UNIVERSITÄT DES SAARLANDES
Dr.-Ing. Hendrik P.A. Lensch
Max Planck Institut Informatik
Art Tevs (tevs@mpi-inf.mpg.de)
Boris Ajdin (bajdin@mpi-inf.mpg.de)
Matthias Hullin (hullin@mpi-inf.mpg.de)

29. November 2007

Computer Graphics I
Rendering Competition

Rendering Competition

During the rendering competition your task is to extend your ray tracing system with more advanced image generation techniques. On this sheet you will find a list of possible technical topics, from which you can choose. But we encourage you of course to come up with your own ideas as well.

The points assigned to the different topics roughly correspond to how hard it is to implement the respective algorithms. The provided references are just meant as hints. You are not supposed to implement every technical detail given in these articles/papers, but rather a basic realization that produces the desired effect.

Apart from improving your renderer you will also have to model a scene of your own, which demonstrates the implemented algorithms. For this part you can use the freely available modeling package *blender* (`www.blender.org`). Take also a look into a free trial version of *AC3D* (`www.inivis.com`).

The goal of the rendering competition is to produce (hopefully) stunning animation, which will be rated based on the technical and artistic achievement (see below).

Rule: You have to present your proposals, i.e. a list of effects/topics you would like to implement, to your Bremser in the last lecture week of this year. Hence in the week from 17th to 21st December.
Hence tutors will be able to provide you with useful tips according to your choice.

Important Dates

- Final deadline: **January 31st, 2008, 23:59h**
- intermediate deadline (low-res animation): **24th January, 23:59h**

General Information

- Submission deadline: **January 31st, 2008, 23:59h**
- A list of possible technical topics is given below. For each topic a value of difficulty ranging from 1 (easy) to 100 (hard) points is given.
- Groups of 2 or less people are allowed. For groups of 2 people you have to mark which topic is handled by whom.
- You have to create an animation of length t, where 15 sec $\leq t \leq$ 60 sec.
- You have to render your results with 24 frames per second, e.g. for a 30 second animation you have render out $30 \cdot 24 = 720$ frames.
- The rendering time of one frame shouldn't exceed 10 minutes, hence $t_{frame} \leq 10$ min. In the worst case you will require $\frac{60 \cdot 24 \cdot 10}{60} = 240$ hours or **10** full days to render your results.
- You have to cover at least three topic groups with at least one topic each.
- For a pre-submission on **24th January, 23:59h** create a **gzipped tar** file which contains:
  - A complete animation sequence rendered with 320×256 resolution with low quality settings as an avi file.
  - Well documented (!) source code (including a **makefile**) used to generate the animation sequance.
  - All data needed to reproduce the sequence (textures, models, etc.), as well as some information where you got the data from (generated by yourselves, downloaded , etc.).
- One week later (on **January 31st, 2008, 23:59h**) you have to submit a **gzipped tar** file which contains:
  - The **same animation sequence** as submitted before in 800×600 resolution with high quality settings.
  - A **web page** that contains:
    - Names of all participants.
    - Title of your animation.
    - A thumbnail of the animation (size 320×256) linking to the a large sized screenshot.
    - For each topic covered:
      - Name of the topic
      - A description and a corresponding screenshot where the effect of the technique could be perceived.
      - A short description of how the topic was implemented.
      - Links to the source code where the topic is implemented.
- We will provide exact guidelines on how to encode your animation, i.e. how to generate the avi file.
- In order to animate your scene you might either export a new scene for each frame or animate your scene in a procedural way. The simplest animation would consist of a camera movement only.
- A jury will give marks on:
  - the technical merit, judging complexity – 60%
  - the artistic merit – 30%
  - quality of the implementation – 10%
- To pass the rendering competition you have to reach at least 100 points if you work alone or 200 for groups of 2 people.

- The mark of the rendering competition makes up 50% of the final grading of the course.
- Send your contribution to (email subject: `[CG0708 Animation]`) :

  `tevs@mpi-inf.mpg.de`

Rendering Topics

- Advanced Camera Properties
  - **Depth of Field (20 points)**
    Simulation of a thin lense with varying focus. [7], see also [19]
  - **Motion Blur (20 points)**
    Rendering with a camera shutter open for a small time interval. [19]
  - **Tone Mapping (30 points)**
    Used to scale pixel radiance quantities to reasonable levels. "A Contrast-Based Scalefactor for Luminance Display" in [10], see also [19]
- Surface Shading
  - **Reflective and Refractive Transparency (30 points)**
    Refraction based on Snell's law including fresnel term and adaptive termination (termination depending on the influence to the current recursive ray to the final pixel). [5]
  - **Optical Bench Simulator (60 points)**
    Simulating thick lenses, apertures, beam splitters with chromatic aberration, vignetting, etc.
  - **Subsurface Scattering (60 points)**
    Rendering of the way light scatters around in some object [14].
  - **Procedural Shading (30 points)**
    Procedural generation of materials such as wood, marble, etc. [4]
  - **Physically-Based Surface Models (20 points)**
    Non empirical BRDF models, e.g. *Cook-Torrance*, *Torrance-Sparrow*, etc. [5]
- Modeling
  - **Point Ray Tracing (60 points)**
    Ray Tracing of point clouds [18].
  - **Fractal Geometry (50 points)**
    Fractal based geometric models, e.g. mountains. [4]
  - **Constructive Solid Geometry (CSG) (60 points)**
    Combination of simple primitives by means of boolean set operators. [5]
  - **Displacement Mapping (50 Points)**
    Mapping a height map onto an object. For rendering the triangulation of the object is refined and then rendered.
- Texturing
  - **Reflection Mapping (20 points)**
    Approximation of remote environments using images. [5]
  - **Bump Mapping (30 points)**
    Normal perturbation using height maps. [1], see also [5]
- Advanced Light Transport
  - **Instant Radiosity (60 points)**
    Light emitting objects with finite extension. [19]
  - **Image Based Lighting (60 points)**
    Illumination using high dynamic range environment maps with non trivial sampling. [3]
  - **Distributed Ray Tracing of Glossy Reflection (30 points)**
    Simulation of blurred specular reflection and translucent refraction of rough surfaces. [2], see also [6]
  - **Bidirectional Path Tracing (80 points)**
    Global illumination computation by stochastically sampling all possible light paths. [19]

  - **Photon Mapping (100 points)**
    Sending photons from light sources and storing them in the scene to compute indirect illumination. [13]

- Advanced Ray-Tracing
  - **Relativistic Ray Tracing (50 Points)**
    Incorporating of Einstein's relativistic formulation of light propagation into a ray tracer[11].
  - **Dispersion Effects in Ray Tracing (60 Points)**
    Physically based simulation of dispersion effects [25].
  - **Double Refraction Rendering in Crystals (50 points)**
    Realistic and physically plausible rendering of birefringence effect in crystals [24].

- Volume Rendering
  - **Integrated Intensity Volume Rendering (40 points)**
    Integration of scalar field intensities along rays, e.g. to simulate smoke. [15], see also [4]
  - **Iso-Surface Volume Rendering (60 points)**
    Computation of level surfaces. [16]

- Nonlinear Ray Tracing
  - **Nonlinear Ray Tracing (50 Points)**
    Ray tracing curved rays to simulate relativistic effects [8].
  - **Eikonal Rendering (60 Points)**
    Incorporating eikonal equation for simulating of light propagation through a refractive object. Can be seen as extension to volumetric rendering [12], [20].

- Optimization
  - **SAH KD Tree (50 points)**
    Optimizing the KD tree implementation using the surface area heuristic. Your construction algorithm must have runtime $O(n\ logn)$ or $O(n\ log^2 n)$. [22].
  - **BVH Structures (40 points)**
    Bounding volume hierarchies to otimize the rendering times. BVHs can be used for dynamic objects, since they are cheap to compute[17], [9].
  - **Rendering Cache (60 Points)**
    Caching and reusing previously rendered results (i.e. from the previous frames) to increase rendering performance [23]
  - **Multithreading (30 Points)**
    Incorporate all of your CPU cores to render the scene
  - **Packed ray tracing and optimized data structures (40 Points)**
    Packet ray tracing with SIMD instructions. Data structures optimized for CPU cache. Implement the propositions made by Ingo Wald in his PhD thesis [21].
  - As this exercises gives no artistic benefit you have to evaluate the construction performance and rendering performance of the implemented acceleration structure, and a non optimized ray tracer (i.e. last version of microTracer). Provide a small table for speed comparison on the webpage using 4 sample scenes with different complexity

- Fun
  - **Interactive Applications (40 points)**
    If you get you ray-tracer fast enough you might try to implement an interactive application, a simple game such as Tetris or Rubik's cube.
  - **Stereo Rendering (40 points)**
    Combine the two images of the left and the right eye into one image using red/green images.
  - **Physics Simulation (30 points)**
    Implement a small physics simulation to animate some of your objects.

- Any other topic not listed here
  If you like to implement other topics not listed here you are free to do this. Post your suggestions, i.e. fair suggestions, what you think is a fair merit to be given for that topic. Include all material you have used to implement the topic, e.g. conference papers (as pdf) etc. The jury will decide based on the complexity of the topic which mark you will earn.

References

[1] James F. Blinn. Simulation of Wrinkled Surfaces. *Computer Graphics (Proceedings of SIGGRAPH 78)*, 12(3):286–292, 1978.

[2] Robert Cook, Thomas Porter, and Loren Carpenter. Distributed Ray Tracing. *Computer Graphics (Proceeding of SIGGRAPH 84)*, 18(3):137–144, 1984.

[3] Paul Debevec. Rendering Synthetic Objects into Real Scenes: Bridging Traditional and Image-based Graphics with Global Illumination and High Dynamic Range Photography. *Computer Graphics (Proceedings of SIGGRAPH 98)*, 32(4):189–198, 1998.

[4] David Ebert, F. Kenton Musgrave, Darwyn Peachey, Ken Perlin, and Steven Worley. *Texturing & Modeling: A Procedural Approach*. Morgan Kaufmann, third edition, 2002. ISBN 1-55860-848-6.

[5] James Foley, Andries van Dam, Steven Feiner, and John Hughes. *Computer Graphics – Principles and Practice, second edition in C*. Addison Wesley, Reading, Massachusetts, U.S.A., 1997. ISBN 0-20184-840-6.

[6] Andrew Glassner. *An Introduction to Ray Tracing*. Morgan Kaufmann, 1989. ISBN 0-12286-160-4.

[7] Andrew Glassner. *Principles of Digital Image Synthesis*. Morgan Kaufmann, 1995. ISBN 1-55860-276-3.

[8] Meister Eduard Gröller. Nonlinear raytracing - visualizing strange worlds. *Visual Computer*, 11(5):263–274, 1995.

[9] Eric Haines. Efficiency Improvements for Hierarchy Traversal in Ray Tracing. In James Arvo, editor, *Graphics Gems II*, pages 267–272. Academic Press, 1991.

[10] Paul S. Heckbert, editor. *Graphics Gems IV*. Morgan Kaufmann, 1994. ISBN 0-12336-155-9.

[11] Andrew Howard, Sandy Dance, and Les Kitchen. Relativistic ray-tracing: Simulating the visual appearance of rapidly moving objects.

[12] Ivo Ihrke, Gernot Ziegler, Art Tevs, Christian Theobalt, Marcus Magnor, and Hans-Peter Seidel. Eikonal rendering: Efficient light transport in refractive objects. *ACM Trans. Graph.*, 26(3):59, 2007.

[13] Henrik Wann Jensen. *Realistic Image Synthesis Using Photon Mapping*. A K Peters Ltd, 2001. ISBN 1-56881-147-0.

[14] Henrik Wann Jensen, Steve Marschner, Marc Levoy, and Pat Hanrahan. A practical model for subsurface light transport. In *Proceedings of SIGGRAPH'2001, pages 511-518, Los Angeles*, August 2001.

[15] Michael Meißner, C.M. Wittenbrink, R.Westermann, and H. Pfister. Volume Visualization and Volume Rendering Techniques. *Eurographics tutorial*, 2000.

[16] Steven Parker, Peter Shirley, Yarden Livnat, Charles Hansen, and Peter-Pike Sloan. Interactive Ray Tracing for Isosurface Rendering. In *IEEE Visualization '98*, pages 233–238, October 1998.

[17] Steve M. Rubin and Turner Whitted. A three-dimensional represenatation for fast rendering of complex scenes. *Computer Graphics*, 14(3):110–116, July 1980.

[18] Gernot Schaufler and Henrik Wann Jensen. Ray tracing point sampled geometry. In *Rendering Techniques 2000. Eds. Peroche and Rushmeier. Springer-Verlag, pages 319-328*, 2000.

[19] Peter Shirley and R. Keith Morley. *Realistic Ray Tracing*. A K Peters Ltd, second edition, 2003. ISBN 1-568811985.

[20] Art Tevs. Realistic real-time rendering of refractive objects. Master's thesis, Department of Computer Science, Saarland University, 2007.

[21] Ingo Wald. *Realtime Ray Tracing and Interactive Global Illumination*. PhD thesis, Computer Graphics Group, Saarland University, 2004. Available at http://www.mpi-sb.mpg.de/~wald/PhD/.

[22] Ingo Wald and Vlastimil Havran. On building fast kd-trees for ray tracing, and on doing that in O(N log N). In *Proceedings of the 2006 IEEE Symposium on Interactive Ray Tracing*, 2006. (accepted for publication, minor revision pending).

[23] Bruce Walter, George Drettakis, and Steven Parker. Interactive rendering using the render cache. In D. Lischinski and G.W. Larson, editors, *Rendering techniques '99 (Proceedings of the 10th Eurographics Workshop on Rendering)*, volume 10, pages 235–246, New York, NY, Jun 1999. Springer-Verlag/Wien.

[24] Andrea Weidlich and Alexander Wilkie. Realistic rendering of birefringency in uniaxial crystals. *ACM Transactions on Graphics*, 2007.

[25] A. Wilkie, R. Tobler, and W. Purgathofer. Raytracing of dispersion effects in transparent materials. *WSCG Conference Proceedings*, 2000.