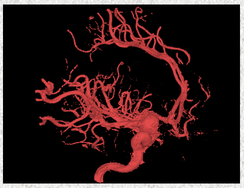
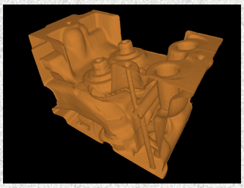
Fast and Accurate Ray-Voxel Intersection Techniques for Iso-Surface Ray Tracing

informatik
saarland.

Introduction

Volume rendering visualizes data values that are provided at samples distributed in 3D space. Here, we consider surfaces defined by an implicit function on the volume data values, called iso-surface rendering.

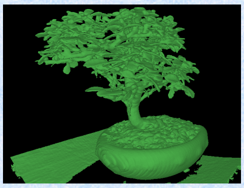




Our approach aims particularly at interactive rendering using ray tracing. Assuming rectilinear organized data, calculating the intersection point within these values is a crucial task.

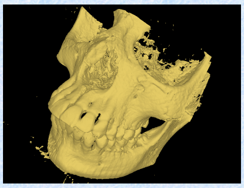
Accurate Intersection Method

In a first step, a trilinear interpolation is applied to calculate the density at any point within the voxel. Several mathematical operations lead to a cubic polynomial. This can be solved using Schwarzes Analytical Inversion.
However, this involves costly operations and can lead to precision problems.



trilinear interpolation for ρ:

$$\rho(u,v,w) = \sum_{i,j,k\in\{0,1\}} u_i v_j w_k \rho_{ijk}$$



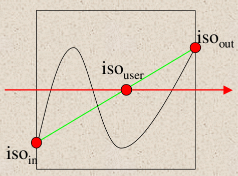
ray intersection for ρ:

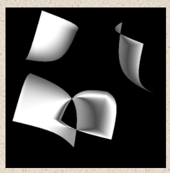
$$R(t) = a + tb$$

$$\rho(t) = \sum_{i,j,k\in\{0,1\}} (u_i^a + tu_i^b)(v_j^a + tv_j^b)(w_k^a + tw_k^b)$$

Linear Interpolation

The density at the entry and exit point of the voxel is calculated bilinearly followed by a linear interpolation to obtain the final intersect position.

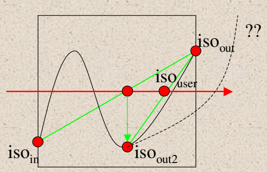


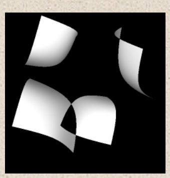


☺ fast and easy to implement
☹ artifacts and holes (see picture for illustration)

Neubauer's Method

Applying the previous algorithm iteratively leads to a refinement of the intersection point which gets very close to the accurate method.



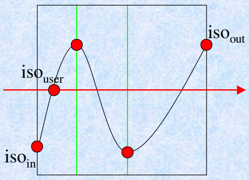


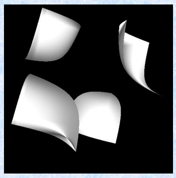
☺ fast and easy to implement
☹ artifacts are less obvious, but holes can still not be avoided (see picture for illustration)

Isolation and Iterative Root Finding

Key steps of our new proposed algorithm:

1. Calculate the Coefficients of the cubic polynomial
2. Calculate the Extremas
3. Reset search area for intersection point to the first segment where the sign of ρ changes
4. Since there is at most one intersection within each segment, iterative refinement will find the correct intersection





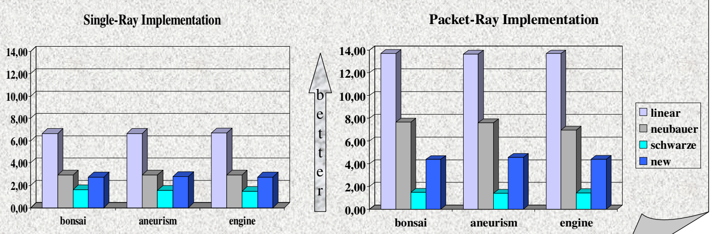
☺ fast and easy to implement
☺ accurate results and no holes (see picture for illustration)

Parallel SIMD Implementation

The SSE-Extension of modern Processors allows to process one operation on four data values at the same time. Our new approach shows much less overhead then the Schwarze Method when using SIMD instructions.

Experiments and Results

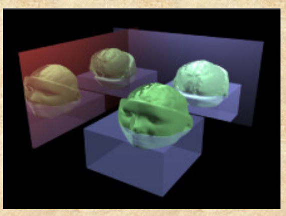
We tested three typical mid-sized data sets on an AMD Opteron with 1.8 GHz and 2 GB RAM. Our new algorithm clearly outperfoms Schwarzes Analytical Inversion since it is roughly as fast as Neubauer's Approximation.



million intersections per second

Conclusion and Future Work

Our new approach is able to reproduce the reliability of accurate intersection methods while preserving the speed of approximate techniques. A simplier control-flow makes Neubauers Method a little faster if one want to use SIMD instructions but can produce holes.



Future extensions aim at a better incorperation into the OpenRT environment allowing global illumination and mixed surface-volume rendering. The left figure shows preliminary results.

References

[Wal04] Ingo Wald. Realtime Ray Tracing and Interactive Global Illumination. PhD thesis, Computer Graphics Lab, Saarland University, 2004

[NMHW02] A. Neubauer, L. Mroz, H. Hauser, and R. Wegenkittle. Cell-based first-hit ray casting. In Proceedings of the Symposium on Data Visualization 2003, pages 77-ff, 2002

[PSL+98] S. Parker, P. Shirley, Y. Livnat, C. Hansen, and P. Sloan. Interactive Ray Tracing for Isosurface Rendering. In IEEE Visualization '98. pages 233-238, October 1998

Gerd Marmitt, Andreas Kleer, Ingo Wald, Heiko Friedrich and Philipp Slusallek

Computer Graphik Universität des Saarlandes

Computer Graphics Lab Saarland University