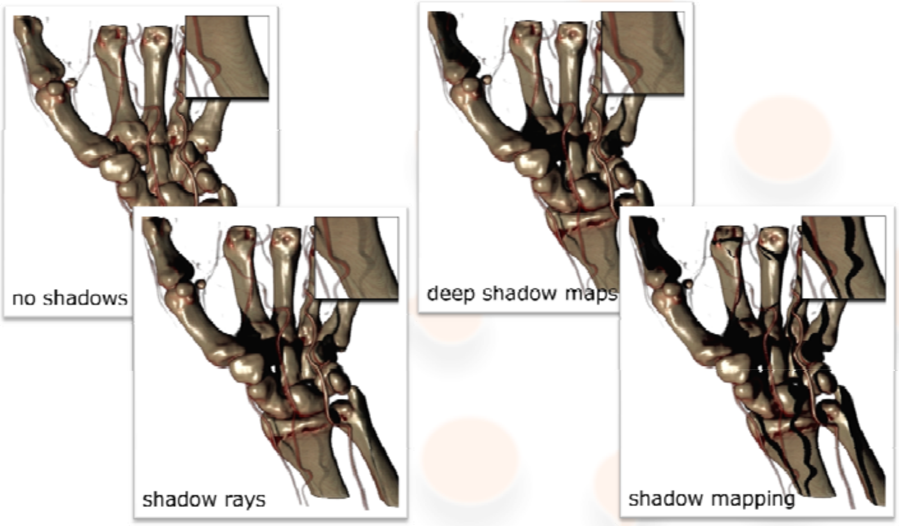
error value of 0.00005

DSM Performance

Shadow Mode	RC	without RC
ShadowRay (B)	10.03	10.03
ShadowRay (A)	10.0	10.0
ShadowRay (B + PP)	5.59	46.0
ShadowMap (B)	29.08	45.5
DeepShadowMap (A)	15.76	34.5
DeepShadowMap (A + PP)	13.06	45.2

- Intel Core2 6600 (2.4GHz), 2GB RAM and an nVidia GeForce 8800GTX

Results



Vis08

Simulation of Color Bleeding

Markus Hadwiger
VR VIS Research Center
Vienna, Austria

Patric Ljung
Siemens Corporate Research
Princeton, NJ, USA

Christof Rezk Salama
Computer Graphics Group
Institute for Vision and Graphics
University of Siegen, Germany

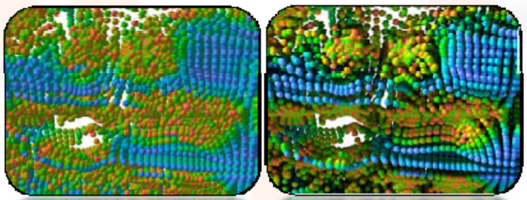
Timo Ropinski
Visualization and Computer
Graphics Research Group,
University of Münster, Germany

Indirect Illumination

- Langer, Bülthoff: *Depth Discrimination from Shading under Diffuse Lighting* [Perception 00]:

  „… depth discrimination under diffuse lighting is superior to that predicted by a classical sunny day model, …“

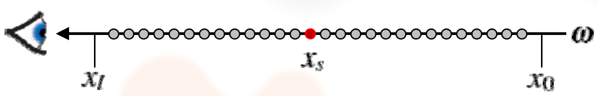
- Gribble, Parker: *Enhancing Interactive Particle Visualization with Advanced Shading Models* [APGV06]



Motivation

- Standard volume rendering integral
  - Captures emission and absorption only

$$I(x_l,\omega) = T(0,l)\cdot I(x_0,\omega) + \int_0^l T(s,l)\cdot E(x_s)ds$$



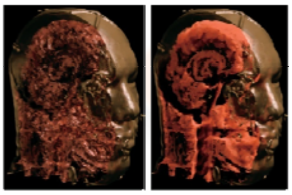
  - No diffuse interreflection (as well as scattering) can be captured

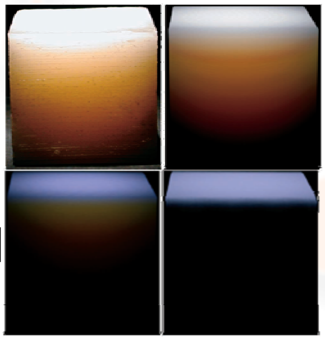
Motivation

- Transfer functions influence displayed structures and therefore light interactions
⇒ Precomputation with surface-based techniques not possible

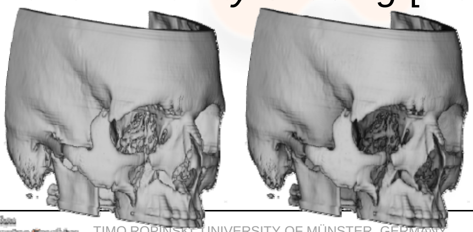
Related Work

- Kniss et al.: *Global Volume Illumination* [TVCG03]





- Stewart: *Vicinity Shading* [Vis03]



Workflow

1 Capture Light Interactions → 2 Compress Lighting Information | 3 Volume Rendering

← Preprocessing | Interactive →

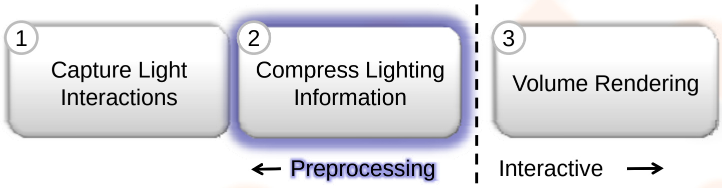
Workflow

1 Capture Light Interactions → 2 Compress Lighting Information | 3 Volume Rendering

← Preprocessing | Interactive →

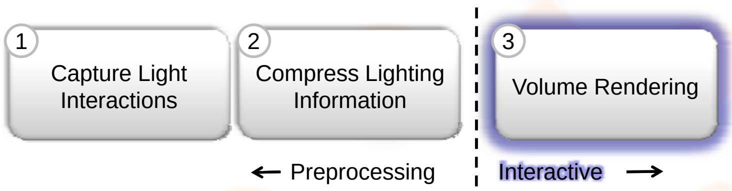
- Light interactions
  - … are captured for the vicinity only
  - … are expressed by using a local histogram

Workflow



- One local histogram for each voxel results in unmanageable data sizes
- Light information is clustered to handle it interactively during rendering

Workflow



- Rendering parameters are changed frequently
- Representative local histograms can be modulated interactively
- Volume rendering requires only two additional texture fetches

TIMO ROPINSKI, UNIVERSITY OF MÜNSTER, GERMANY
ADVANCED ILLUMINATION TECHNIQUES FOR GPU-BASED VOLUME RAYCASTING

Workflow

1 2 3

1. Capture Light Interactions
2. Compress Lighting Information
3. Volume Rendering

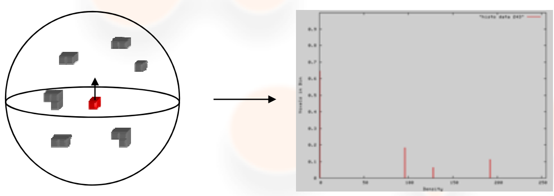
Histogram Generation

1 2 3

- Analyze vicinity of each voxel
- Compute a normalized local histogram w$n = 2^b$ bins

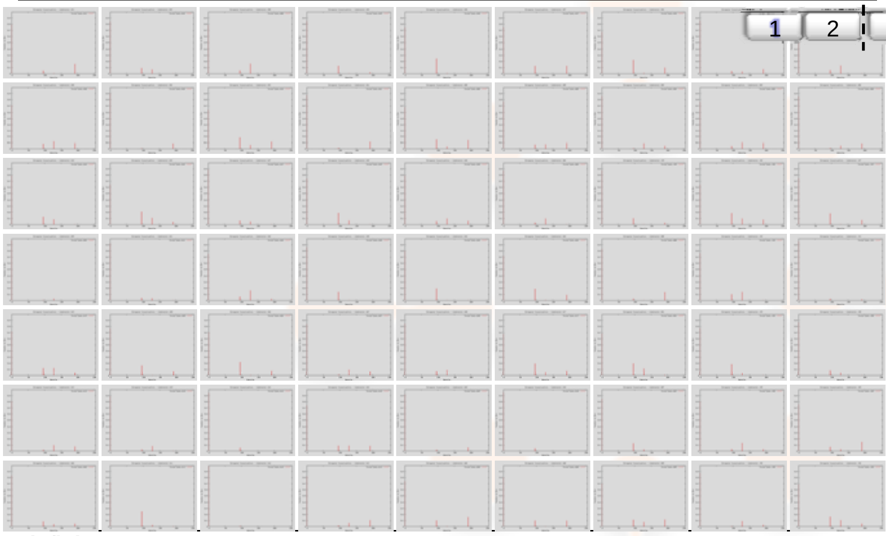
$$LH(x) = (LH_0(x), \ldots, LH_{n-1}(x))$$

$$LH_k(x) = \sum_{\substack{\tilde{x} \in S_r(x) \\ \tilde{x} \neq x}} f_{dist}\left(\frac{|x - \tilde{x}|}{d_{min}}\right) \cdot g(f(\tilde{x}), k)$$



Histogram Generation

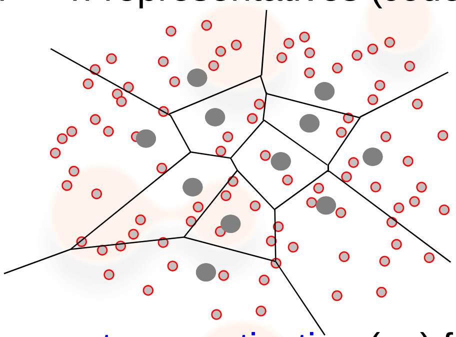
1 2 3



Histogram Clustering

1 2 3

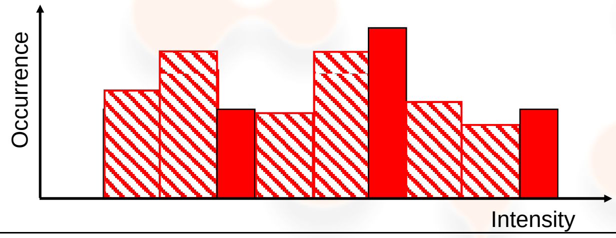
- Given: *n* local histograms (vectors)
- Desired: *m* << *n* representatives (*code book*)



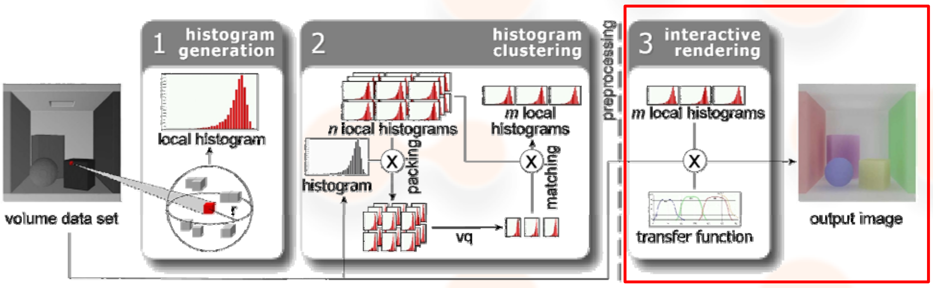
- We exploit a vector quantization (vq) for the clustering

Histogram Compression

- Goal: generate a packed histogram with *j* << *i* bins (*i* = initial number of bins)
- *Iterative splitting* is used to reduce histogram dimensionality
- Example: *i = 12*, *j = 9*



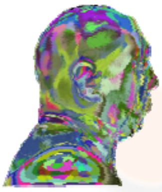
Detailed Workflow

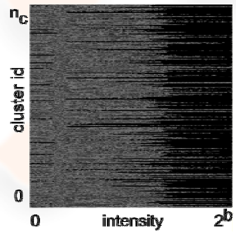


Interactive Rendering

1 2 3

- Two additional texture fetches required
  1. Obtain the cluster ID of the current sample *x*
  2. Fetch the current environment color $E_{env}(x)$





- $E_{env}(x)$ is computed by considering the current transfer function

Isosurface Shading

1 2 3

- Combination with the transfer function

$$O_{env}(x, \bigtriangledown\tau(f(x))) = \frac{1}{\frac{2}{3}\pi r^3} \sum_{0 \le j < 2^b} \tau_\alpha(j) \cdot LH_j(x)$$

- Apply Phong shading by using

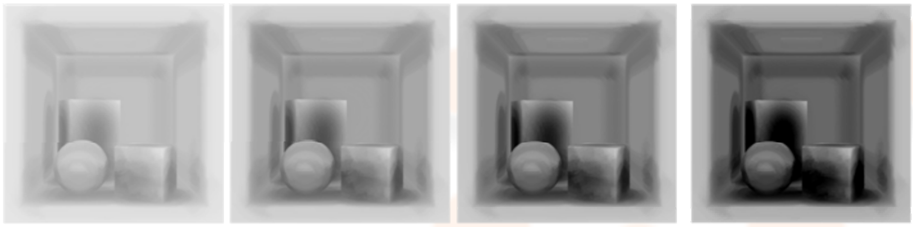
$$I_a(x) = 1.0 - O_{env}(x, \bigtriangledown\tau(f(x))) \cdot Col_{iso}$$

Isosurface Shading

1 2 3

- Occlusion term can be weighted

$$I_a(x) = 1.0 - O_{env}(x, \nabla\tau(f(x))) \cdot Col_{iso} \cdot g$$

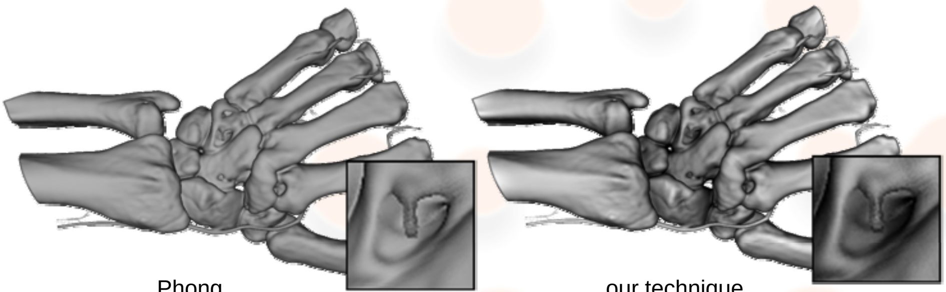


g=0.75 g=1.0 g=1.25 g=1.5

n_c=512

Isosurface Shading

1 2 3



Phong our technique

n_c=2048

Demonstration Video I

n_c=2048

Direct Volume Rendering

- Challenges
  - More than one hue is present
  - Areas with participating media may occur

- Combination with the transfer function

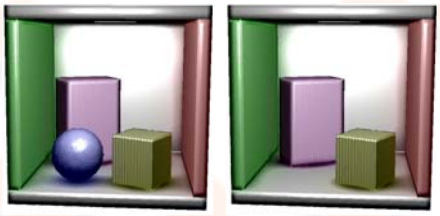
$$E_{env}(x, \triangledown \tau(f(x))) = \frac{1}{\frac{2}{3}\pi r^3} \sum_{0 \leq j < 2^b} \tau_\alpha(j) \cdot \tau_{rgb}(j) \cdot LH_j(x)$$

0 intensity *2^b*

Direct Volume Rendering

- Rendering is done in YUV color space
- Color: Interpolate between E_{env} and $\tau_{rgb}(x)$
  - O_{env} is used as the interpolation factor
- Luminance: minimum of $1.0 - O_{env}$ and $\nabla\tau(f(x))\cdot L$
- Specular highlights can be added

E_{env} only

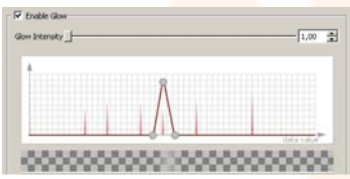


Rendering – 3D *Glow*

- Volumetric glow can be realized by exploiting an additional texture fetch

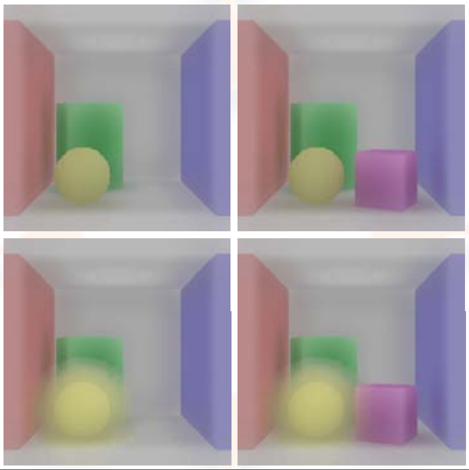
$$E_{env}(x,\nabla\tau(f(x))) = \frac{1}{\frac{2}{3}\pi r^3}\sum_{0\le j<2^b}\tau_\alpha(f(x))\cdot\tau_{rgb}(f(x))\cdot LH_j(x)\cdot h(f(x))$$

- Exploiting a glow mapping function

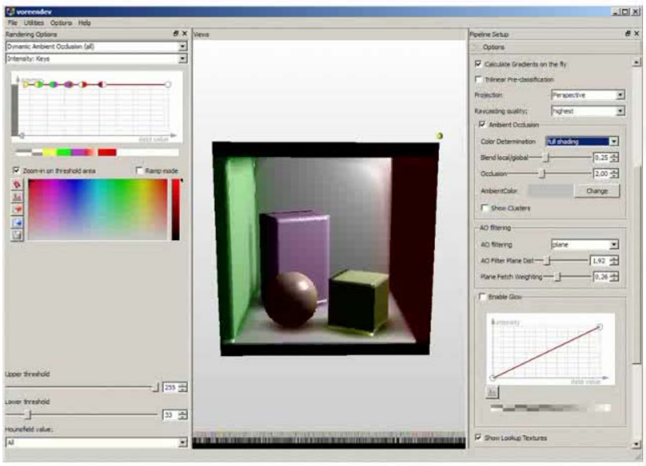


Rendering Wrapup

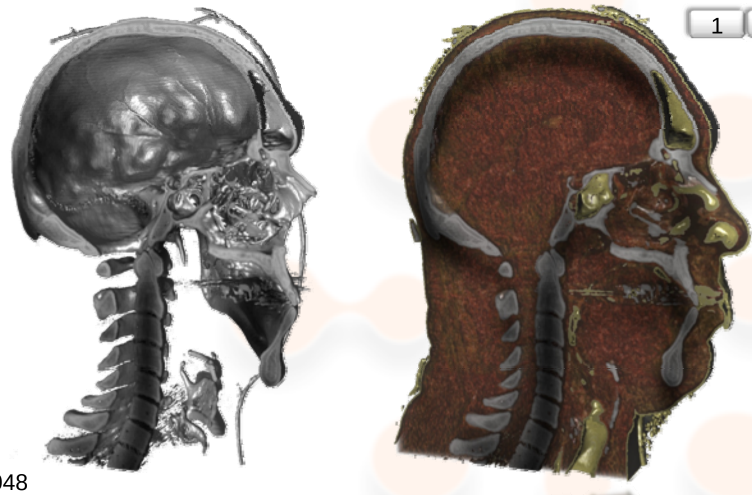
- Indirect illumination can be captured independently of the currently set transfer function



Demonstration Video II

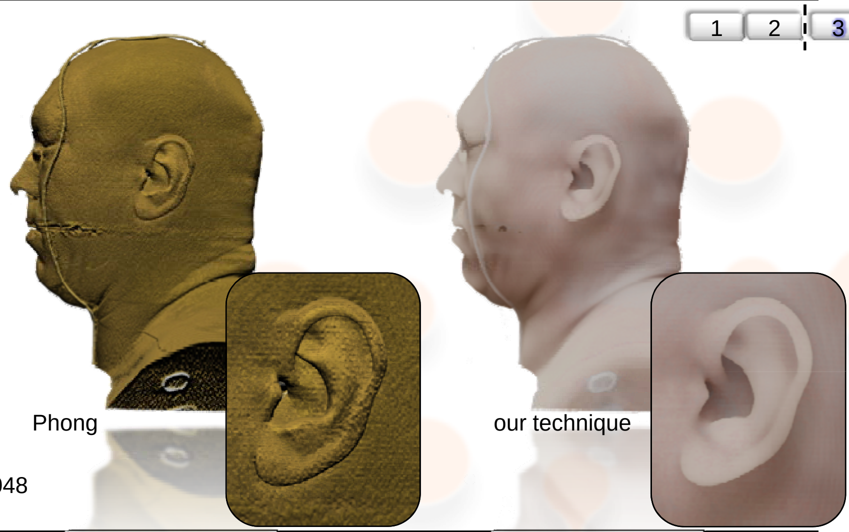


Application Example - DVR



n_c=2048

Application Example - DVR



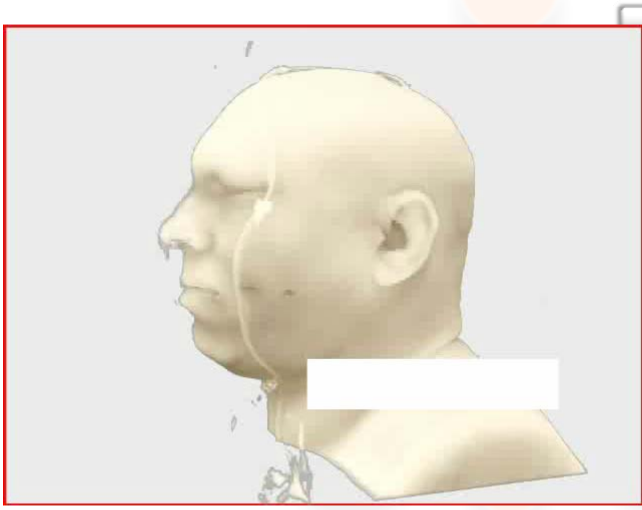
Phong

our technique

n_c=2048

Demonstration Video III

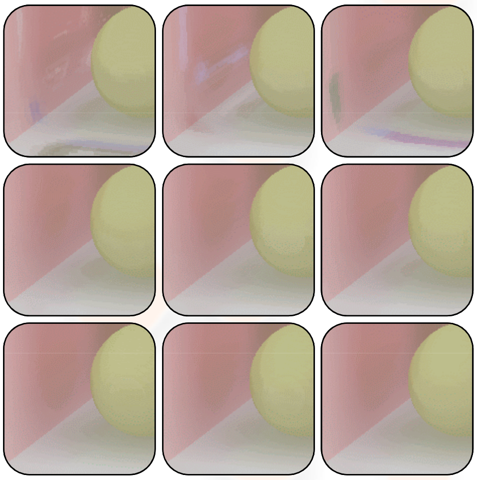
1 2 3



n_c=2048

Varying Distance Metrics

1 2 3



n_c=256

n_c=512

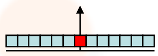
n_c=1024

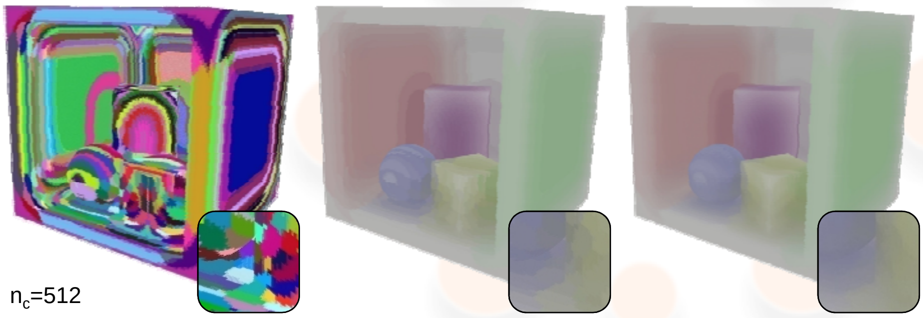
L_{max} L_1 L_2

Surface Filtering

1 2 3

- Too few clusters lead to quantization artifacts
- A surface filter is used to reduce artifacts





n_c=512

Performance

1 2 3

data set	size (voxel³)	sphere size (radius)	hist. gen. (min.)	training parameters (codewords/packed dim)	training (min.)	local histo–gram size	codebook size
Cornell box	128 × 128 × 128	32	236.03	256/16	2.10	2048 MB	0.01 MB
				512/16	7.08		
				1024/16	21.90		
Visible Human head	192 × 192 × 110	12	16.63	2048/64	528.58	3960 MB	0.5 MB
		16	38.71		484.31		
		24	132.40		510.01		
	256 × 256 × 147	16	101.60	2048/64	704.21	9408 MB	0.5 MB
	512 × 512 × 294	32	3025.38		–[a]	75264 MB	0.5 MB
hand	244 × 124 × 257	20	514.31[b]	2048/64	633.80	7593 MB	0.5 MB
feet[b]	128 × 64 × 128	12	15.98	2048/64	320.81	1024 MB	0.5 MB
data sets computed using pre-processing with performance improved implementation							
cloud	256 × 128 × 128	24	6.03	2048/16	5.03	4096 MB	0.5 MB
Visible Human head	256 × 256 × 147	16	10.26	2048/64	61.60	9408 MB	0.5 MB

[a] not calculated. [b] calculated on a machine with 4 × Xeon 2.8.

Implementation

1 2 3

- Preprocessing
  - Realized in C++
  - Parallelization with OpenMP
  - 8 processor cluster system (AMD Opteron 852, 2.6 GHz)
- Rendering
  - OpenGL in combination with GLSL
  - GPU-based ray-casting [Roettger et al. 2003]

Vis08

Advanced Illumination Techniques for GPU-Based Volume Raycasting

Markus Hadwiger
VR VIS Research Center
Vienna, Austria

Patric Ljung
Siemens Corporate Research
Princeton, NJ, USA

Christof Rezk Salama
Computer Graphics Group
Institute for Vision and Graphics
University of Siegen, Germany

Timo Ropinski
Visualization and Computer
Graphics Research Group,
University of Münster, Germany

Vis08

Local Ambient Occlusion in DVR

Markus Hadwiger
VR VIS Research Center
Vienna, Austria

Patric Ljung
Siemens Corporate Research
Princeton, NJ, USA

Christof Rezk Salama
Computer Graphics Group
Institute for Vision and Graphics
University of Siegen, Germany

Timo Ropinski
Visualization and Computer
Graphics Research Group,
University of Münster, Germany

Motivation

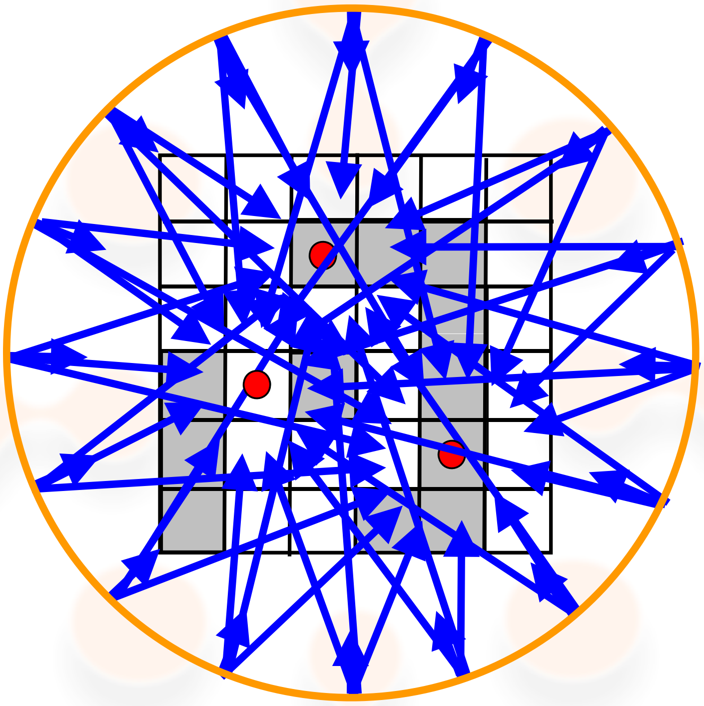
- **Improve medical diagnosis**
  - **Perception of shapes, densities and depth**
- **Diffuse (surface) illumination not suitable for**
  - **Noisy data sets**
  - **Volumetric data**
- **Hernell et al. 2007, Ljung et al. 2004, 2006**

Diffuse illumination

Motivation

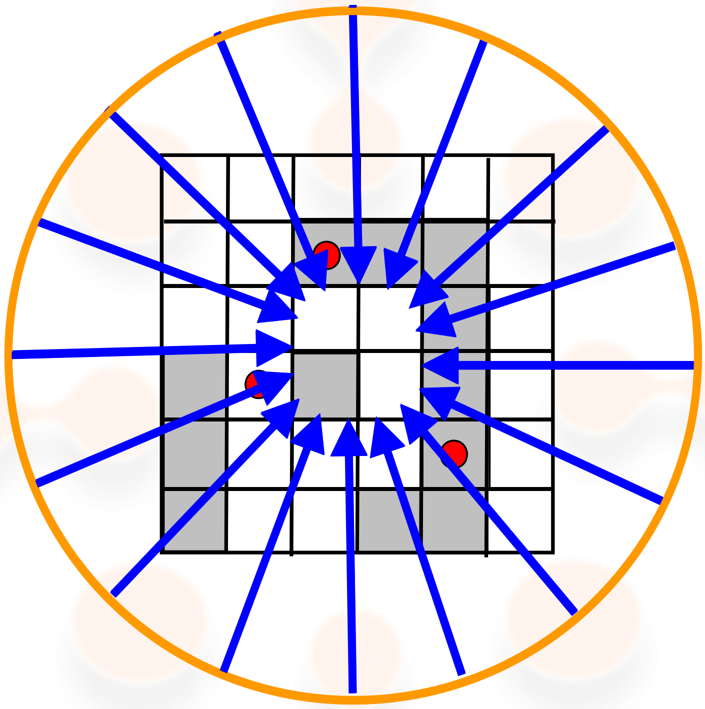
Global Illumination

- **Higher realism**
- **Computationally demanding**
- **Occludes too much?**



Concept of Local Ambient Occlusion

- Compute light contribution for each voxel
- Local spheres



Concept of Local Ambient Occlusion

Diffuse illumination Volumetric Ambient Occlusion

Advantages

- **Efficient local occlusion method for DVR**
  - **Supports Multiresolution Data**
- **Interactive TF-based emission**
  - **Highlight user-specified data ranges**

LAO – single direction

LAO – multiple directions

Local Ambient Occlusion Equations

- **Integration of incoming light from one direction**
- $I_k(x) = \int_a^{R_\Omega} \frac{1}{R_\Omega - a} \exp(-\int_a^s \tau(u)du)ds$
- $I_k(x) = \sum_{m=0}^{M} \frac{1}{M} \prod_{i=0}^{m-1} (1 - \alpha_i)$ **n K number of directions**

$$I(x) = I_{\text{bias}} + \frac{1}{K} \sum_k^K w_k I_k(x)$$

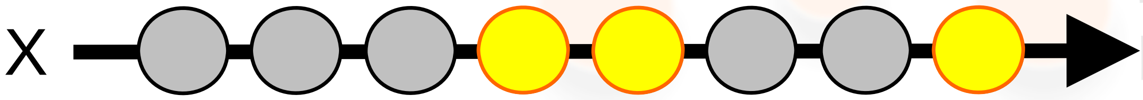
Light Contribution

$$I_k(x) = \sum_{m=0}^{M} \boxed{\frac{1}{M}} \prod_{i=0}^{m-1} (1 - \alpha_i)$$

Light is contributed only at the boundary

Light is contributed at each sample point

LAO with Emission

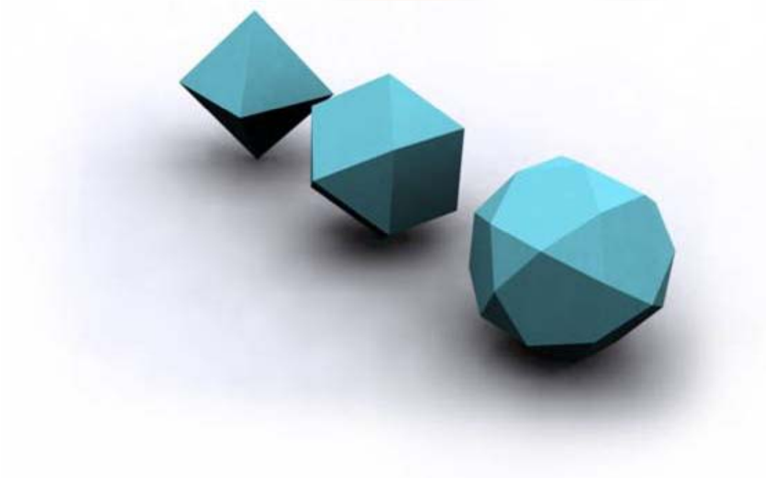


- **Affect intensity and color of x**
- **Emissive component in the TF**
- **Add C_E (color light emission) to the integral**

$$I_k(x) = \int_a^{R_\Omega} \frac{1 \boxed{+ c_E(s)}}{R_\Omega - a} \exp(-\int_a^s \tau(u)du)ds$$

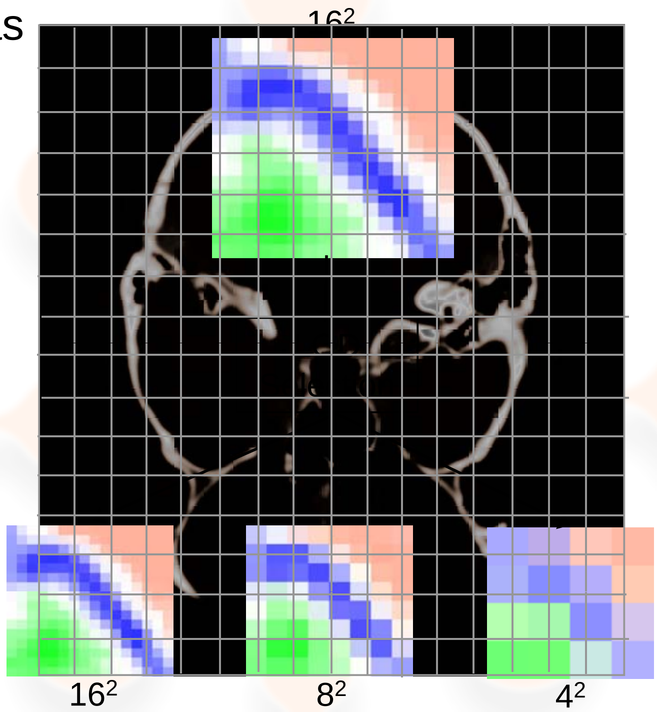
Ray directions

- Dynamically configured
- Subdividing a tetrahedron, isocahedron or octahedron

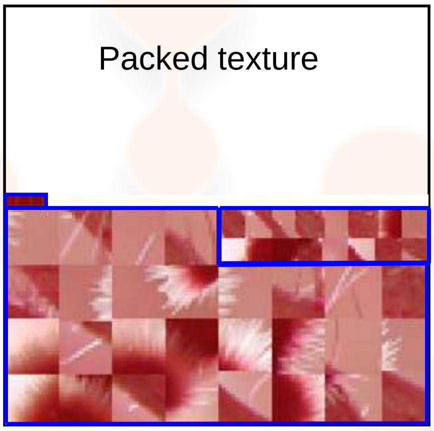


Multiresolution volume

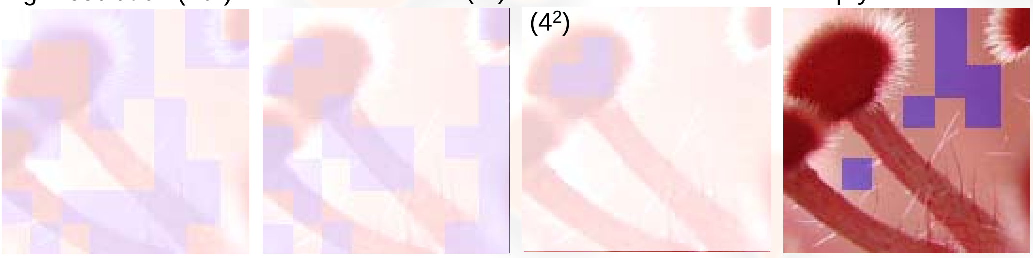
- Superfluous sampling in areas of low occlusion contribution
- Flat Multiresolution Blocking (Ljung et al. '04)
- Preprocessing stage
  - 16^3 voxels are organized into a multiresolution representation
- TF based LOD selection
  - Optimizing the resulting image quality



Multiresolution volume



High resolution (16^2) | Middle resolution (8^2) | Low resolution (4^2) | Empty



Multiresolution LAO

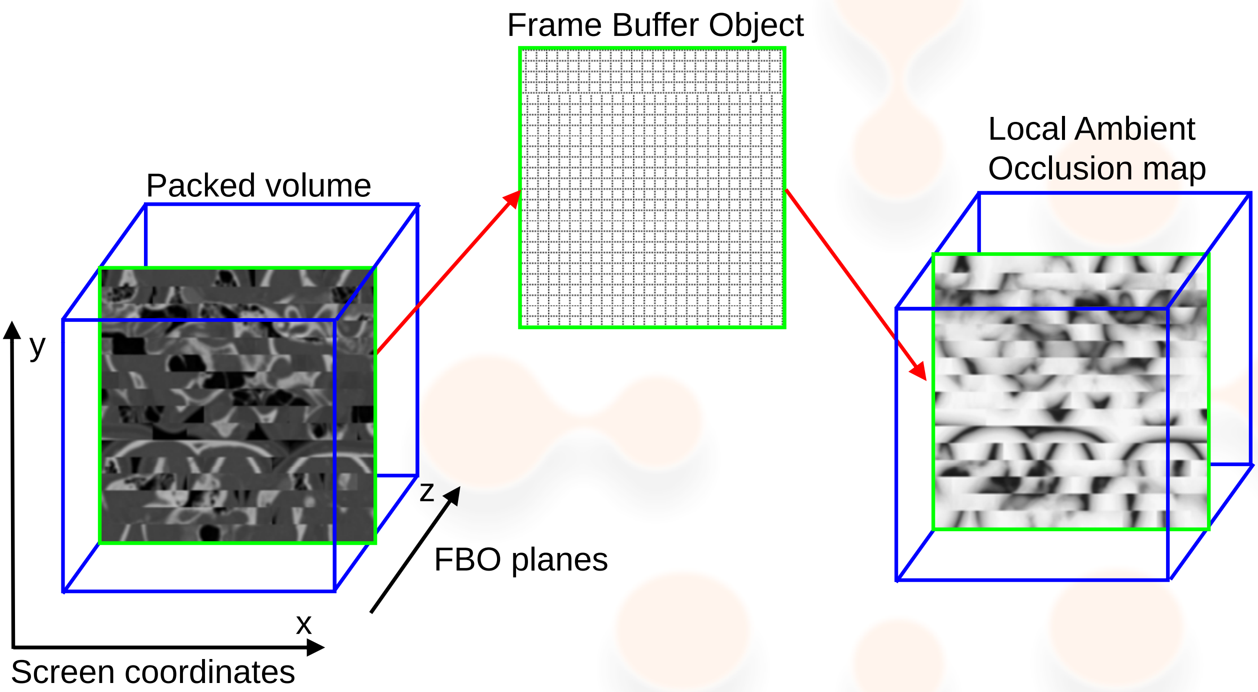
- **Compute LAO directly in packed coordinates**
- **Reduce the number of voxels to compute**
  - **Empty space skipping (Volume block < 4^3)**
- **Increase the sampling distance for low resolution blocks**

Local Ambient Occlusion Pipeline

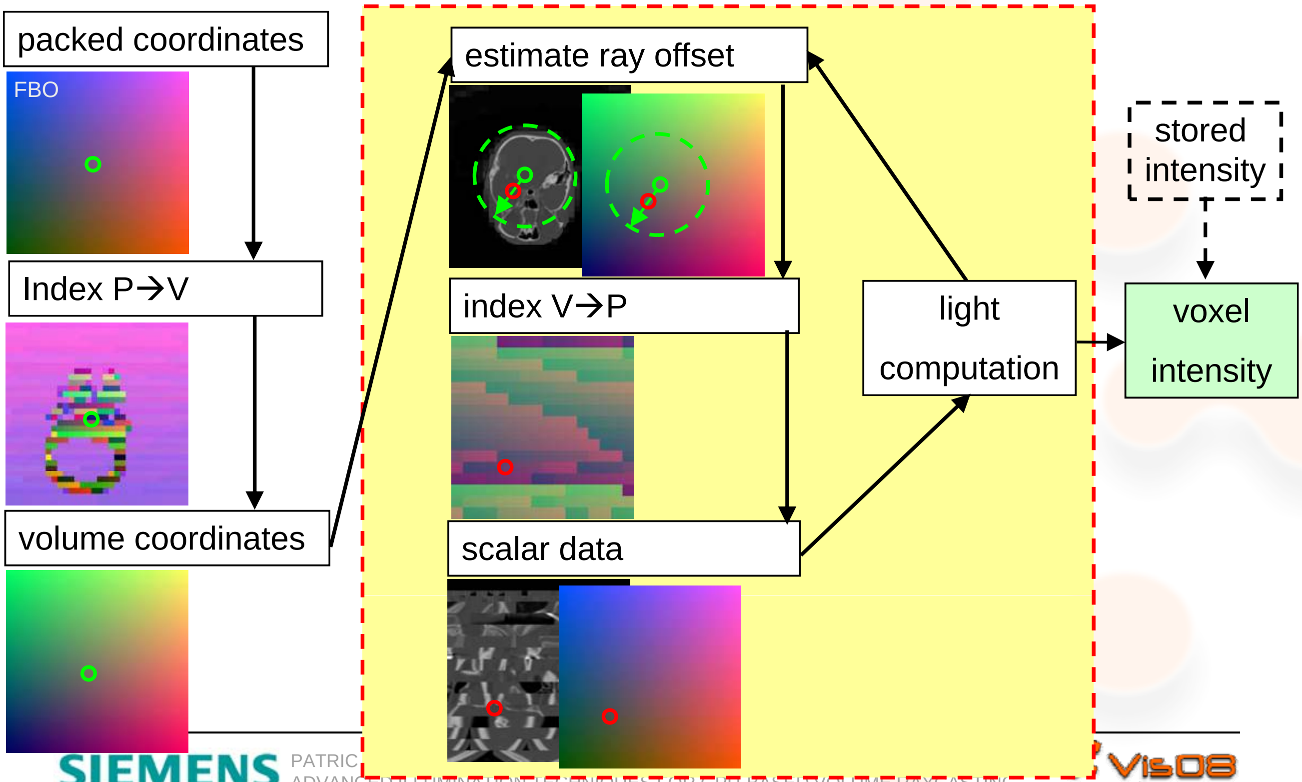
1. LAO computation → LAO map

2. Ray casting with LAO look up

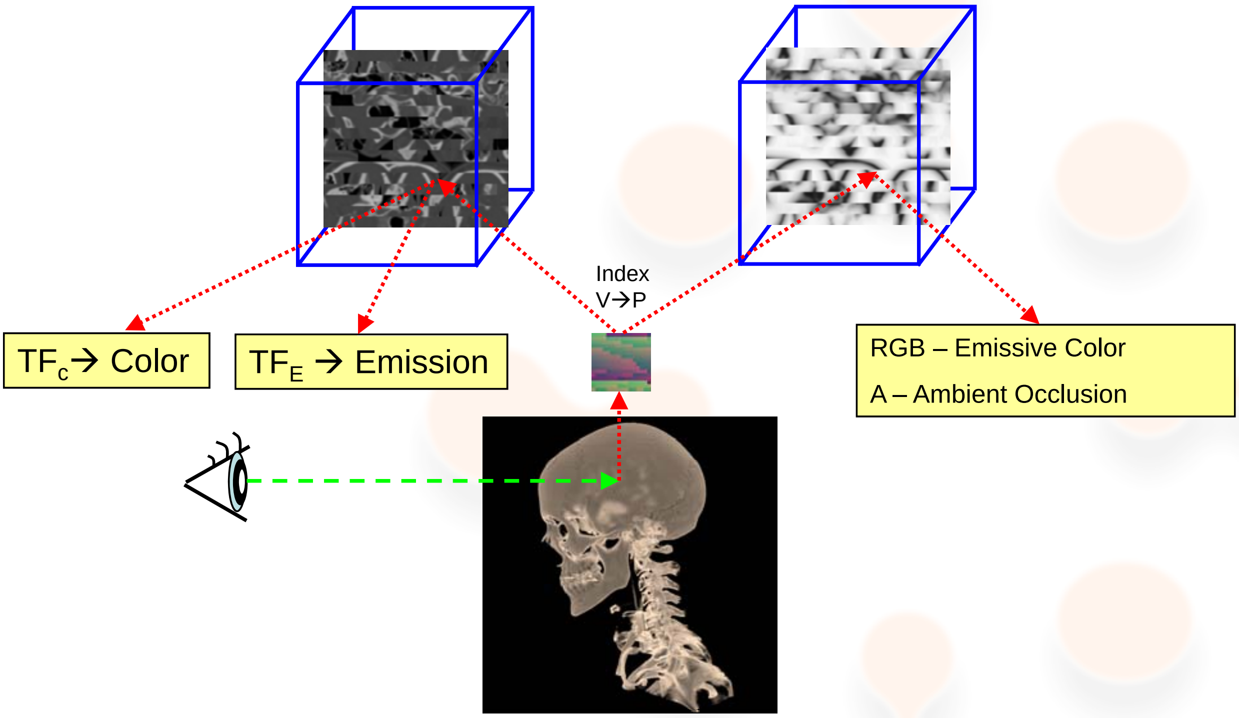
LAO Computation



Per-fragment Pipeline



Final Ray Casting



Results

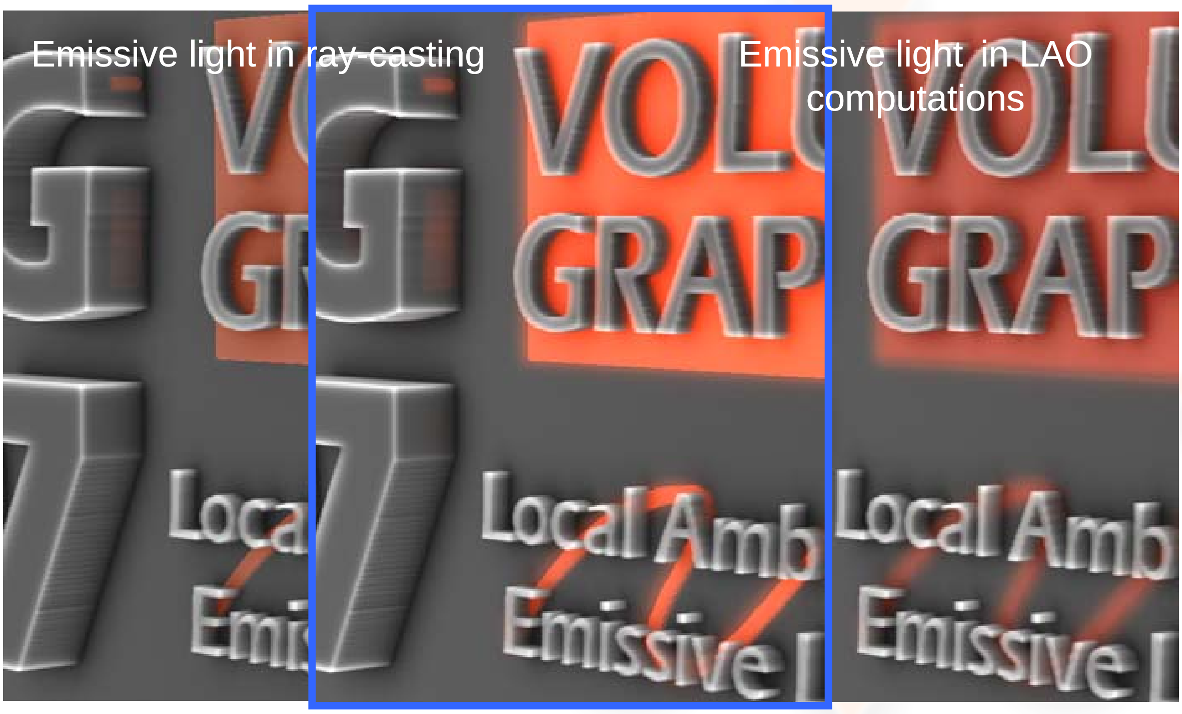
Diffuse Illumination

LAO with 1 direction

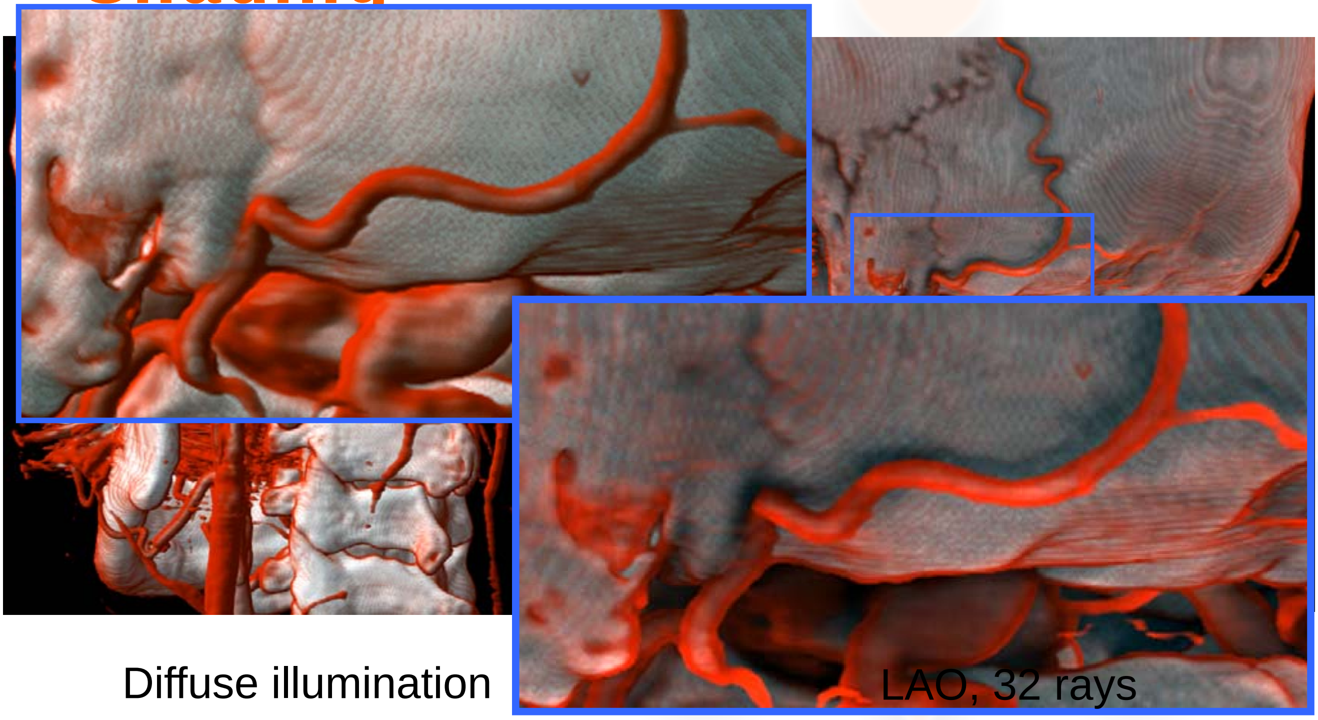
LAO with 8 directions

- NVIDIA GF8800 Ultra
- 768 MB of graphics texture memory
- 51 ms/frame to update one ray in the LAO map
- 27 FPS

Results

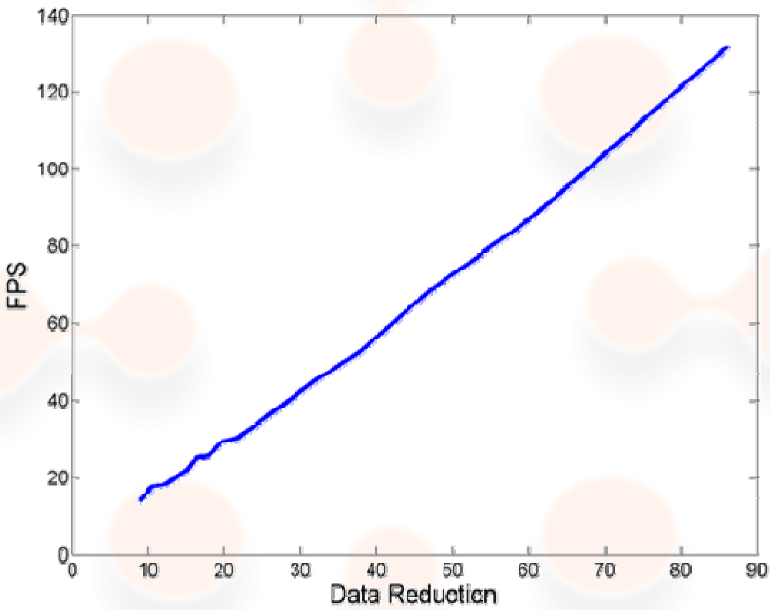


Comparison w/ Diffuse Shading



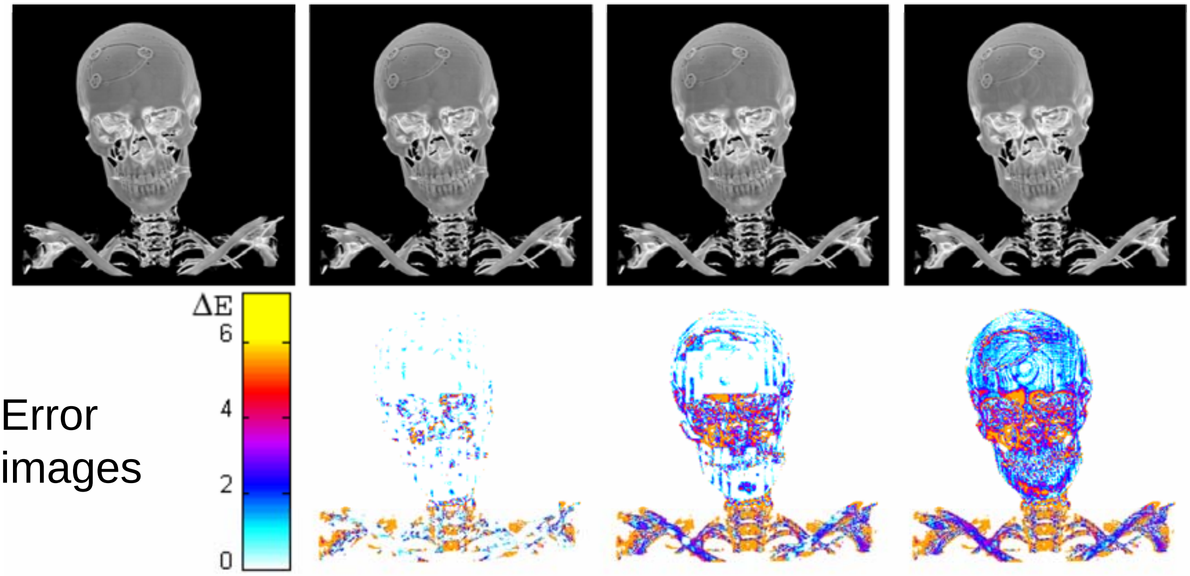
Multiresolution Speed-up

- Volume of 512^3 voxels
- Super-linear performance increase
  - Less voxels to compute
  - Increased step length for low resolution blocks



Results - Multiresolution

Data reduction: 8.5 : 1 | 15.7 : 1 | 31.4 : 1 | 92.8 : 1

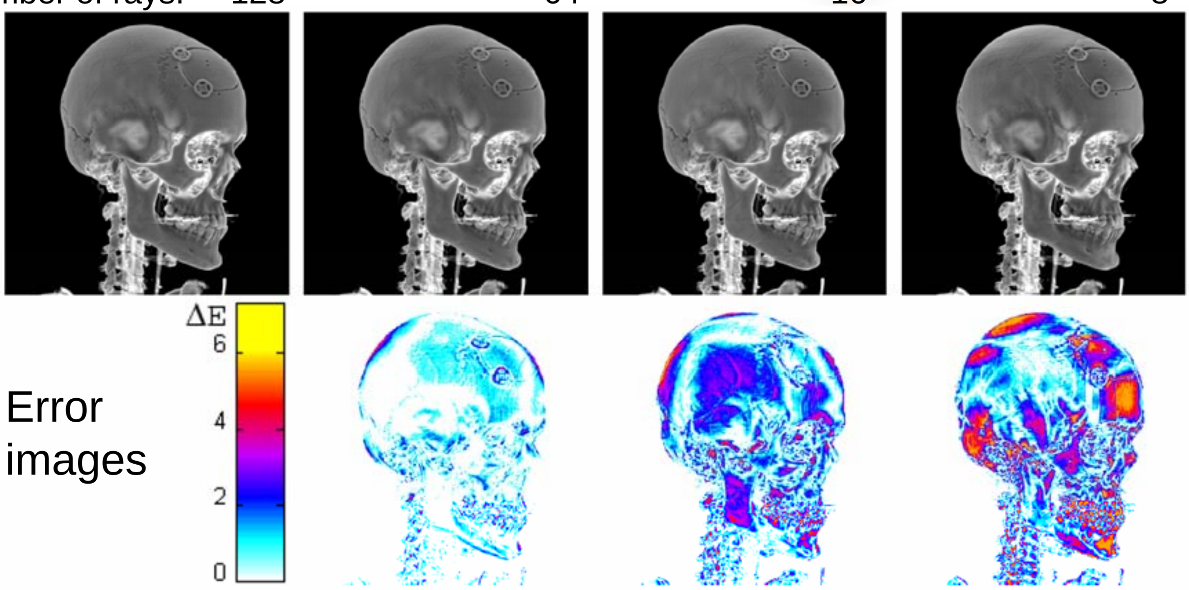


Error images

Mean error: 0.6 | 1.2 | 1.6

Varying Number of LAO Rays

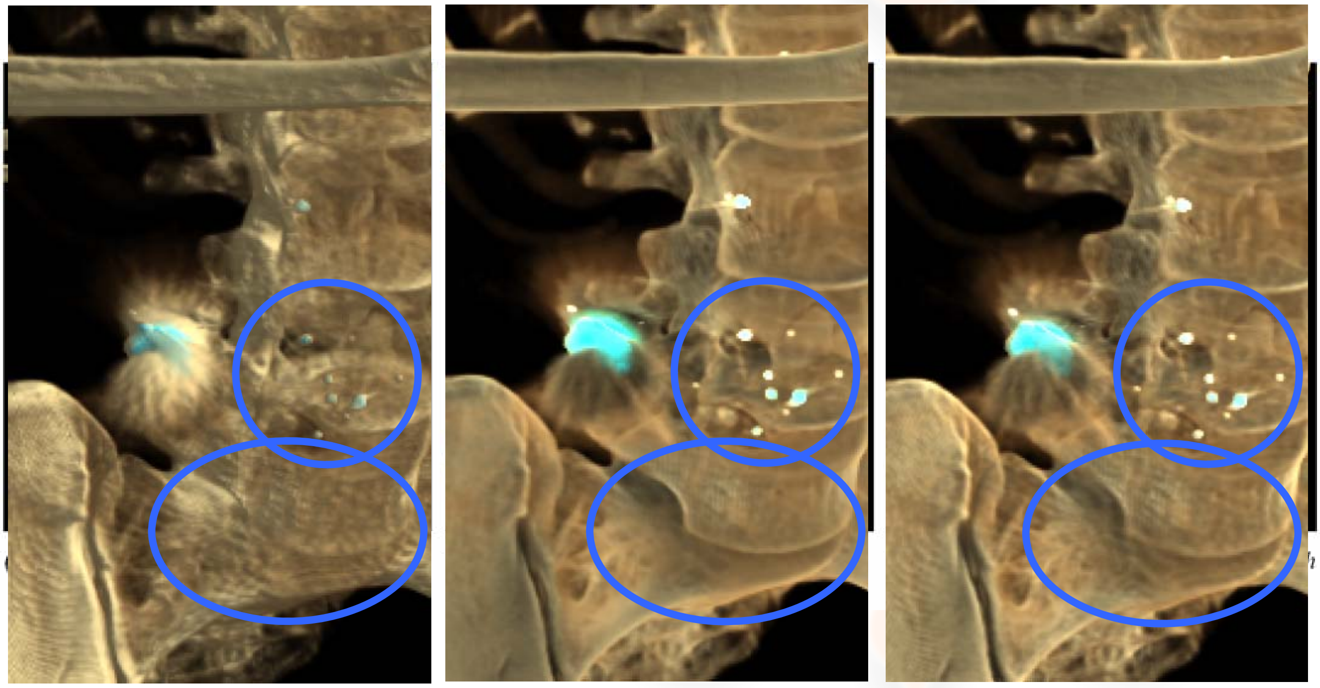
Number of rays: 128 | 64 | 16 | 8



Error images

Mean error: 0.3 | 0.7 | 0.9

Application to Virtual Autopsy Case



Vis08

Global Light Propagation

Markus Hadwiger
VR VIS Research Center
Vienna, Austria

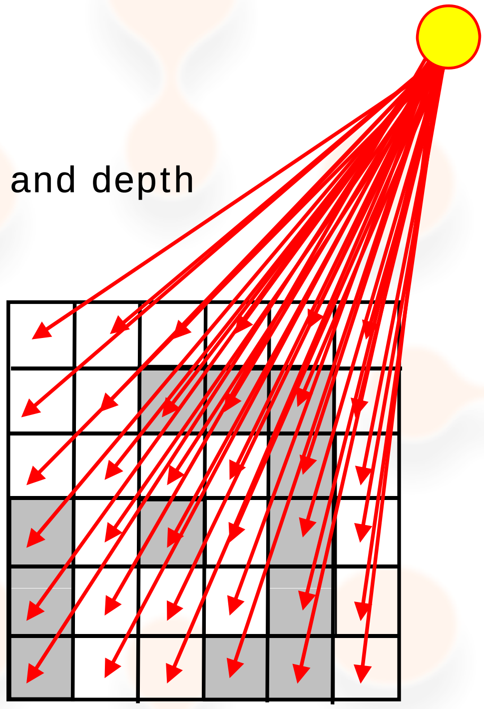
Patric Ljung
Siemens Corporate Research
Princeton, NJ, USA

Christof Rezk Salama
Computer Graphics Group
Institute for Vision and Graphics
University of Siegen, Germany

Timo Ropinski
Visualization and Computer Graphics Research Group,
University of Münster, Germany

Motivation

- Improved medical diagnosis
  - Perception of shapes, densities and depth
- Correct global integration
  - Computationally demanding
- Need for real-time performance of global lighting in DVR



Motivation 2

- **Multiresolution data sets**
  - **Exploit Data Reduction to speed-up Global Lighting.**
- **Existing Techniques has limited applicability for Raycasting**
  - **Adaptive ray-space sampling**
  - **Multiresolution rendering**

Main Advantages

- **An efficient approximation of volumetric light propagation from a point light source**
- **A fast approach for refinements of locally in-scattered light contribution in high resolution**
- **Support for interactive and arbitrary positioning of the light source**

Theory: DVR Integral

The direct volume rendering integral

$$I(D) = I_0 \cdot e^{-\int_0^D \tau(t)dt} + \int_0^D g(s) \cdot e^{-\int_s^D \tau(t)\mathrm{d}t} \mathrm{d}s$$

where

$$g(s) = c(s) \cdot \alpha(s)$$

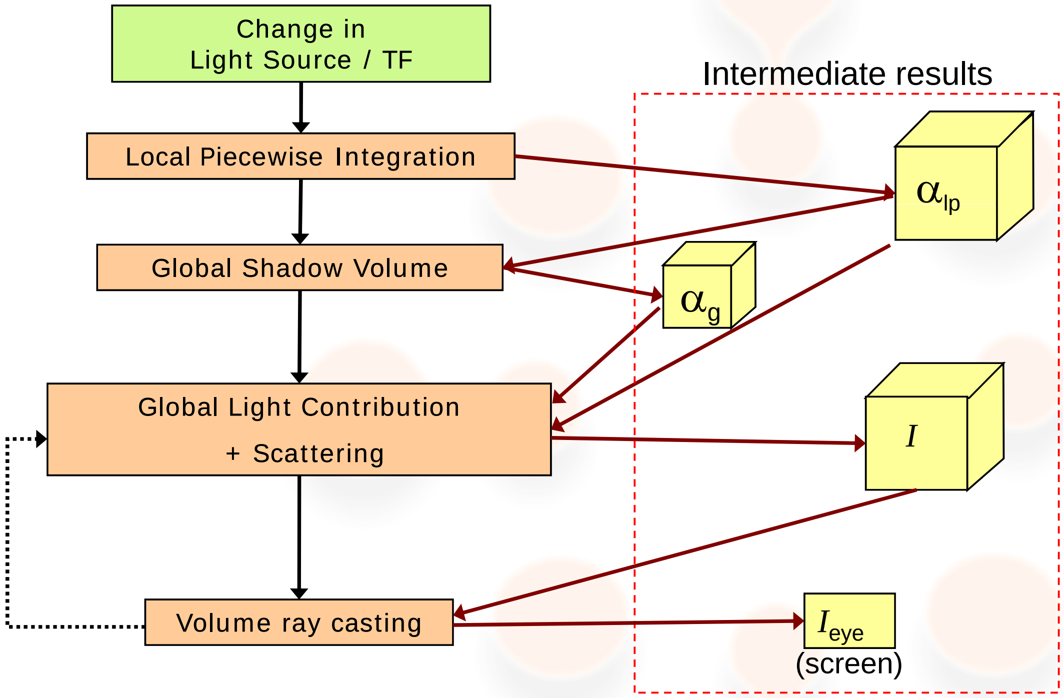
for an unshaded rendering. $c(s)$ **is the color and** $\alpha(s)$ **is the opacity at each sample point,** s**.**

More advanced illumination:

$$g(s) = I(s) \cdot c(s) \cdot \alpha(s)$$

where $I(s)$ **is a combination of direct and in-scattered intensities.**

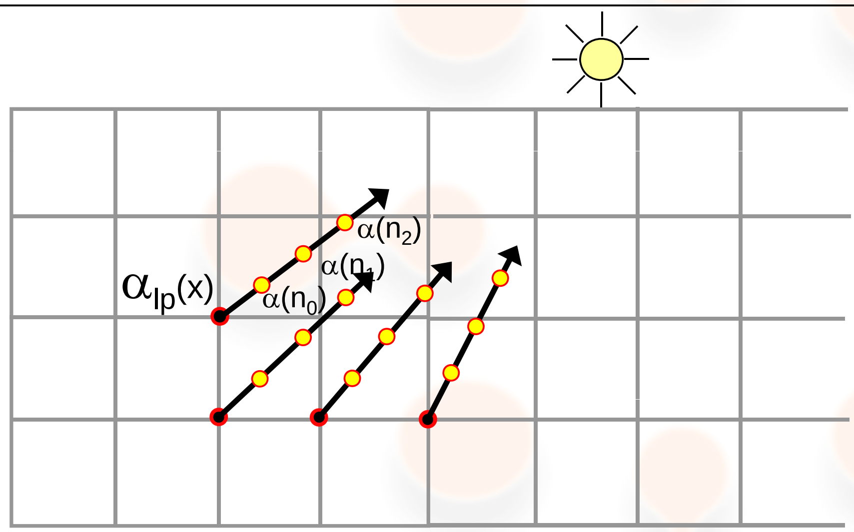
Pipeline: Algorithm Outline



Piecewise Integration

$$T_{lp}(x) = e^{-\int_x^{x'} \tau(u)du}$$

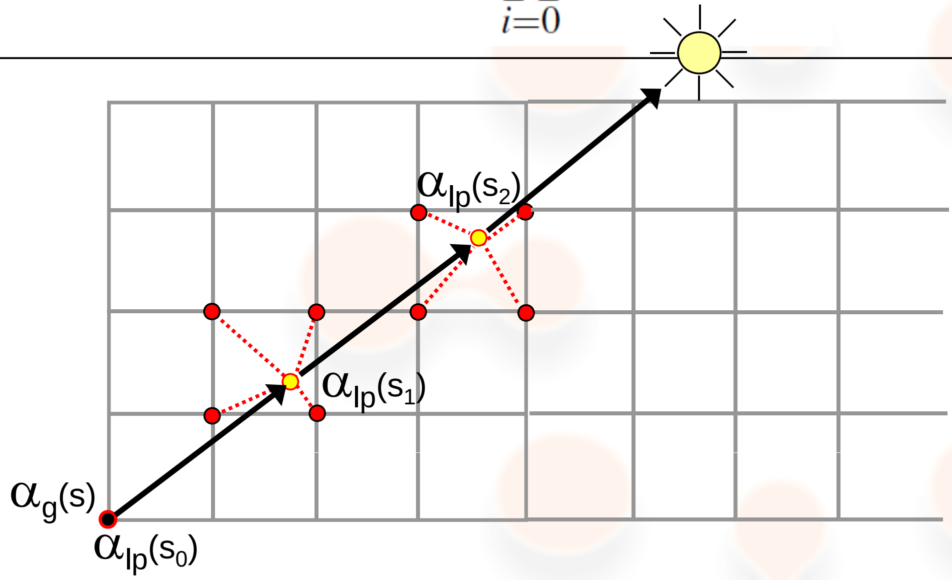
$$\alpha_{lp}(x) = 1 - \prod_{n=0}^{M}(1-\alpha(n))$$



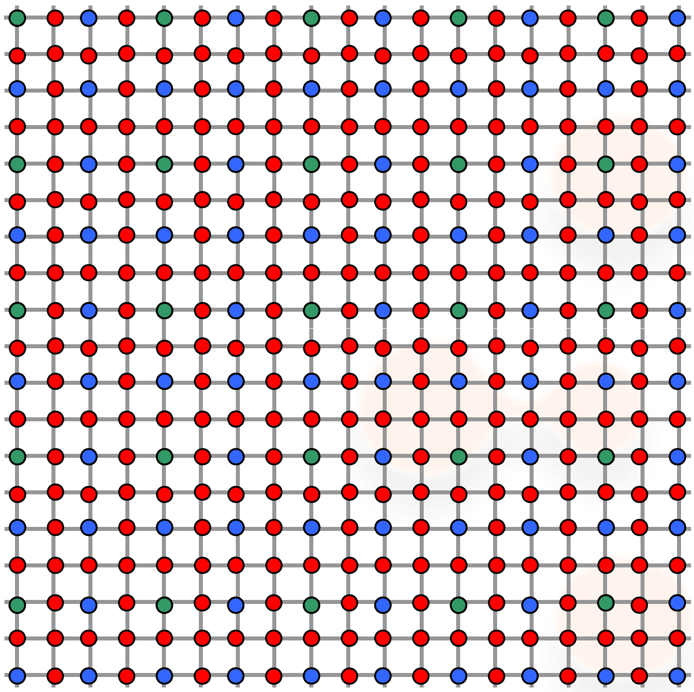
SIEMENS PATRIC LJUNG, SIEMENS CORPORATE RESEARCH, PRINCETON, USA
ADVANCED ILLUMINATION TECHNIQUES FOR GPU-BASED VOLUME RAYCASTING
Vis08

Piecewise Integration

$$\alpha_g(s_0) = 1 - \prod_{i=0}^{k}(1 - \alpha_{lp}(s_i))$$



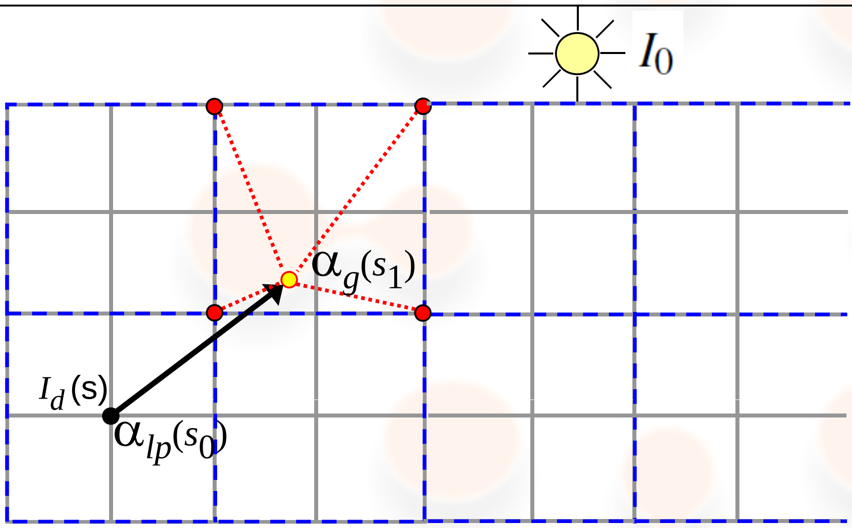
Speed-up of global integration



- Opacity is computed for each local segment originating at red points
- Global integration using the pre-computed piecewise integration can be computed for fewer points
- Leading to a Shadow Volume Representation (SVR)
- Note that the integration for each point still is at high resolution

Improving Accuracy of Direct Light

$$I_d(s_0) = I_0 \cdot (1 - \alpha_g(s_1)) \cdot (1 - \alpha_{lp}(s_0))$$

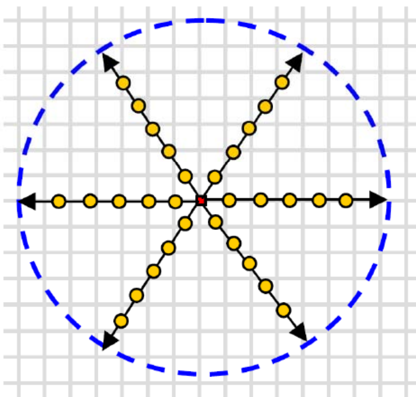


In-scattering

$$I_e(s_0) = \int_{s_0}^{R_\Omega} q(s) e^{-\int_{s_0}^{r} \tau(t)\mathrm{d}t} \mathrm{d}r$$

$$I_d(s_0) = I_0 \cdot (1 - \alpha_g(s_1)) \cdot (1 - \alpha_{lp}(s_0))$$

- **Emittance integral**
- **Similar to Local Ambient Occlusion (LAO) as described earlier**



In-scattering

$$I_s(s_0) = \int_\Omega \varphi(\omega) \int_{s_0}^{R_\Omega} I_d(r) \cdot e^{-\int_{s_0}^{r} \tau(t)\mathrm{d}t} \mathrm{d}r\mathrm{d}\omega$$

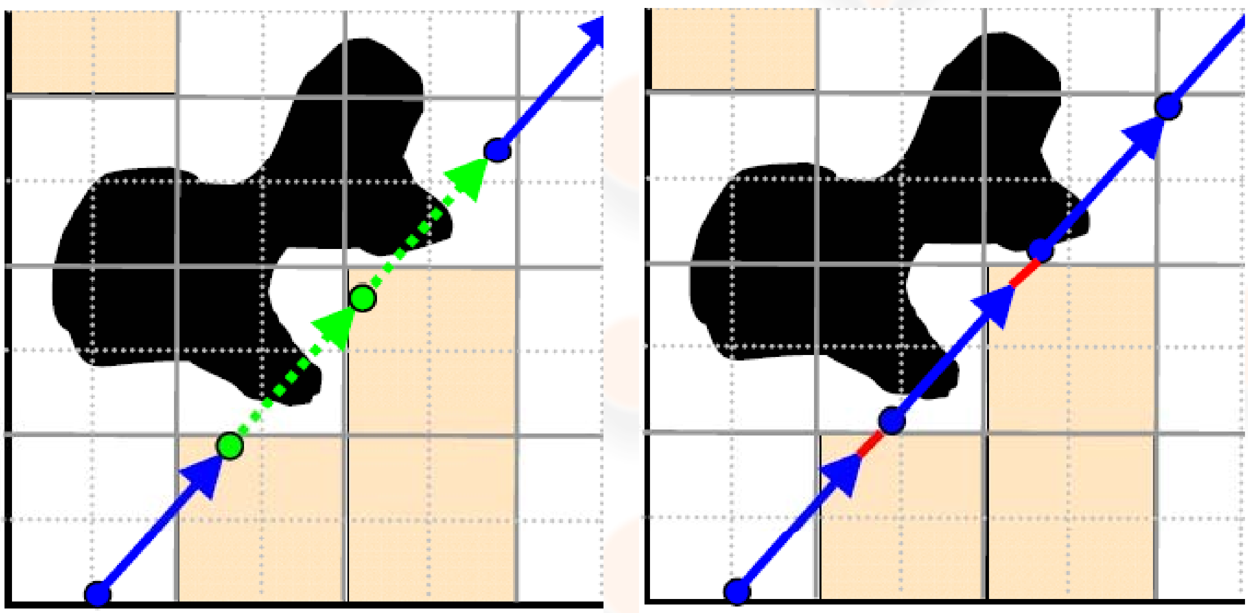
- **Discretized, compositing scheme**

$$I_s = \sum_{j=0}^{J} \varphi_j \sum_{m=0}^{M} I_d(s_m) \prod_{i=0}^{m-1} (1 - \alpha(s_i))$$

- **Progressive refinement, 1 direction per pass**

$$I_{blended} = \frac{1}{j} \cdot I_{final} + (1 - \frac{1}{j}) \cdot I_{\mathrm{stored}}$$

Empty Space Handling



Rendering Pass

$$I_{eye} = \sum_{k=0}^{n} g_k \prod_{p=0}^{k-1} (1 - \alpha(s_p))$$

$$g_k = \underbrace{I_{bias} + (1 - I_{bias}) \cdot I_s(s_k)}_{I_{final}} \cdot c(s_k) \cdot \alpha(s_k)$$

- I_{bias} **allows for setting the minimum light for a volume sample**

Final Result: Global+LAO

Compare w/ Diffuse Surface

Gradient based local lighting

Novel global + ambient lighting

Results: Translucency Effects

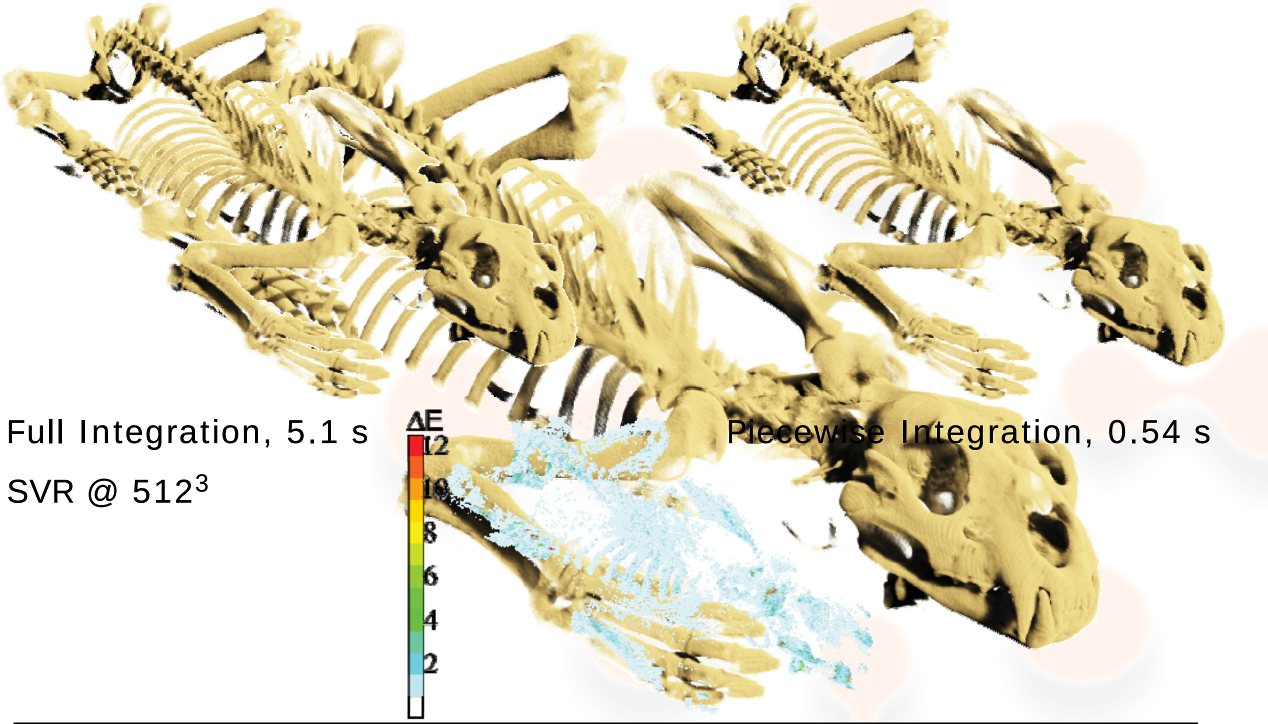
$R_{\Omega} = 16,\ I_{bias} = 0$

$R_{\Omega} = 48,\ I_{bias} = 0$

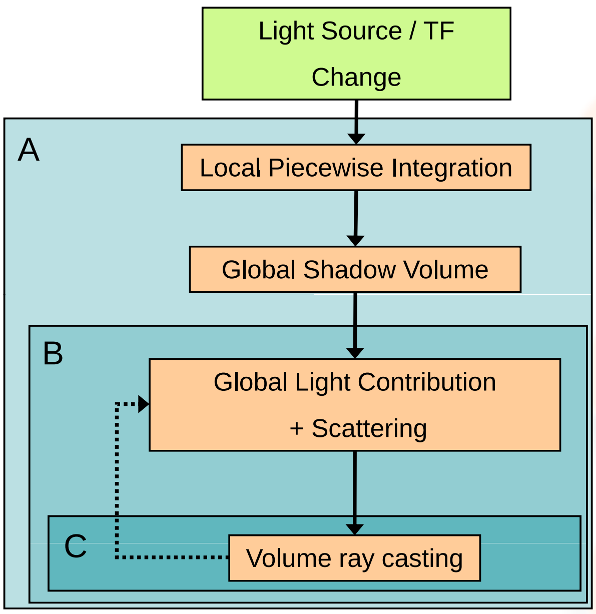
$R_{\Omega} = 16,\ I_{bias} = 0.2$

$R_{\Omega} = 48,\ I_{bias} = 0.2$

Comparison: Full integration



Performance Timing



Piecewise segment length (voxels)	A 32^3	A 64^3	A 128^3	A 256^3
4	261	267	439	1428
8	284	297	373	552
16	331	339	380	641
32	436	439	463	862

milliseconds

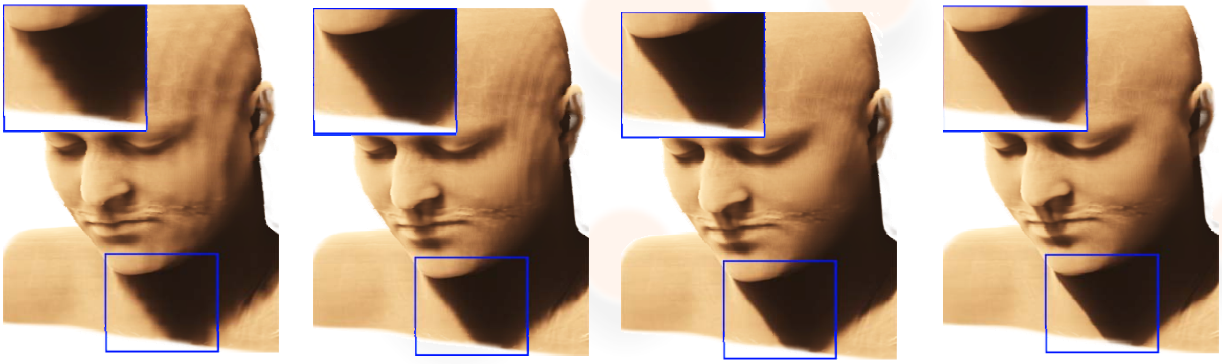
Data reduction	A 32^3	A 256^3	B	C
8.9:1	284	552	233	68
14.8:1	178	515	145	68
22.1:1	121	403	96	48
35.2:1	81	365	62	46

milliseconds

Original volume: 512^3 voxels
Datareduction: 8.9:1
Segment Length: 16
Viewport: 1024x1024

Shadow Volume Sensitivity

Original volume: 512^3 voxels, Datareduction: 8.9:1, Segment Length: 16 voxels



32^3 64^3 128^3 256^3

Summary

- **Efficient computations, adapting to data reduction**
  - **Interactive frame rates**
- **Volumetric Global Lighting and Ambient Occlusion**
- **Improved perception of local features, depth, and 3D structures**
- **Can handle color-bleeding and emissive tissues**
- **Gradient Free Shading**
  - **Good for noisy data sets**

References

- Frida Hernell, Patric Ljung, and Anders Ynnerman. Efficient Ambient and Emissive Tissue Illumination using Local Occlusion in Multiresolution Volume Rendering. In *Proc. Of Eurographics/IEEE Volume Graphics 2007*.
- Frida Hernell, Patric Ljung, and Anders Ynnerman. Interactive Global Light Propagation in Direct Volume Rendering using Local Piecewise Integration. In *Proc. Of Eurographics/IEEE Volume Graphics 2008*.
- Patric Ljung, Claes Lundström, Ken Museth, and Anders Ynnerman. Transfer Function Based Adaptive Decompression for Volume Rendering of Large Medical Data Sets. In *Proc. of IEEE Volume Visualization and Graphics 2004*.
- Patric Ljung, Claes Lundström, and Anders Ynnerman. Multiresolution Interblock Interpolation in Direct Volume Rendering. In *Proc. of Eurographics/IEEE Symposium on Visualization 2006*.
- Patric Ljung. Adaptive Sampling in Single Pass, GPU-based Raycasting of Multiresolution Volumes. In *Proc. of Eurographics/IEEE Volume Graphics 2006*.

Scattering Effects

Markus Hadwiger
VR VIS Research Center
Vienna, Austria

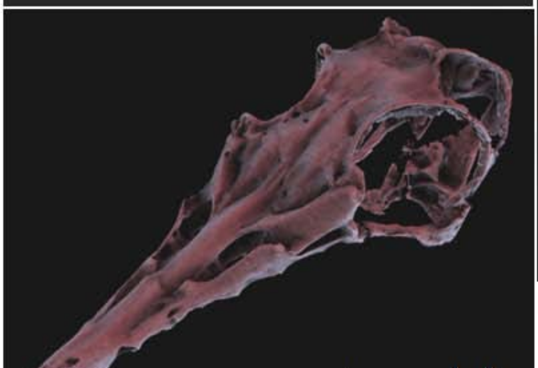
Patric Ljung
Siemens Corporate Research
Princeton, NJ, USA

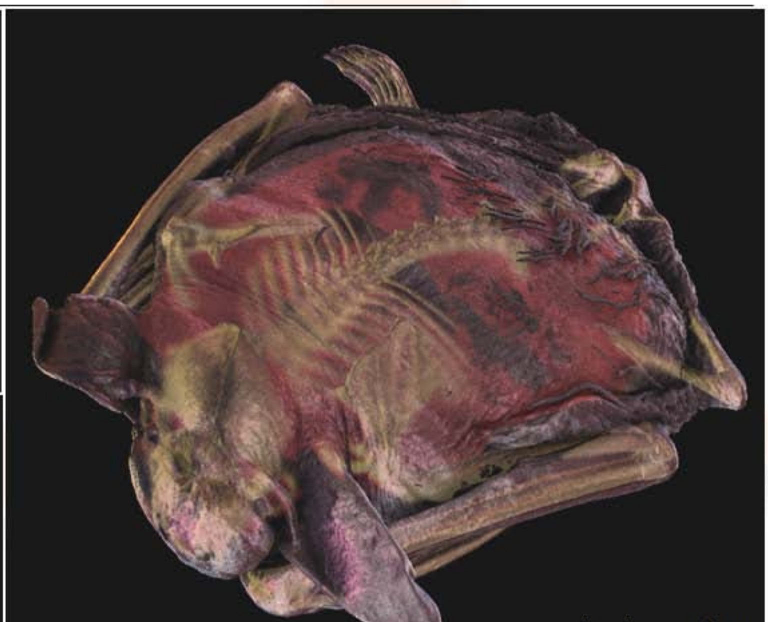
Christof Rezk Salama
Computer Graphics Group
Institute for Vision and Graphics
University of Siegen, Germany

Timo Ropinski
Visualization and Computer Graphics Research Group,
University of Münster, Germany

Advanced Illumination

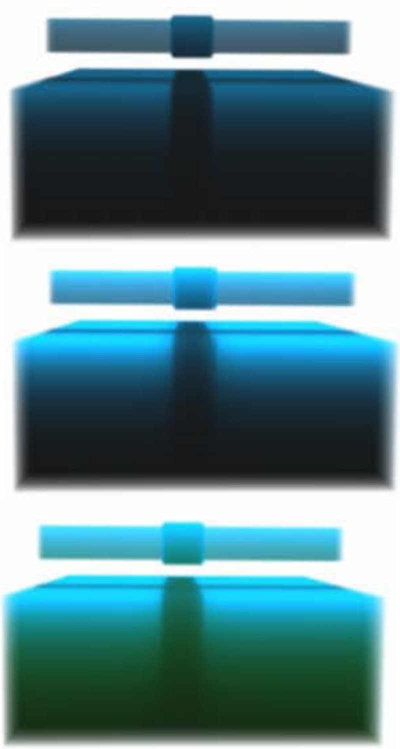

cheetah skull


pterosaur skull


big brown bat

Data sets available at the
UTCT data archive, DIGIMORPH
http://utct.tacc.utexas.edu

Translucency



Light Transport

Wave-Particle Duality

- **Photons**
  - Quantum of light (the smallest possible packet of light at a given wavelength)
  - Photoelectric effect (van Lenard, 1902)
- **Wave Theory** (Maxwell)
  - Electro-magnetic wave characteristics of light
  - Effects such as interference and diffraction
- **Quantum Mechanics** (Einstein)
  - *Universal theory of light transport*
  - *probabilistic* characteristics of the motions of atoms and photons (quantum optics)

Light Transport

Wave-Particle Duality

- **Photons**
  - Quantum of light (the smallest possible packet of light at a given wavelength)
  - Photoelectric effect (van Lenard, 1902)
- **Wave Theory** (Maxwell)
  - Electro-magnetic wave characteristics of light
  - Effects such as interference and diffraction
- **Quantum Mechanics** (Einstein)
  - *Universal theory of light transport*
  - *probabilistic* characteristics of the motions of atoms and photons (quantum optics)

Vis08

Scattering Effects
Single and Multiple Scattering

Markus Hadwiger
VR VIS Research Center
Vienna, Austria

Patric Ljung
Siemens Corporate Research
Princeton, NJ, USA

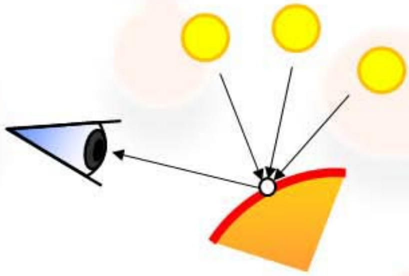
Christof Rezk Salama
Computer Graphics Group
Institute for Vision and Graphics
University of Siegen, Germany

Timo Ropinski
Visualization and Computer Graphics Research Group,
University of Münster, Germany

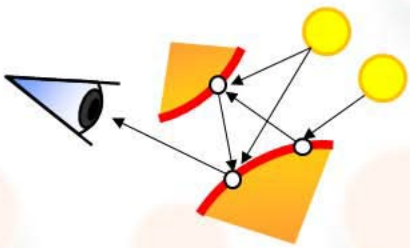
Scattering Effects

When a photon hits a surface, it changes both direction and energy

- *Single Scattering:*
  - Light is scattered *once* before it reaches the eye
  - Local illumination model

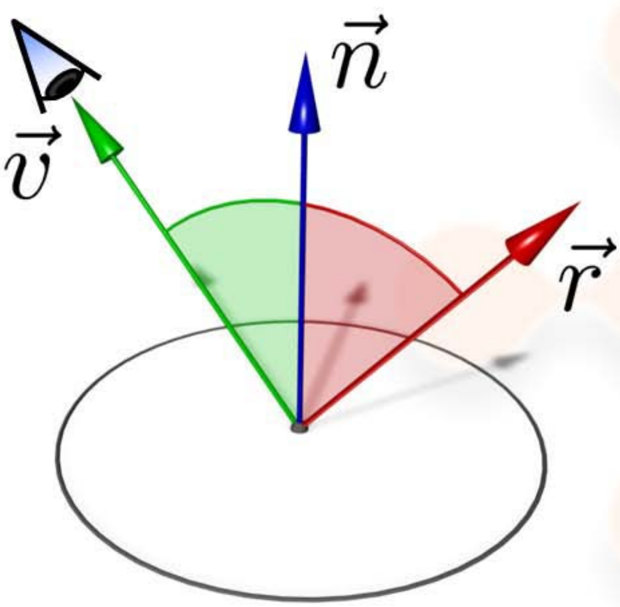


- *Multiple Scattering*
  - Soft shadows
  - Translucency
  - Color bleeding



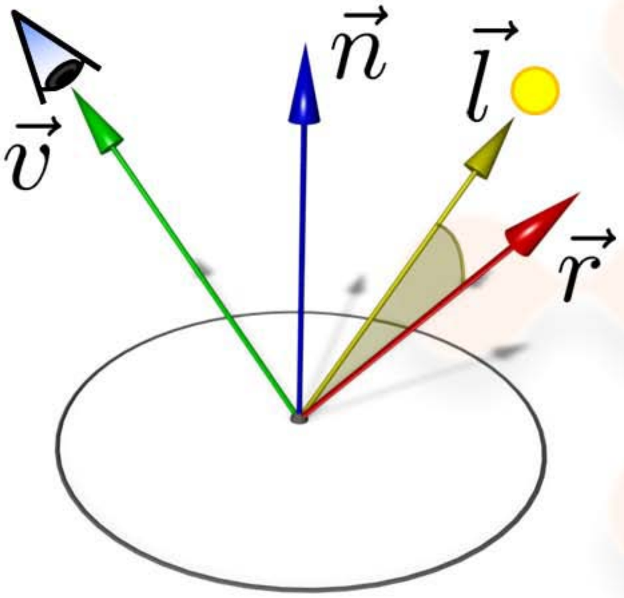
Single Scattering

Phong illumination with point light sources



Single Scattering

Phong illumination with point light sources

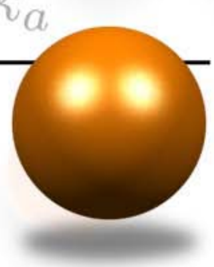


$$I_{\text{Lambert}} = L_d\, k_d\, (\vec{l} \cdot \vec{n})$$

$$I_{\text{Specular}} = L_s\, k_s\, (\vec{l} \cdot \vec{r})^s$$

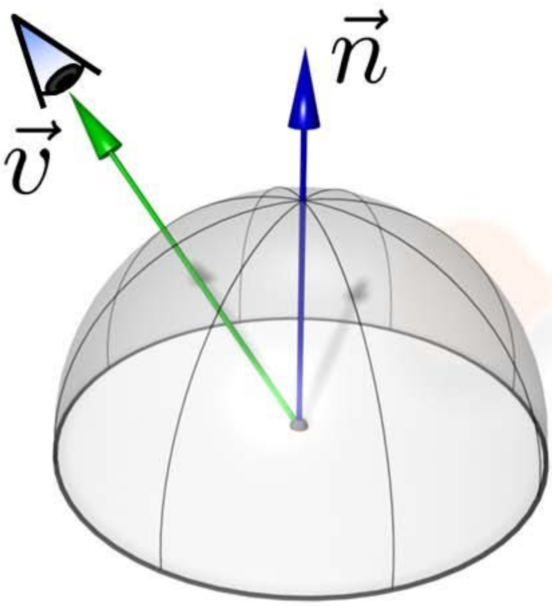
$$I_{\text{Ambient}} = L_a\, k_a$$

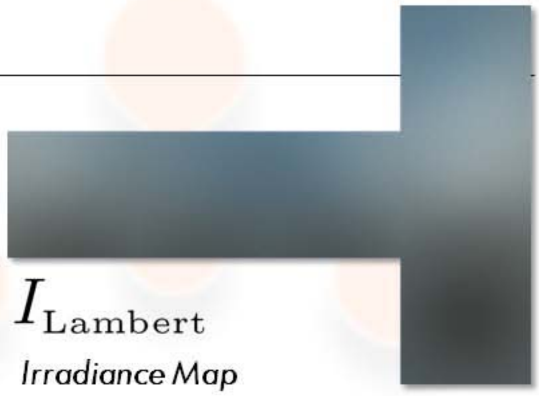
$$I_{\text{Phong}}$$



Single Scattering

Environment Light



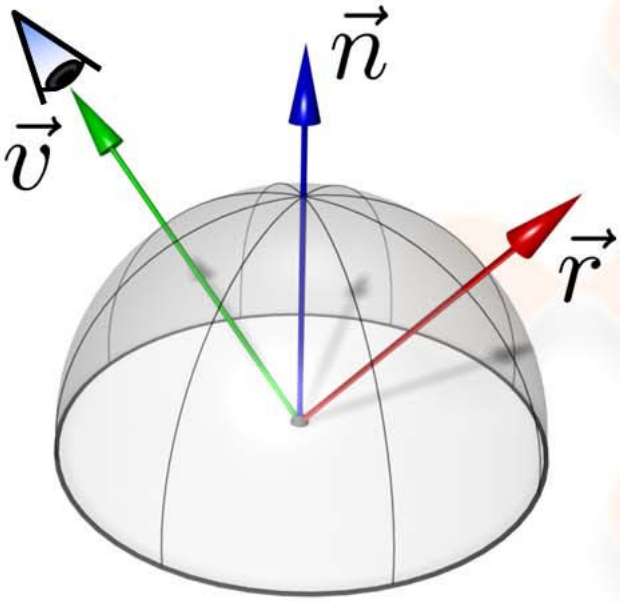


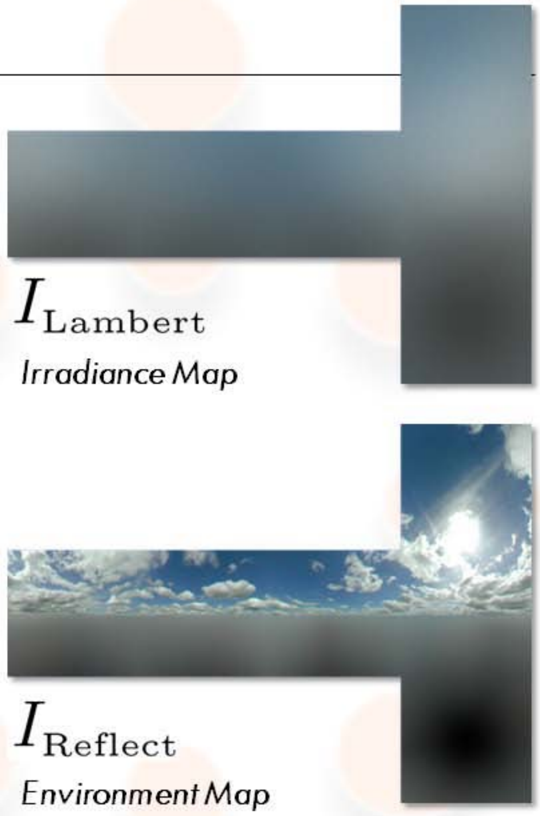
I_{Lambert}

Irradiance Map

Single Scattering

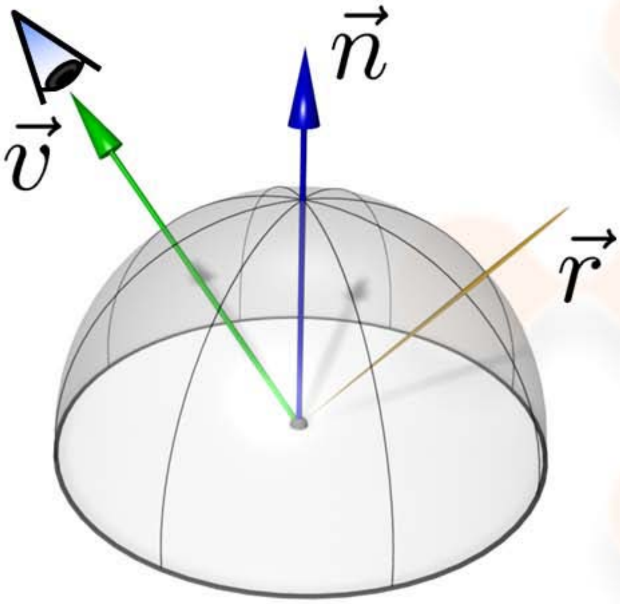
Environment Light

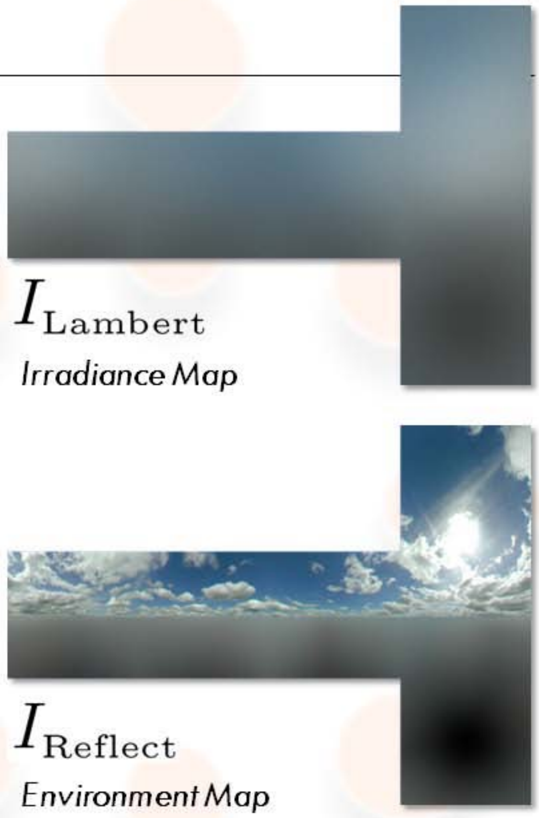




Single Scattering

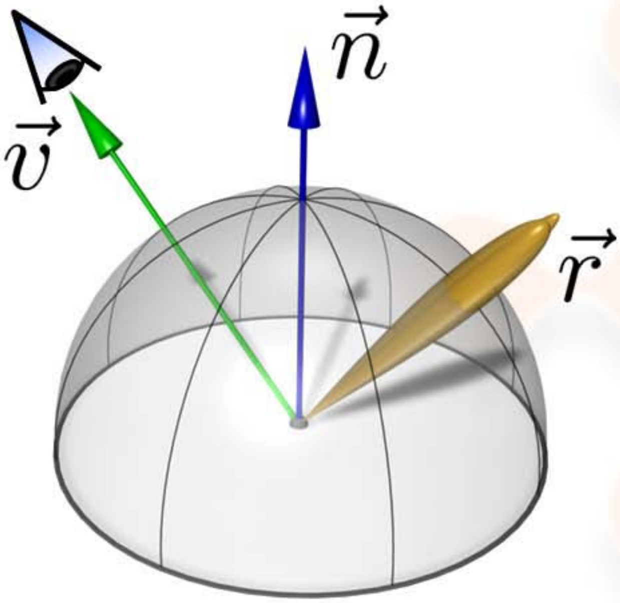
Environment Light

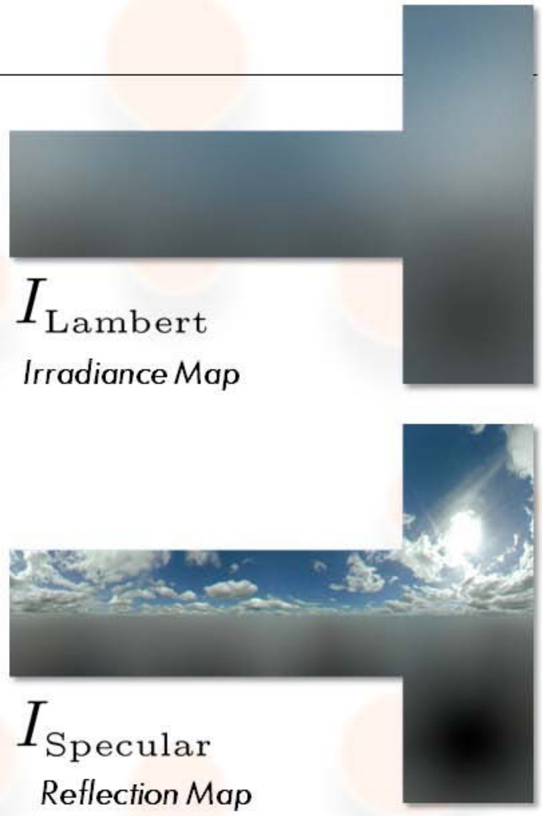




Single Scattering

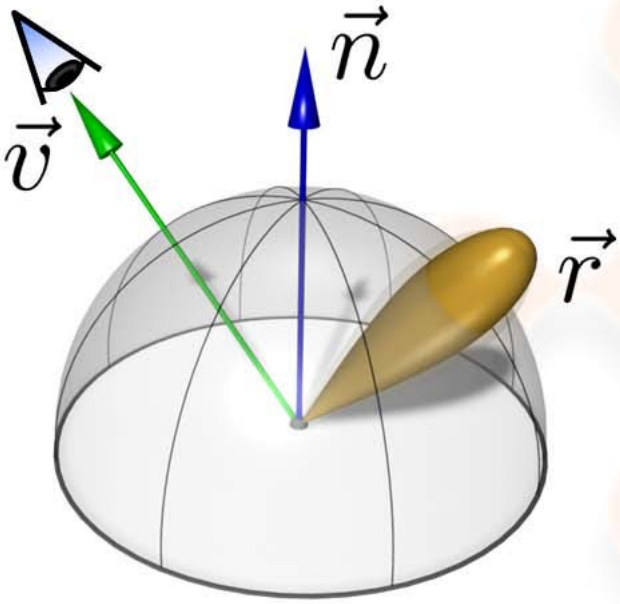
Environment Light

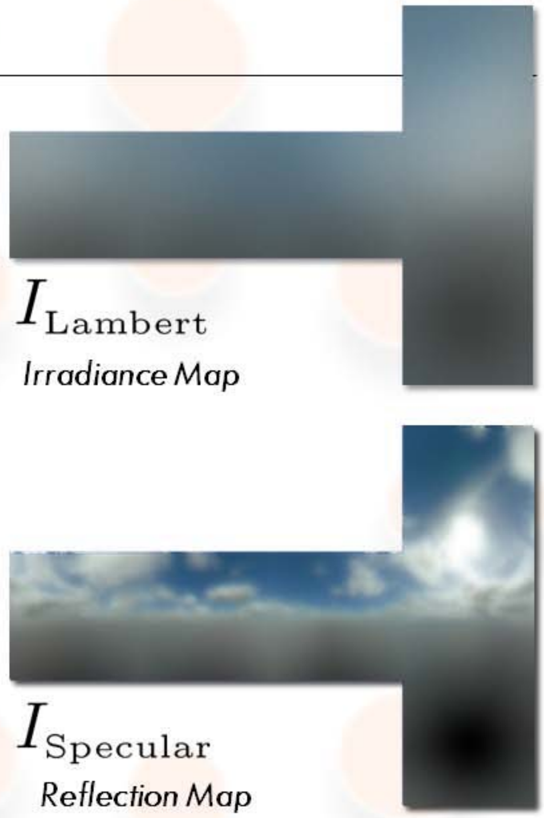




Single Scattering

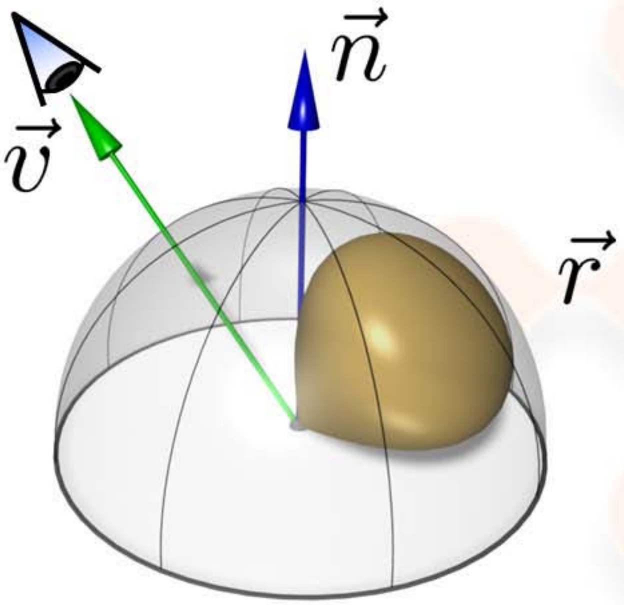
Environment Light

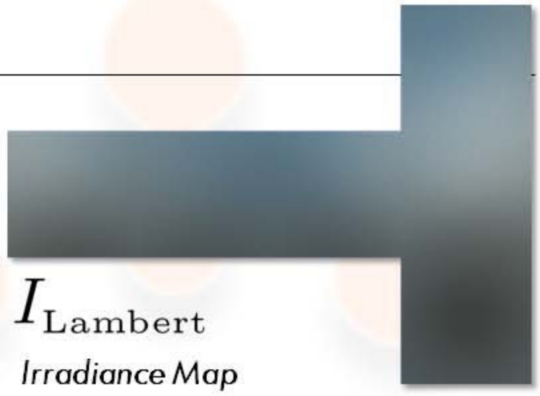




Single Scattering

Environment Light





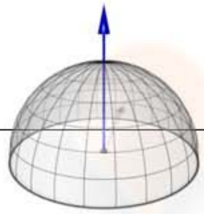
I_{Lambert}

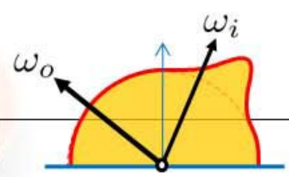
Irradiance Map

I_{Specular}

Reflection Map

Math Notation

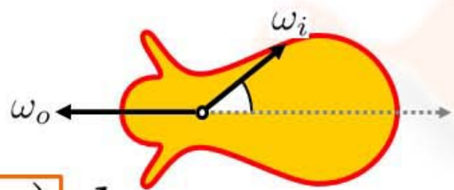




- **Surface Illumination**

$$L(\mathbf{x}, \omega_o) = \int_{\Omega+} f(\mathbf{x}, \omega_o \rightarrow \omega_i) \cos\theta_i \, d\omega_i$$

Hemisphere | BRDF | Elevation Angle



- **Volume Illumination**

$$L(\mathbf{x}, \omega_o) = \int_{\Omega} p(\mathbf{x}, \omega_o \rightarrow \omega_i) \, d\omega_i$$

Sphere | Phase Function

Scattering Effects
Monte-Carlo Methods

Markus Hadwiger
VR VIS Research Center
Vienna, Austria

Patric Ljung
Siemens Corporate Research
Princeton, NJ, USA

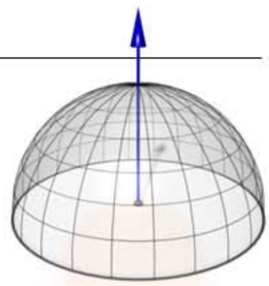
Christof Rezk Salama
Computer Graphics Group
Institute for Vision and Graphics
University of Siegen, Germany

Timo Ropinski
Visualization and Computer
Graphics Research Group,
University of Münster, Germany

Math Notation

Mathematical Model

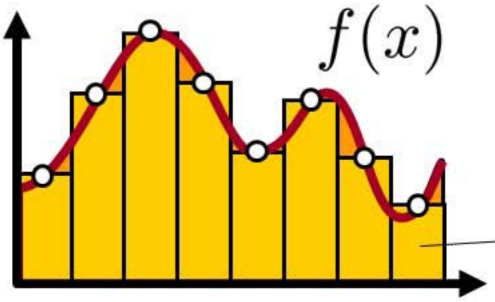
$$L(\mathbf{x}, \omega_o) = \int_\Omega p(\mathbf{x}, \omega_o \rightarrow \omega_i)\, L(\mathbf{x}, \omega_i)\, d\omega_i$$



integrates over the entire sphere/hemisphere

- Integral must be solved for every intersection point
- *Fredholm Equation* (cannot be solved analytically)

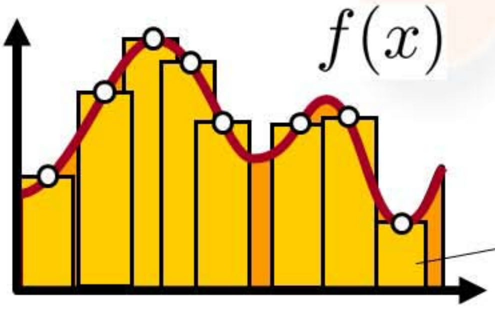
Numerical Integration



Equidistant Sampling

- Approximation integral by a Riemann sum

$$\int_a^b f(x)dx \approx \sum_{i=0}^{N} f(x_i)\frac{b-a}{N}$$



Stochastic Sampling

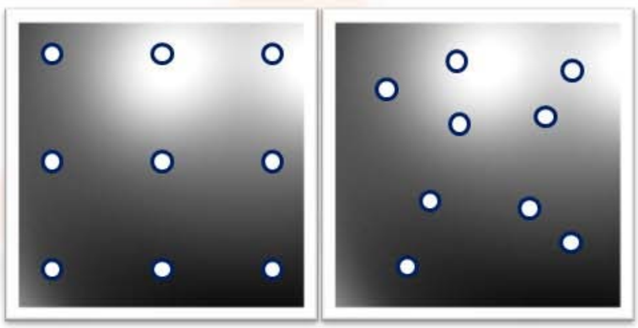
- Uniformly distributed samples
- Approximation by sum

$$\int_a^b f(x)dx \approx \sum_{i=0}^{N} f(x_i)\frac{b-a}{N}$$

Stochastic Sampling

Cons:

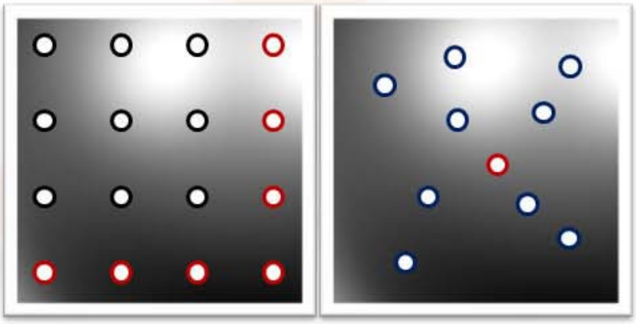
- Slower convergence than Riemann sum



Pros:

- *Better Scalability for multidimensional functions:* increase number of samples in arbitrary steps

Stochastic Sampling



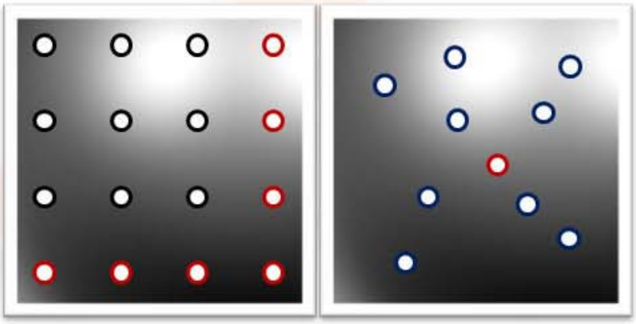
Cons:

- Slower convergence than Riemann sum

Pros:

- *Better Scalability for multidimensional functions:* increase number of samples in arbitrary steps

Stochastic Sampling

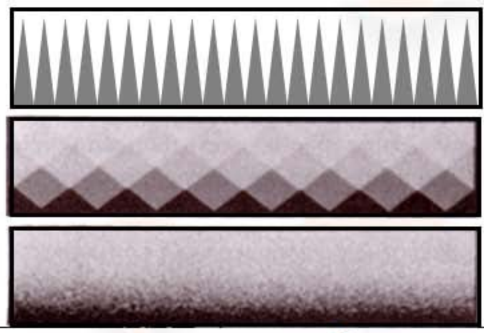


Cons:

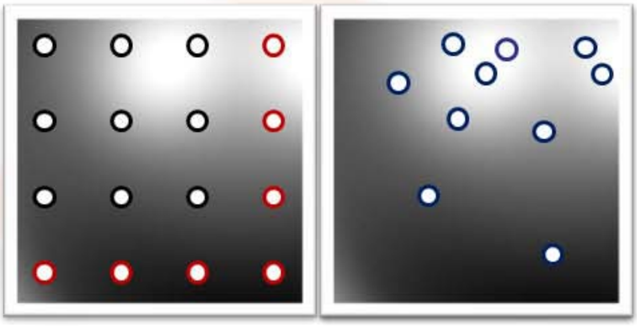
- Slower convergence than Riemann sum

Pros:

- *Better Scalability for multidimensional functions:* increase number of samples in arbitrary steps
- *Noise* instead of Aliasing



Stochastic Sampling

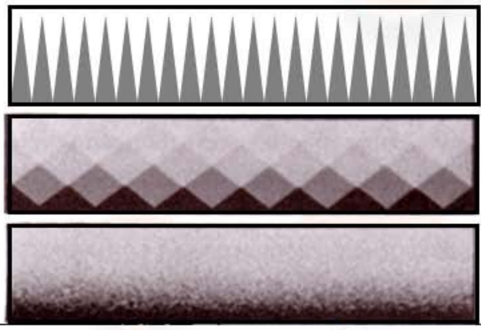


Cons:

- Slower convergence than Riemann sum

Pros:

- *Better Scalability for multidimensional functions:* increase number of samples in arbitrary steps
- *Noise* instead of Aliasing
- *Independent of sampling grid:* Clever placement of samples will improve the convergence!



Blind Monte-Carlo Sampling

- **Example: Filtering an Environment Map**

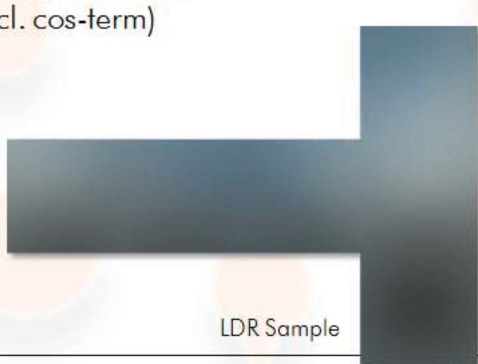
Given an Environment Map

(i.e. photograph: fisheye or mirror ball)

Calculate an Irradiance Map

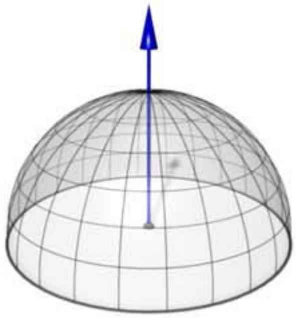
For each pixel of the irradiance map:

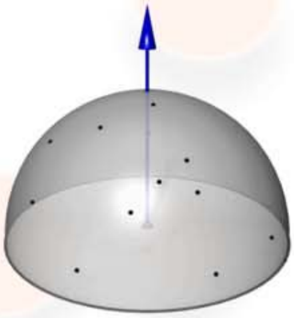
- Determine n random directions on the hemisphere
- Sample the Environment Map and
- Average the results (incl. cos-term)



Rendering

- Calculate the radiance from a point
  - depending on the incoming light on the sphere/hemisphere

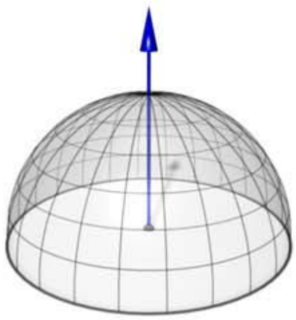


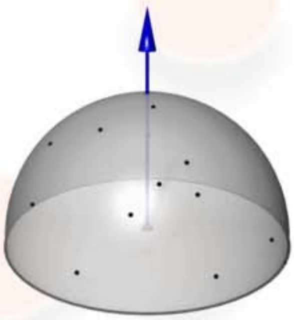


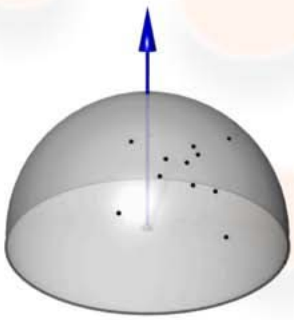
Deterministic	*Blind Monte-Carlo*
Uniform sampling of the sphere/hemisphere.	Randomized sampling of the sphere/hemisphere.
High computational load good approximation	Visually better images for fewer samples, slow convergence

Rendering

- Calculate the radiance from a point
  - depending on the incoming light on the sphere/hemisphere
  - depending on the phase function/BRDF

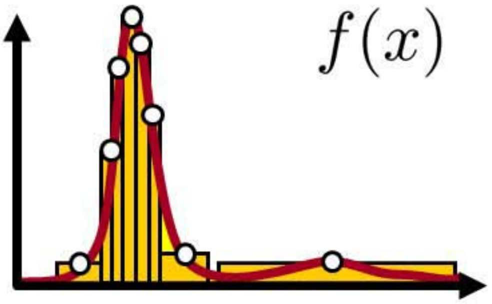






Deterministic	*Blind Monte-Carlo*	*Importance Sampling*
Uniform sampling of the sphere/hemisphere.	Randomized sampling of the sphere/hemisphere.	*Place samples where* contribution is high
High computational load good approximation	Visually better images for fewer samples, slow convergence	Faster!

Importance Sampling



Stochastic Sampling

- Non-uniformly distributed samples
- Approximation by sum

$$\int_a^b f(x)dx \approx \sum_{i=0}^{N} \frac{f(x_i)}{p(x_i)}$$

Probability Distribution Function (PDF)

Clever placement of samples

- Many samples where function is high
- Few samples where function is low

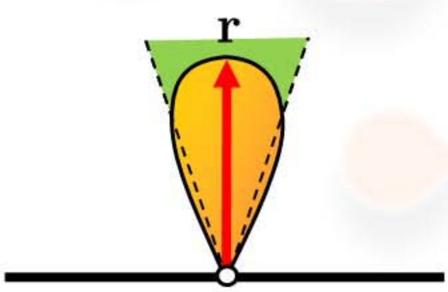
Sampling a Specular Lobe

- **Simple Approach**

  Specular term $f(\varphi) = \cos^s(\varphi) = (\mathbf{r} \cdot \mathbf{v})^s$

  Non-optimal, but easy to implement

  Idea: uniform distribution of directions restricted to a cone



Sampling a Specular Lobe

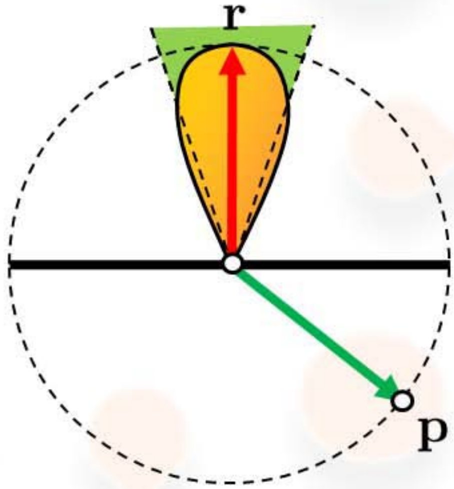
- **Simple Approach**

  Specular term $f(\varphi) = \cos^s(\varphi) = (\mathbf{r} \cdot \mathbf{v})^s$

  Non-optimal, but easy to implement

  Idea: uniform distribution of directions restricted to a cone

  - Precompute random unit vectors with uniform PDF
  - Randomly pick one vector $\mathbf{p}$



Sampling a Specular Lobe

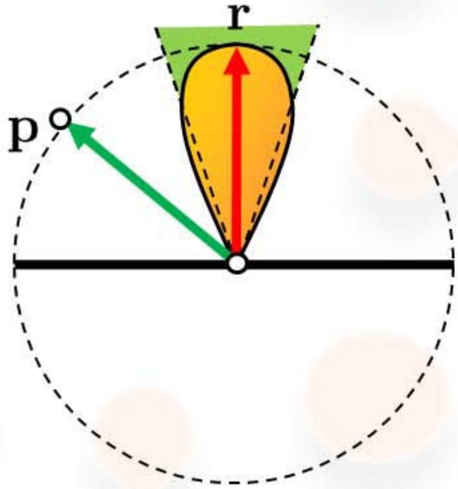
- **Simple Approach**

  Specular term $f(\varphi) = \cos^s(\varphi) = (\mathbf{r} \cdot \mathbf{v})^s$

  Non-optimal, but easy to implement

  Idea: uniform distribution of directions restricted to a cone

  - Precompute random unit vectors with uniform PDF
  - Randomly pick one vector $\mathbf{p}$
  - Negate vector, if $(\mathbf{r} \cdot \mathbf{p}) < 0$



Sampling a Specular Lobe

- **Simple Approach**

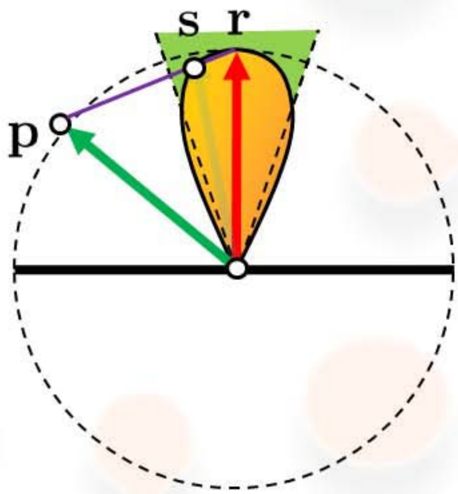
  Specular term $f(\varphi) = \cos^s(\varphi) = (\mathbf{r} \cdot \mathbf{v})^s$

  Non-optimal, but easy to implement

  Idea: uniform distribution of directions restricted to a cone

  - Precompute random unit vectors with uniform PDF
  - Randomly pick one vector $\mathbf{p}$
  - Negate vector, if $(\mathbf{r} \cdot \mathbf{p}) < 0$
  - Blend with vector $\mathbf{r}$ and normalize

    $$\mathbf{s} = \alpha\,\mathbf{r} + (1-\alpha)\mathbf{p}$$



Sampling a Specular Lobe

- **Simple Approach**

  Specular term $f(\varphi) = \cos^s(\varphi) = (\mathbf{r} \cdot \mathbf{v})^s$

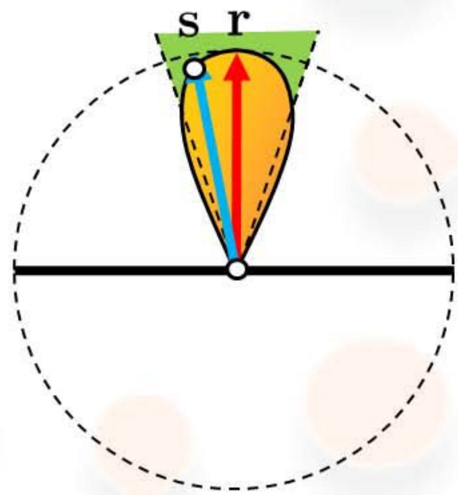
  Non-optimal, but easy to implement

  Idea: uniform distribution of directions restricted to a cone

  - Precompute random unit vectors with uniform PDF
  - Randomly pick one vector $\mathbf{p}$
  - Negate vector, if $(\mathbf{r} \cdot \mathbf{p}) < 0$
  - Blend with vector $\mathbf{r}$ and normalize

    $$\mathbf{s} = \alpha\,\mathbf{r} + (1-\alpha)\mathbf{p}$$

  - Blend weight α controls the size of the specular highlight and can be calculated from shininess s



CHRISTOF REZK-SALAMA, COMPUTER GRAPHICS GROUP, UNIVERSITY OF SIEGEN, GERMANY
ADVANCED ILLUMINATION TECHNIQUES FOR GPU-BASED VOLUME RAYCASTING

Stochastic Sampling

$$\int_a^b f(x)dx \approx \sum_{i=0}^{N} \frac{f(x_i)}{p(x_i)}$$

- What is the *ideal PDF* for sampling a given function $f(x)$?
- Variance is minimal, if

$$p(x) = \lambda \cdot f(x)$$

- λ must be chosen to normalize the distribution
- Problem:

$$\int_a^b p(x)dx = 1 \quad \Rightarrow \; p(x) = \frac{f(x)}{\int_a^b f(x)dx}$$

- The ideal PDF requires knowing the integral beforehand!

Stochastic Sampling

unknown

$$L(\mathbf{x}, \omega_o) = \int_\Omega f(\mathbf{x}, \omega_i \rightarrow \omega_o)\, L(\mathbf{x}, \omega_i) \cos\theta_i \, d\omega_i$$

Although we do not know the integral completely,
we still know parts of it

Stochastic Sampling

unknown

$$L(\mathbf{x}, \omega_o) = \int_{\Omega} f(\mathbf{x}, \omega_i \to \omega_o) \, L(\mathbf{x}, \omega_i) \cos\theta_i \, d\omega_i$$

known

Although we do not know the integral completely, we still know parts of it

$$\hat{p}(\omega_i) = f(\mathbf{x}, \omega_i \to \omega_o) \, \cos\theta_i$$

$$p(\omega_i) = \frac{\hat{p}(\omega_i)}{\int_{\Omega} \hat{p}(\omega) d\omega}$$

Solid Angle

$$A_{\text{Hemisphere}} = \int_{\Omega+} 1 \, d\omega \; = 2\,\pi$$

$$du = r \, d\theta$$

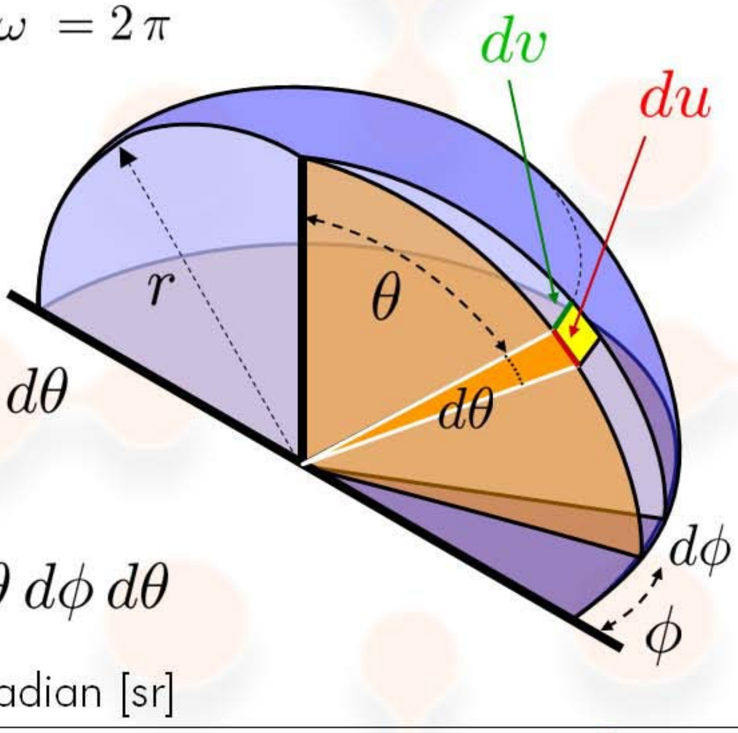
$$dv = r \; \sin\theta \, d\phi$$

Area (yellow):

$$dA = r^2 \; \sin\theta \, d\phi \, d\theta$$

Solid Angle:

$$d\omega = \frac{dA}{r^2} = \sin\theta \, d\phi \, d\theta$$



Unit of solid angle: Steradian [sr]

Sampling a Specular Lobe

- **Ideal Sampling**

$$f(\omega_i) = \cos^n(\theta_i) \qquad p(\omega_i) = \frac{f(\omega_i)}{\int_{\Omega+} f(\omega_i) d\omega}$$

$$\int_{\Omega+} \cos^n(\theta) d\omega = \int_0^{2\pi} \int_0^{\pi/2} \cos^n(\theta) \sin(\theta)\, d\theta\, d\phi = \frac{2\pi}{(n+1)}$$

Sampling a Specular Lobe

- **Ideal Sampling**

$$f(\omega_i) = \cos^n(\theta_i) \qquad p(\omega_i) = \frac{f(\omega_i)}{\int_{\Omega+} f(\omega_i) d\omega}$$

$$p(\theta_i, \phi_i) = \frac{(n+1)}{2\pi} \cos^n \theta_i \, \sin \theta_i$$

Sampling a Specular Lobe

- **Ideal Sampling**

$$f(\omega_i) = \cos^n(\theta_i) \qquad p(\omega_i) = \frac{f(\omega_i)}{\int_{\Omega+} f(\omega_i)d\omega}$$

$$p(\theta_i, \phi_i) = \frac{(n+1)}{2\pi} \cos^n \theta_i \sin \theta_i$$

$$\swarrow \qquad \searrow$$

$$p(\theta_i) = (n+1)\cos^n \theta_i \sin \theta_i \qquad p(\phi_i | \theta_i) = \frac{1}{2\pi}$$

- Convert to CDF and invert

$$\theta_i = \cos^{-1} \xi_1^{\left(\frac{1}{n+1}\right)} \qquad \phi_i = 2\pi \xi_2$$

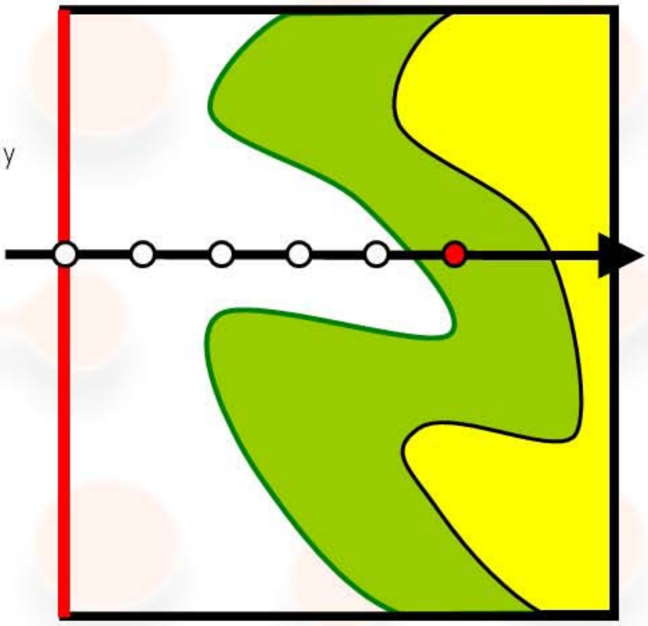
Importance Sampling

Literature:

- M. Pharr, G. Humphries: **Physically Based Rendering**, Morgan Kauffman (Elsevier), 2004
- M. Colbert, J. Křivánek, **GPU-Based Importance Sampling** *in H.Nguyen (edt.):* **GPU Gems 3**, *Addison-Wesley, 2008*

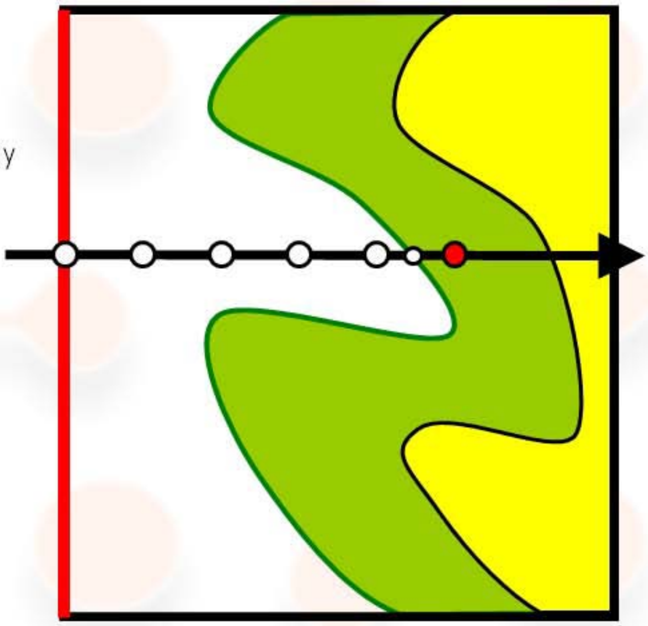
GPU Ray-Casting

- Calculate First Intersection with Isosurface
  - Rasterize the front faces of the bounding box
  - For each fragment, cast a ray
  - Find first intersaction point with isosurface by sampling along the ray



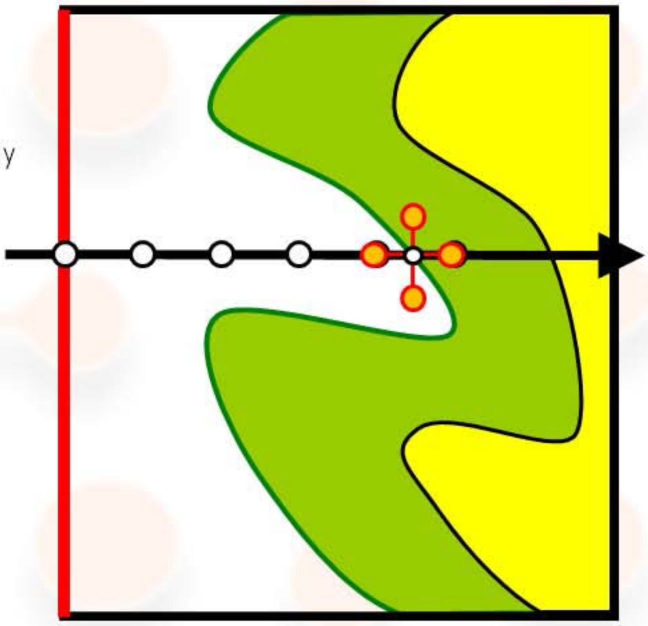
GPU Ray-Casting

- Calculate First Intersection with Isosurface
  - Rasterize the front faces of the bounding box
  - For each fragment, cast a ray
  - Find first intersaction point with isosurface by sampling along the ray
    - interval bisection

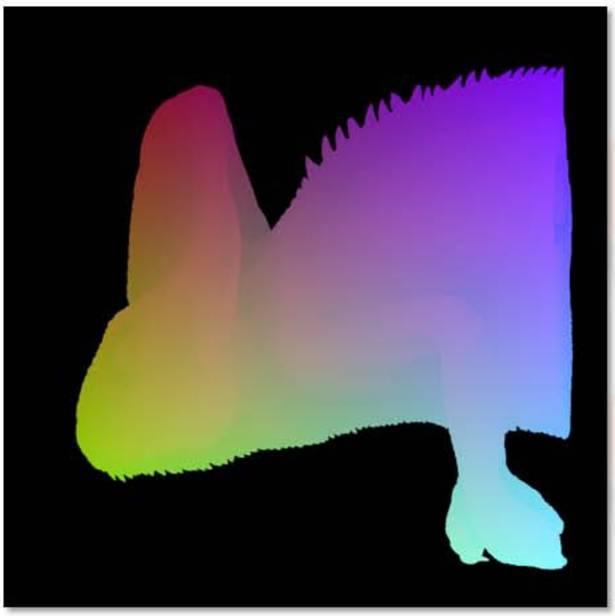


GPU Ray-Casting

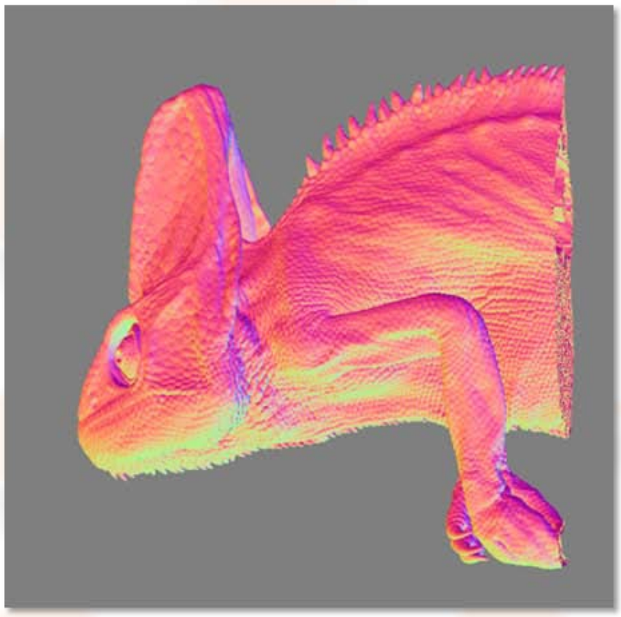
- Calculate First Intersection with Isosurface
  - Rasterize the front faces of the bounding box
  - For each fragment, cast a ray
  - Find first intersaction point with isosurface by sampling along the ray
    - interval bisection
  - Store the intersection point in render target 0
  - Estimate the gradient vector using central differences
  - Store the gradient vector in render target 1



First Render Pass



MRT0: xyz-coordinates of first intersection point with isosurface

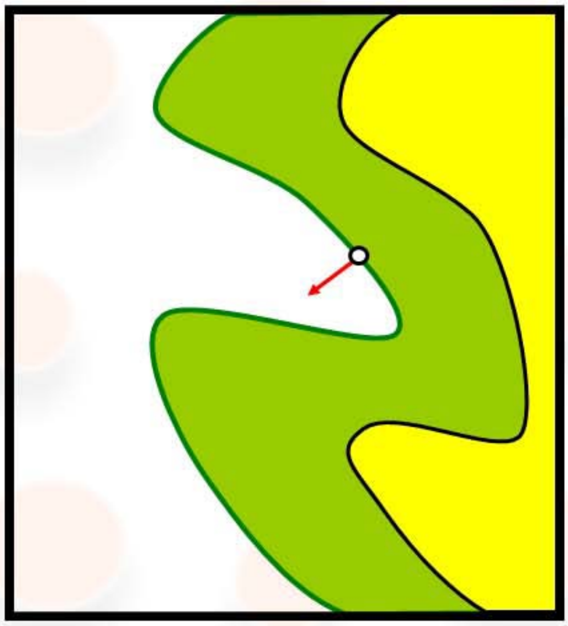


MRT1: xyz-components of gradient vector (color coded)

Deferred Shading

Single Scattering (no shadows)

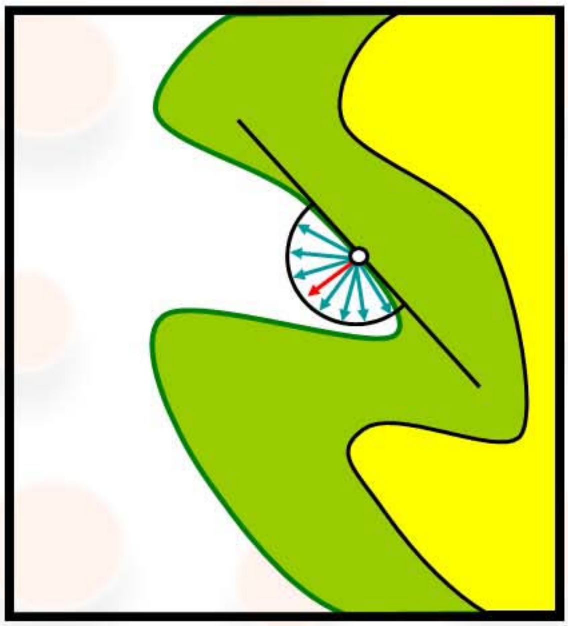
- Diffuse term:
  - Sample irradiance cube using gradient direction



Deferred Shading

Single Scattering (no shadows)

- Diffuse term:
  - Sample irradiance cube using gradient direction
- Specular term:
  - Calculate random directions on the specular lobe
  - Sample environment cube

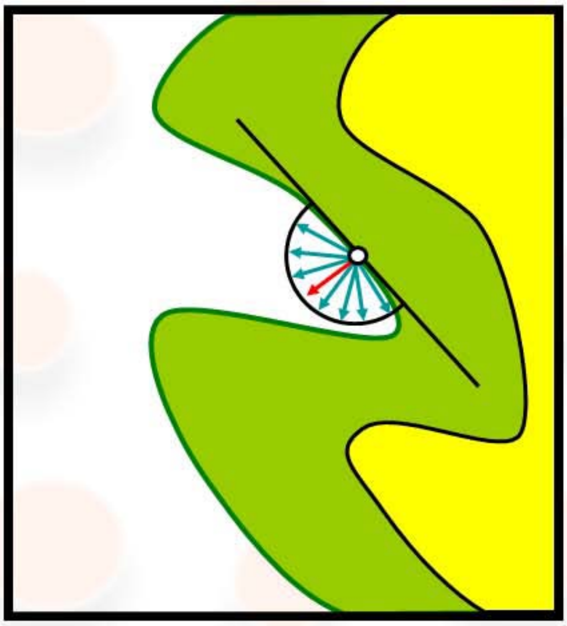


CHRISTOF REZK-SALAMA, COMPUT CS GROUP, UNIVERSITY OF SIEGEN, GERMANY
ADVANCED ILLUMINATION TECHNIC PU-BASED VOLUME RAYCASTING

Deferred Shading

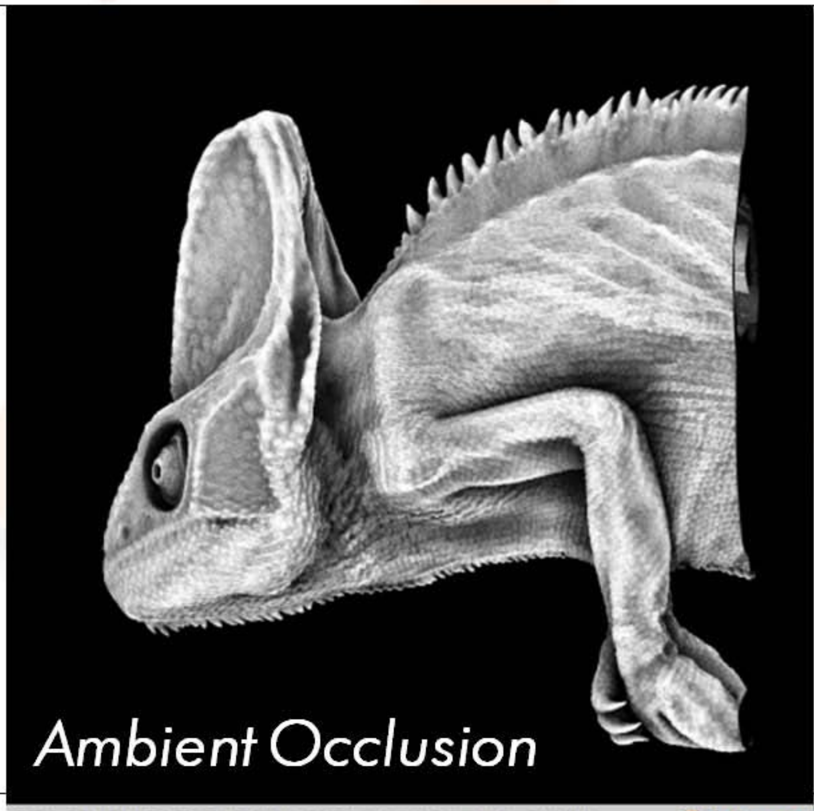
Single Scattering (no shadows)

- Diffuse term:
  - Sample irradiance cube using gradient direction
- Specular term:
  - Calculate random directions on the specular lobe
  - Sample environment cube
  - Weight each sample with its BRDF/phase function and its probability distribution

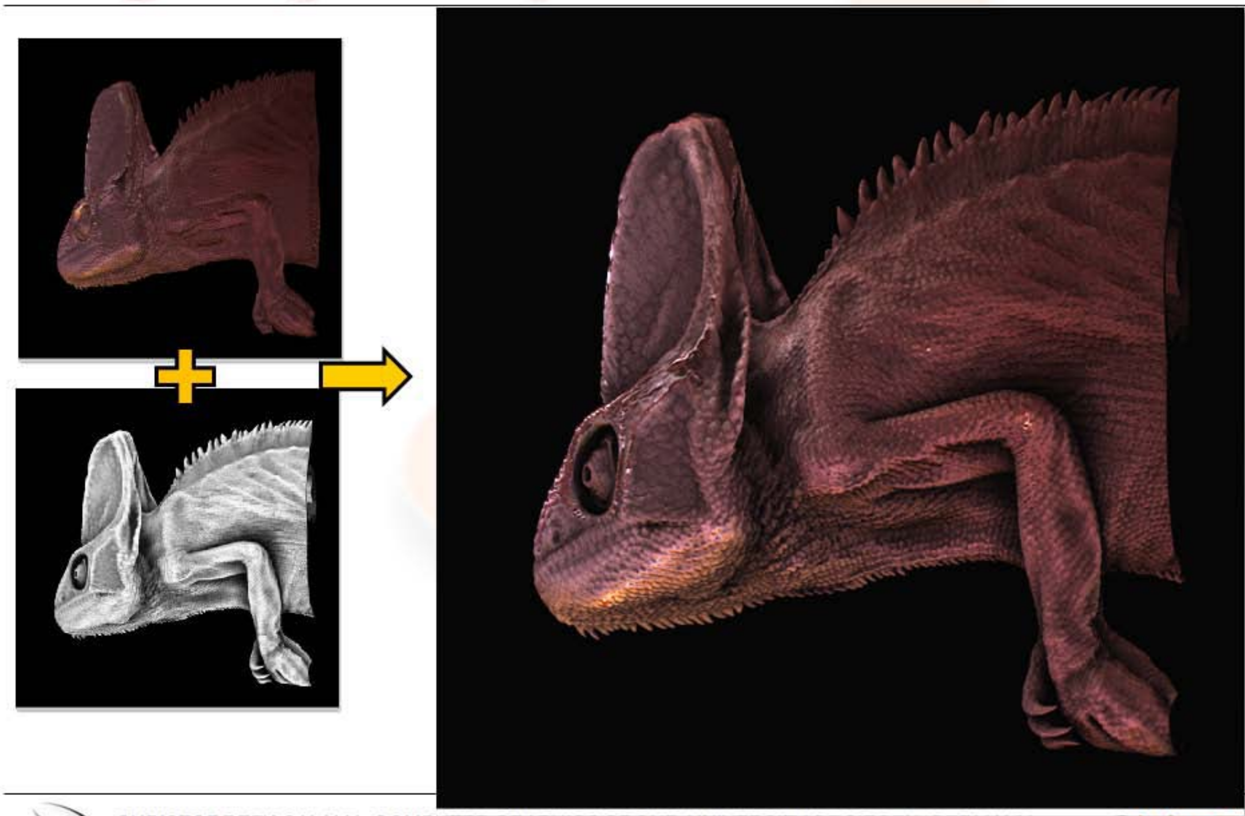


High Quality Isosurface

High Quality Isosurface



High Quality Isosurface



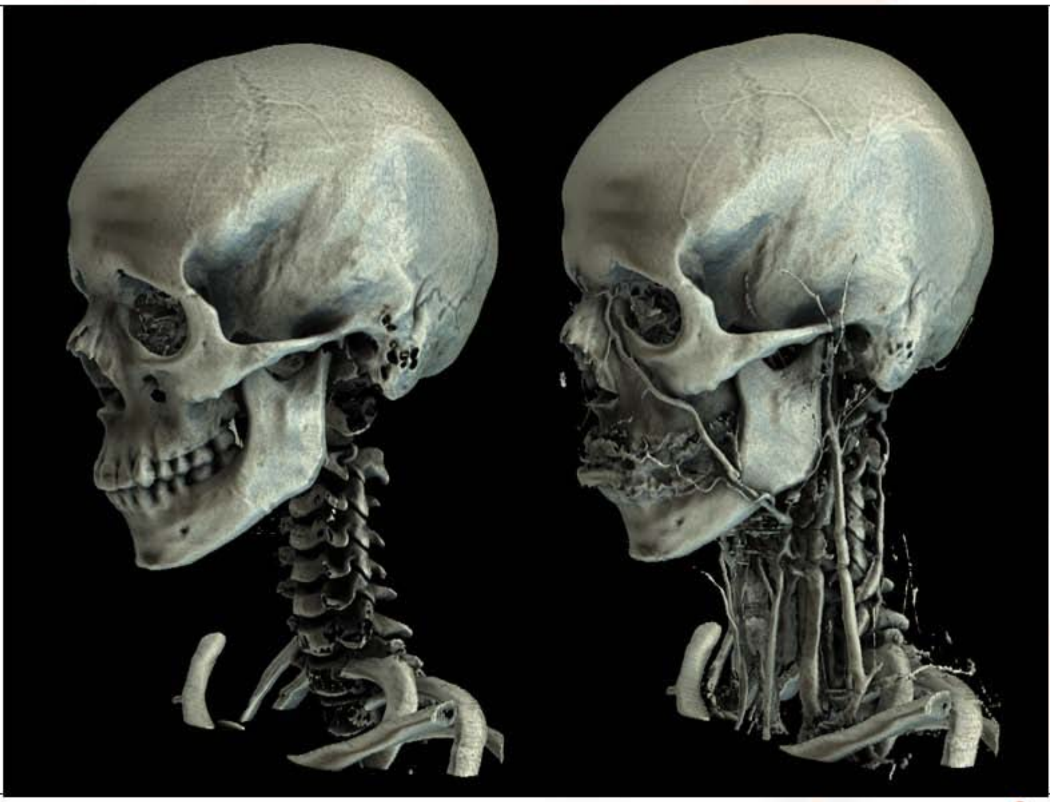
Our First Implementation

Why not use a pre-filtered environment map?

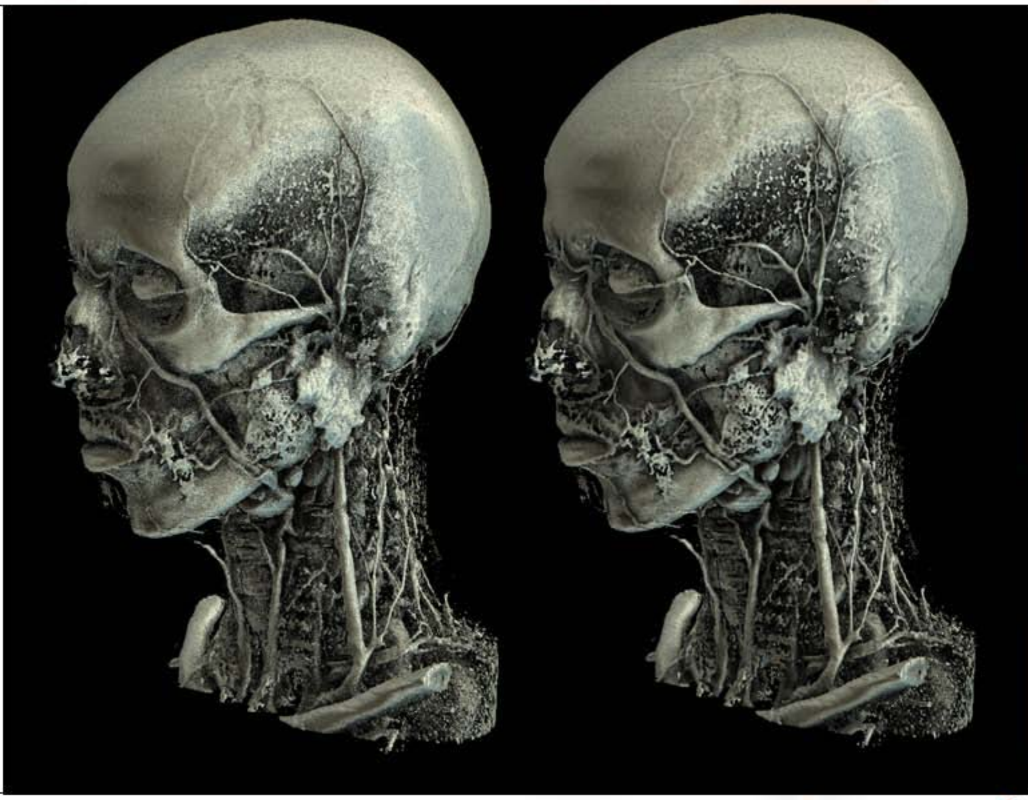
You can, but

- it only works for ***one*** specular exponent per object
- Variable shininess may be used to ***visualize additional surface properties*** (e.g. gradient magnitude)

Single Scattering Example

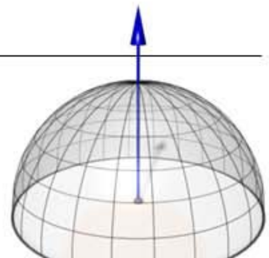


Single Scattering Example



Multiple Scattering

Mathematical Model



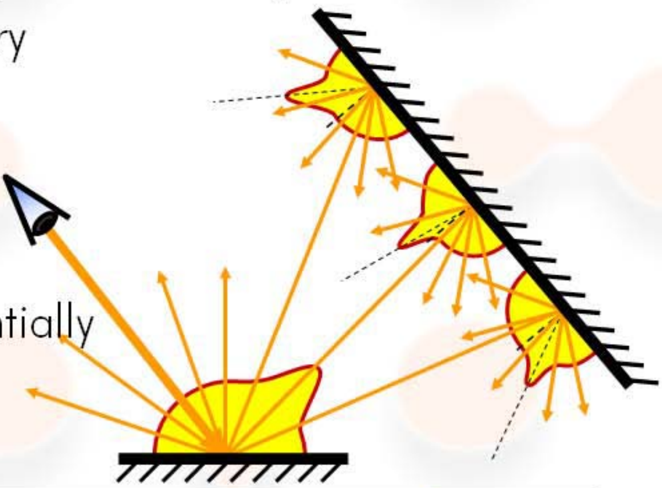
$$L(\mathbf{x}, \omega_o) = \int_\Omega p(\mathbf{x}, \omega_o \rightarrow \omega_i)\, L(\mathbf{x}, \omega_i)\, d\omega_i$$

integrates over the entire sphere/hemisphere

- Integral must be solved for every intersection point
- *Fredholm Equation* (cannot be solved analytically)

Numerical Solution:

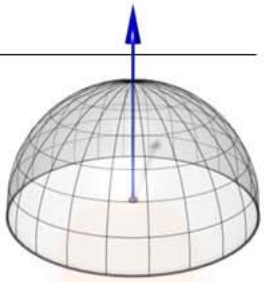
- Number of rays grows exponentially
- Much workload spent for little contribution



Multiple Scattering

Mathematical Model

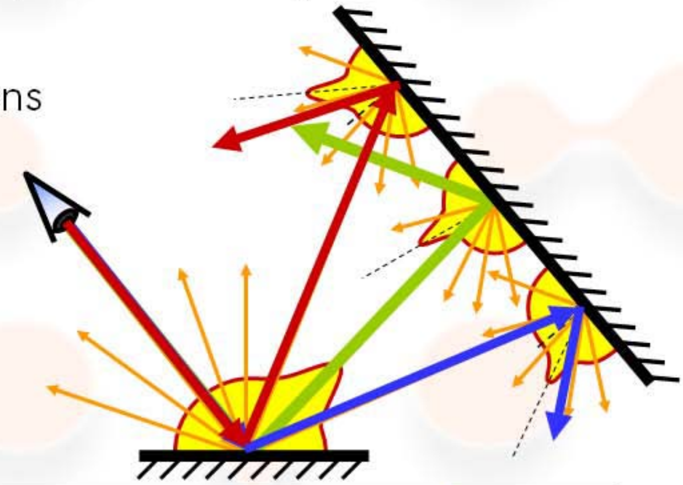
$$L(\mathbf{x}, \omega_o) = \int_{\Omega} p(\mathbf{x}, \omega_o \rightarrow \omega_i)\, L(\mathbf{x}, \omega_i)\, d\omega_i$$



integrates over the entire sphere/hemisphere

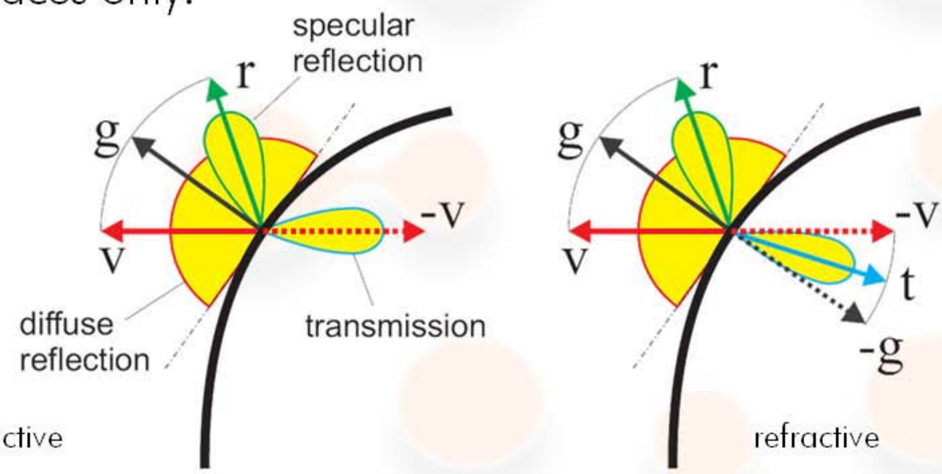
Quantum Optics

- Trace the path of single photons
- Photons are scattered randomly
- Probability of scattering direction given by BRDF/phase function
- *Monte Carlo path tracing*



Phase Function Model

- Scattering of light at every point inside the volume
  - Too expensive (extremely slow convergence)
  - Not practicable. Controlling the visual appearance is difficult
- *Idea:* Restrict scattering events to a fixed number of isosurfaces only.



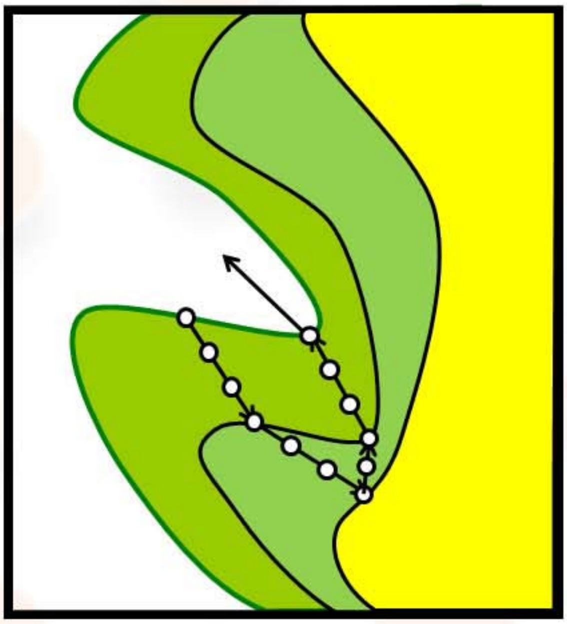
non-refractive

refractive

GPU Ray-Casting

Scattering Pass

- Start at first isosurface and trace inwards
- Account for absorption along the rays
- Proceed until next isosurface
- Calculate scattering event
- Sample the environment on exit



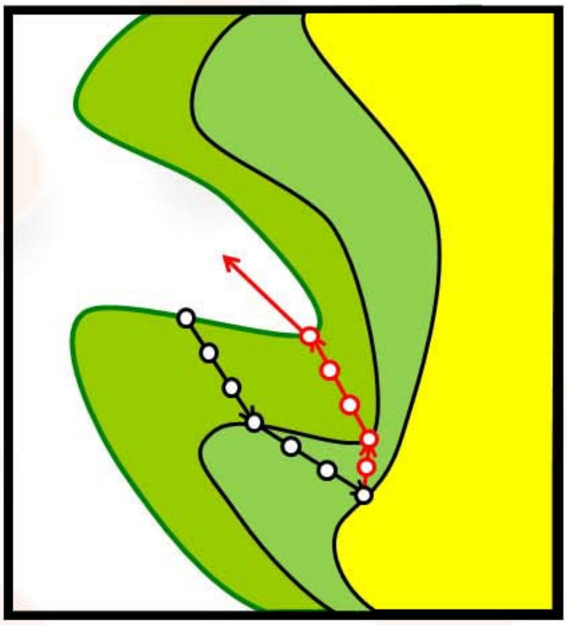
GPU Ray-Casting

Scattering Pass

Simplifying Assumption:

- Absorption on the „way in“ is same as on the „way out“
- Abort the ray inside the volume square the absorption and sample irradiance map

- *Not very accurate but good visual results*

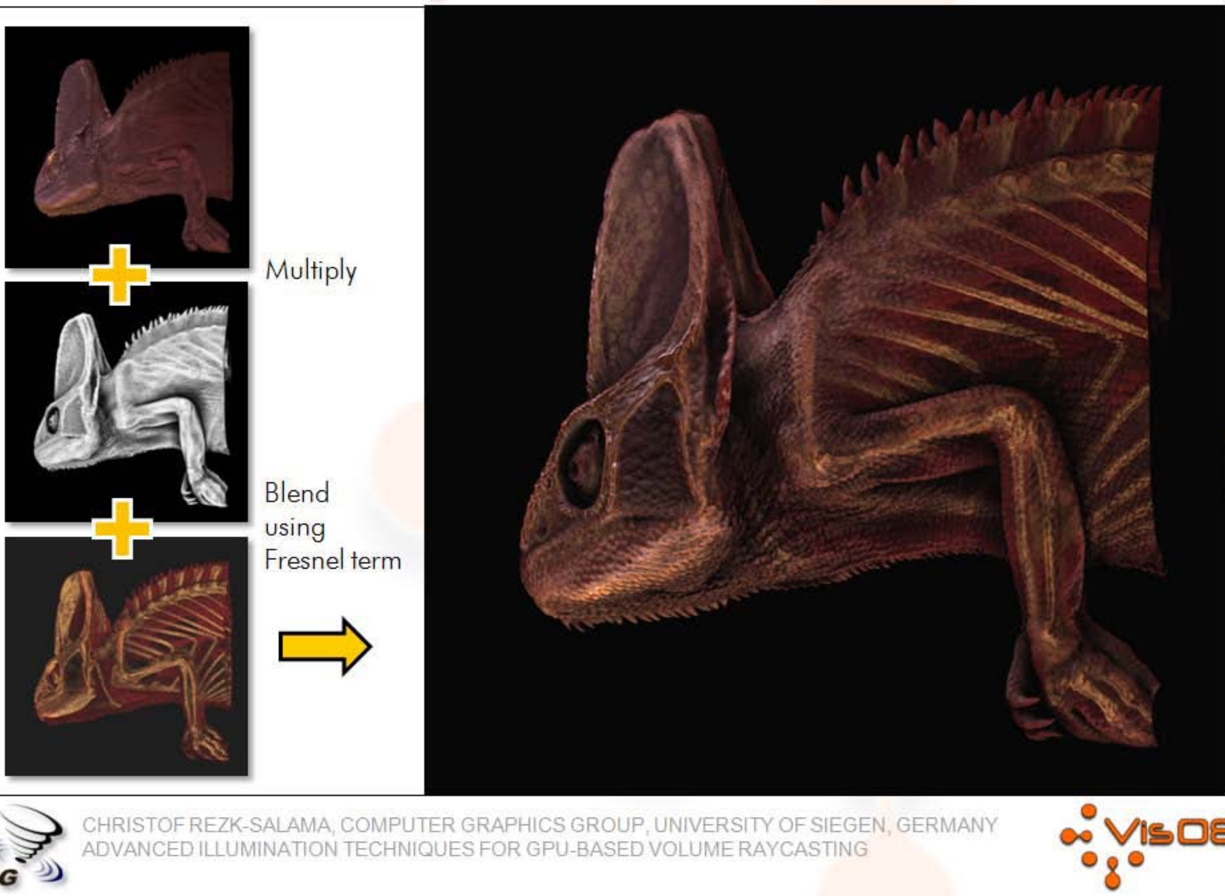


Scattering Pass

preview in real-time

final version in ½-1 seconds

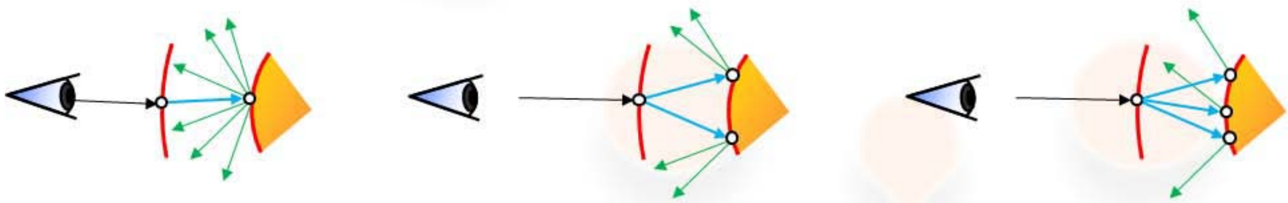
Final Composite



CHRISTOF REZK-SALAMA, COMPUTER GRAPHICS GROUP, UNIVERSITY OF SIEGEN, GERMANY
ADVANCED ILLUMINATION TECHNIQUES FOR GPU-BASED VOLUME RAYCASTING

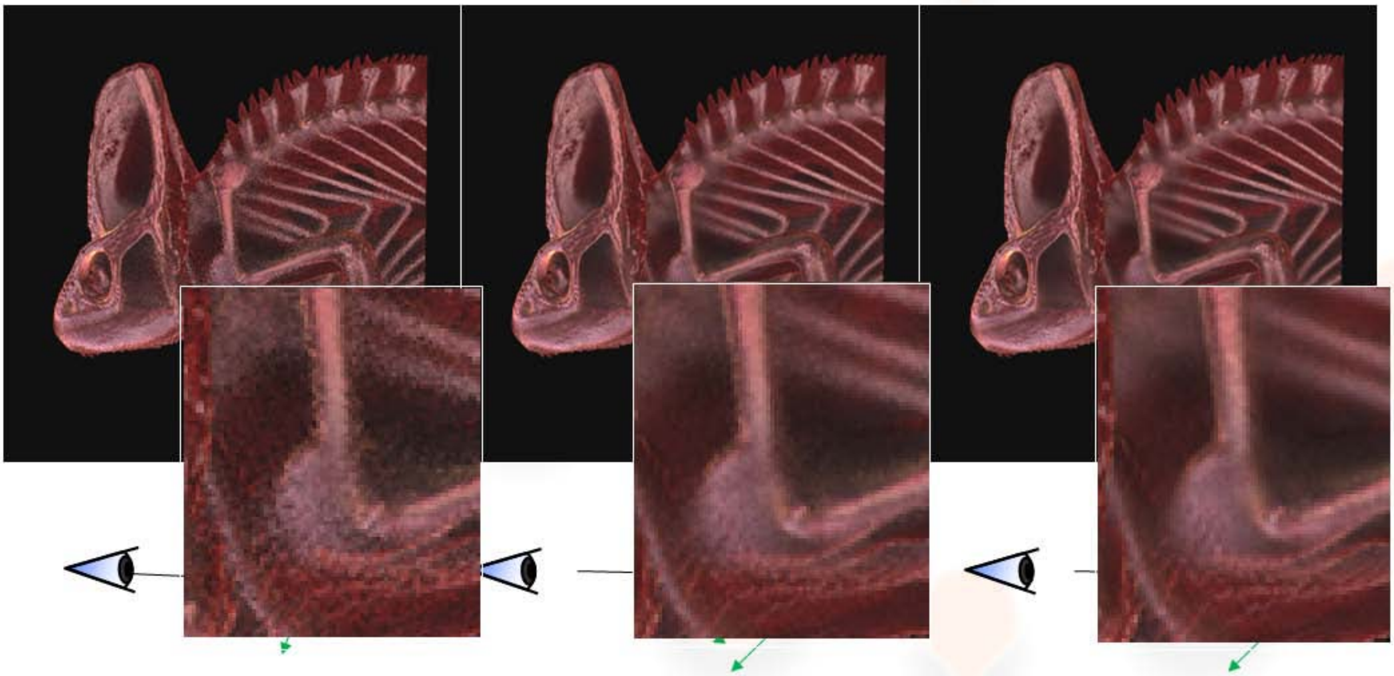
Path Tracing

Primary rays: 1 Secondary rays: 64	Primary rays: 8 Secondary rays: 8	Primary rays: 64 Secondary rays: 1



Path Tracing

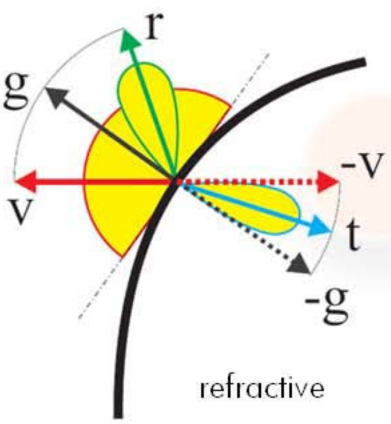
Primary rays: 1 Secondary rays: 64	Primary rays: 8 Secondary rays: 8	Primary rays: 64 Secondary rays: 1



CHRISTOF REZK-SALAMA, COMPUTER GRAPHICS GROUP, UNIVERSITY OF SIEGEN, GERMANY
ADVANCED ILLUMINATION TECHNIQUES FOR GPU-BASED VOLUME RAYCASTING

Examples

Different scattering cone angles for the „inward-looking" (transmissive) Phong-lobe



refractive

Vis08

Scattering Effects
Light Map Approaches

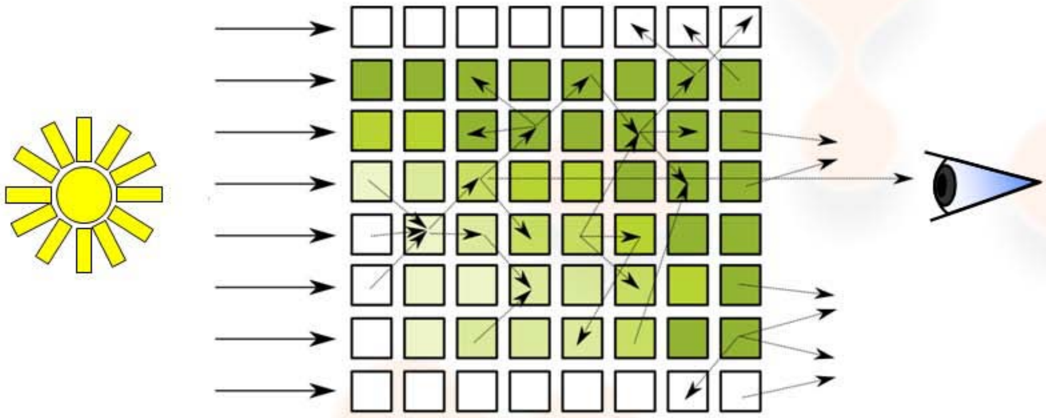
Markus Hadwiger
VR VIS Research Center
Vienna, Austria

Patric Ljung
Siemens Corporate Research
Princeton, NJ, USA

Christof Rezk Salama
Computer Graphics Group
Institute for Vision and Graphics
University of Siegen, Germany

Timo Ropinski
Visualization and Computer Graphics Research Group,
University of Münster, Germany

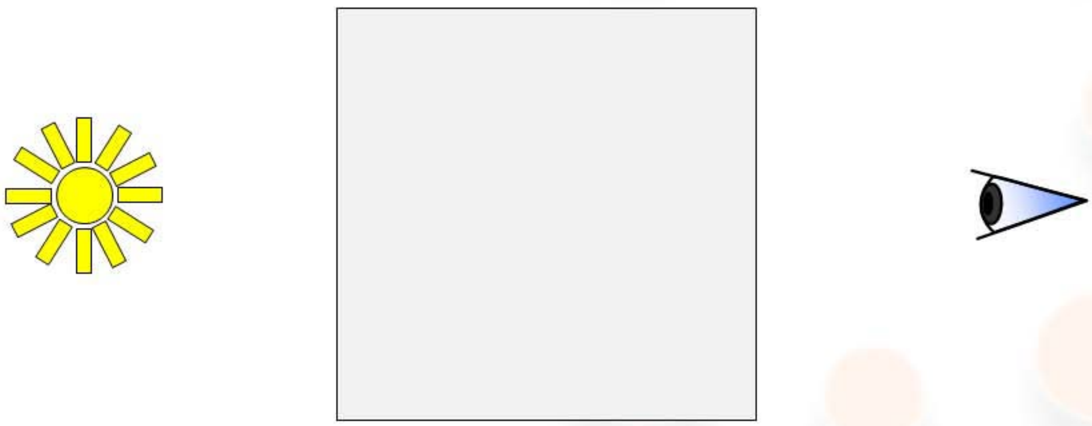
3D Light Map



- Direct light by shadow volume or deep shadow map
- Consider the *exchange of radiant energy between neighbouring voxels*
- Approximate by blur operation *(like [Kniss, 2002])*

Generate a 3D Light Map

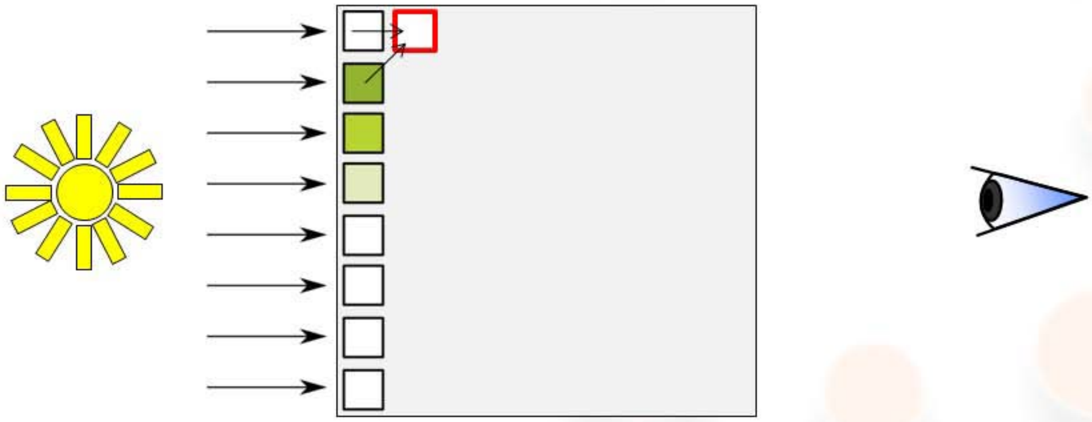
- **Based on Shadow Volume**
- Calculate shadow volume for direct light as in
  - U. Behrens and R. Ratering. **Adding Shadows to a Texture-Based Volume Renderer**. In Proc. IEEE Symposium on Volume Visualization, 1998, p.39–46.
- Blur the direct light slice by slice



CHRISTOF REZK-SALAMA, COMPUTER GRAPHICS GROUP, UNIVERSITY OF SIEGEN, GERMANY
ADVANCED ILLUMINATION TECHNIQUES FOR GPU-BASED VOLUME RAYCASTING

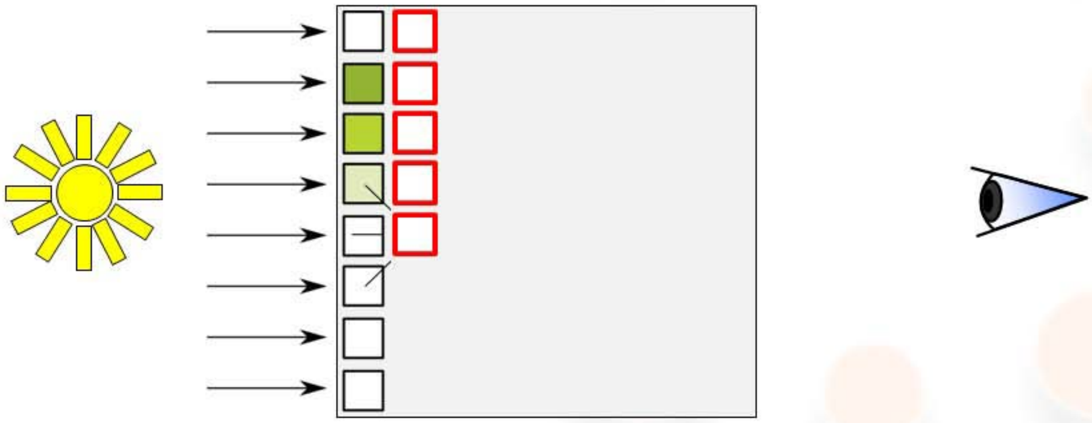
Generate a 3D Light Map

- **Based on Shadow Volume**
- Calculate shadow volume for direct light as in
  - U. Behrens and R. Ratering. **Adding Shadows to a Texture-Based Volume Renderer**. In Proc. IEEE Symposium on Volume Visualization, 1998, p.39–46.
- Blur the direct light slice by slice



Generate a 3D Light Map

- **Based on Shadow Volume**
- Calculate shadow volume for direct light as in
  - U. Behrens and R. Ratering. **Adding Shadows to a Texture-Based Volume Renderer**. In Proc. IEEE Symposium on Volume Visualization, 1998, p.39–46.
- Blur the direct light slice by slice



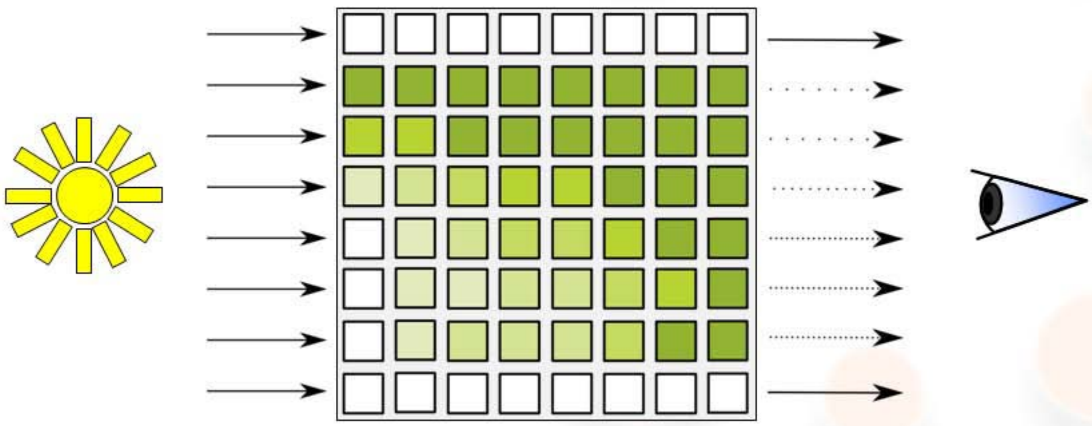
CHRISTOF REZK-SALAMA, COMPUTER GRAPHICS GROUP, UNIVERSITY OF SIEGEN, GERMANY
ADVANCED ILLUMINATION TECHNIQUES FOR GPU-BASED VOLUME RAYCASTING

Generate a 3D Light Map

- **Based on Shadow Volume**
- Calculate shadow volume for direct light as in
  - U. Behrens and R. Ratering. **Adding Shadows to a Texture-Based Volume Renderer**. In Proc. IEEE Symposium on Volume Visualization, 1998, p.39–46.
- Blur the direct light slice by slice

Generate a 3D Light Map

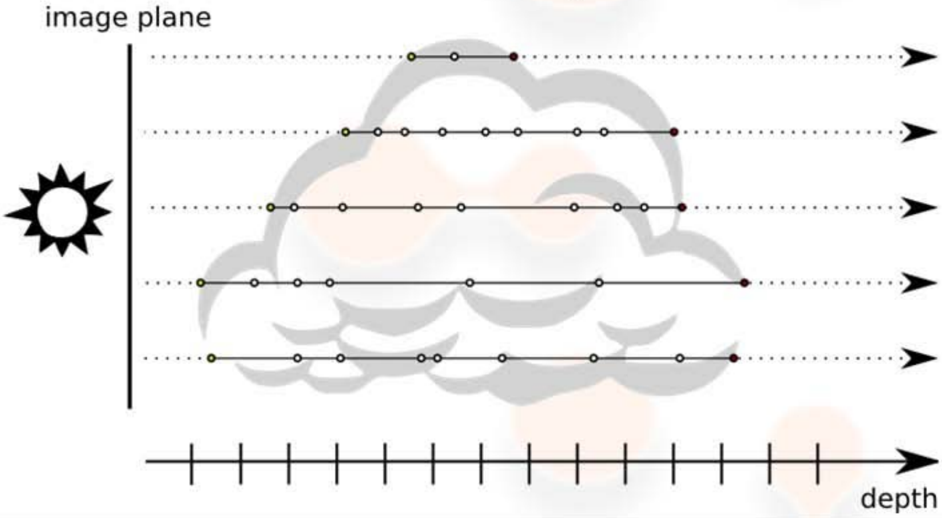
- **Based on Shadow Volume**
- Calculate shadow volume for direct light as in
  - U. Behrens and R. Ratering. **Adding Shadows to a Texture-Based Volume Renderer**. In Proc. IEEE Symposium on Volume Visualization, 1998, p.39–46.
- Blur the direct light slice by slice



Scattering 3D Light Map

Calculate a 3D Light Map

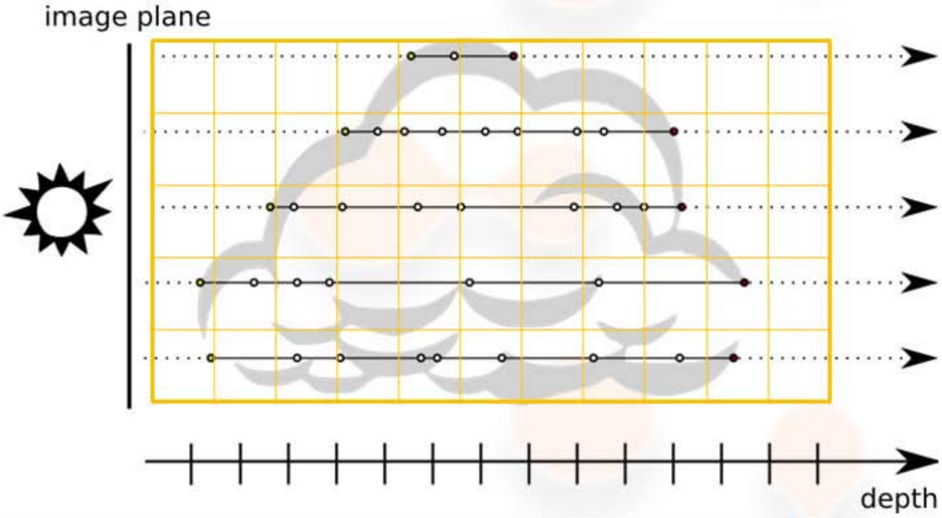
- Based on Deep Shadow Map
- Resample the deep shadow map on a *uniform voxel grid*
- *Coarse grid resolution* is sufficient due to the low-frequent nature of volumetric scattering



CHRISTOF REZK-SALAMA, COMPUTER GRAPHICS GROUP, UNIVERSITY OF SIEGEN, GERMANY
ADVANCED ILLUMINATION TECHNIQUES FOR GPU-BASED VOLUME RAYCASTING
Vis08

Calculate a 3D Light Map

- Based on Deep Shadow Map
- Resample the deep shadow map on a *uniform voxel grid*
- *Coarse grid resolution* is sufficient due to the low-frequent nature of volumetric scattering



Scattering Deep Shadow Map

Direct light

Direct plus indirect light

Light Map Approaches

Shadow Volume Appraoch

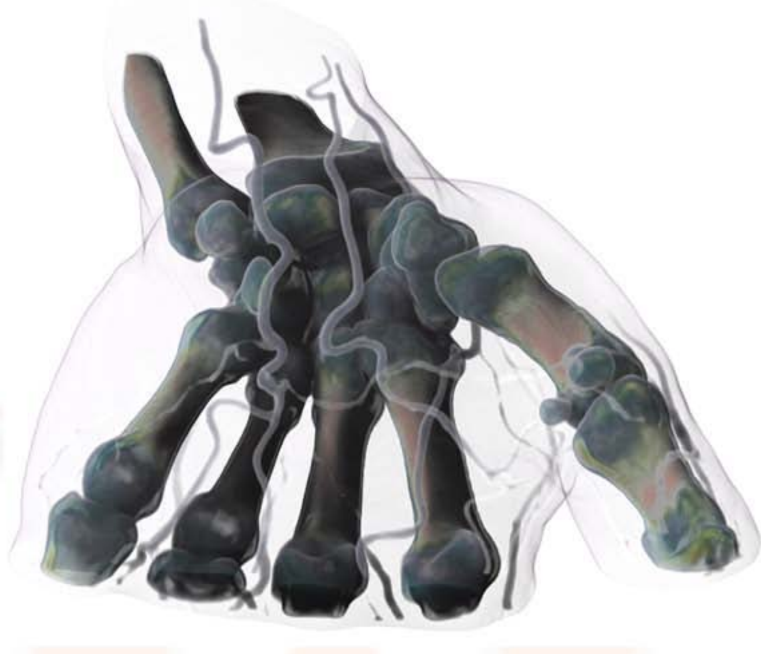
- Calculated in Model Space
  - Limited by Resolution of Shadow Volume
- High Memory Requirements

Deep Shadow Map Appraoch

- Calculated in Screen Space
  - Limited by Resolution of Shadow Volume
- Reduced Memory Requirements
- High Precision

High Dynamic Range

Direct light and shadows



Direct light, shadows and translucency

Summary

Scattering Effects

- **Single Scattering**
  - *Filtered Environment Maps*
  - *Monte-Carlo Integration*
- **Multiple Scattering**
  - *Monte-Carlo Integration*
  - *3D Light Maps (Shadow Volume/Deep Shadow Map)*

Acknowledgements

- Images and Slides on Deep Shadow Maps:
  Andrea Kratz, Zuse Institute, Berlin
  Markus Hadwiger, VRVis, Vienna
- Volume Data Sets:
  - Medical data sets courtesy of *Agfa Vienna and Dept. Of Neurosurgery, Medical University Vienna*
  - Chameleon, Cheetah, Bat, Pterosaur courtesy of *UTCT data archive, University of Texas at Austin*
  - Carp Data set courtesy of *Univ. of Erlangen-Nuremberg*

ADVANCED ILLUMINATION TECHNIQUES FOR GPU-BASED VOLUME RAYCASTING