UNIVERSITÄT DES SAARLANDES
Prof. Dr. Philipp Slusallek
Lehrstuhl für ComputerGraphik
Sven Woop (woop@graphics.cs.uni-sb.de)
Stefan Popov (poppov@graphics.cs.uni-sb.de)

December 4, 2006

Computer Graphics I
Rendering Competition

Submission deadline for the rendering competition: January 21th, 2007, 23:59h

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.

The points assigned to the different topics roughly corresponds 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`).
The goal of the rendering competition is to produce (hopefully) stunning images, which will be rated based on the technical and artistic achievement (see below).

General Information

- 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. However, each student must implement at least 100 points, which means a group of two people must implement at least 200 points. Each student has to implement one topic with ≥ 40 points.
- For the submission on 21th January create a **gzipped tar** file which contains:
  - A single image as a 320×256 sized JPG with low quality settings.
  - Well documented (!) source code (including a **makefile**) used to generate the image.
  - All data needed to reproduce the image (textures, models, etc.), as well as some information where you got the data from (generated by yourselves, downloaded , etc.).
- Two days later (on 23th January, 23:59h) you have to submit a **gzipped tar** file which contains:
  - The **same image** as submitted before in 1280×1024 resolution with high quality settings.
  - A **web page** that contains:
    - Names of all participants.
    - Name of the image.
    - A thumbnail of the image (size 320×256) linking to the large sized image.
    - For each topic covered (only topics mentioned here will be rated):
      - Name of the topic.
      - A description where in the image 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.
- Send your contribution to:

  `woop@graphics.cs.uni-sb.de`
- A jury will give:
  - a technical merit
  - and an artistic merit
- The mark of the rendering competition makes up 40% of the final mark.

Rendering Topics

- Advanced Camera Properties
  - **Depth of Field (20 points)**
    Simulation of a thin lense with varying focus. [7], see also [14]
  - **Motion Blur (20 points)**
    Rendering with a camera shutter open for a small time interval. [14]
  - **Tone Mapping (30 points)**
    Used to scale pixel radiance quantities to reasonable levels. "A Contrast-Based Scalefactor for Luminance Display" in [8], see also [14]
- 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]
  - **Subsurface Scattering (100 points)**
    Rendering of the way light scatters around in some object [10].
  - **Procedural Shading (50 points)**
    Procedural generation of materials such as wood, marble, etc. [4]

    a.) Procedural Bump mapping (+10 points)

  - **Physically-Based Surface Models (30 points)**
    Non empirical BRDF models, e.g. *Cook-Torrance*, *Torrance-Sparrow*, etc. [5]

- Modeling
  - **Point Ray Tracing (60 points)**
    Ray Tracing of point clouds [13].
  - **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. [14]
  - **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. [14]
  - **Photon Mapping (100 points)**
    Sending photons from light sources and storing them in the scene to compute indirect illumination. [9]

- Volume Rendering
  - **Integrated Intensity Volume Rendering (60 points)**
    Integration of scalar field intensities along rays, e.g. to simulate smoke. [11], see also [4]
  - **Iso-Surface Volume Rendering (80 points)**
    Computation of level surfaces. [12]

- Spatial Index Structures
  - **SAH KD Tree (60 points)**
    Implement a KD tree traversal and KD tree construction using the surface area heuristic. Your construction algorithm must have runtime $O(n\ logn)$ or $O(n\ log^2 n)$. As this exercise gives no artistic benefit you have to evaluate the construction performance and rendering performance of the SAH KD trees, and a naive implementation that performs a spatial median split. Provide a small table for speed comparison on the webpage using 4 sample scenes with different complexity [15].

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

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

[10] 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.

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

[12] 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.

[13] 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.

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

[15] 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).