Master Project - Real Time Rendering of skeletal structures - Motivations Goals and Plan

OLIVIER ROUILLER - S090842
Department of Informatics and Mathematical Modelling
Technical University of Denemark

February 16, 2011

1 Our problem

Design an algorithm to raytrace efficiently a implicit surface defined by a skeleton.

- The algorithm must raytrace the surface efficiently, if possible as fast as rasterizing a mesh.
- The algorithm must get the entire surface.
- The algorithm must allow dynamic skeleton.
- The algorithm must be integrable in a classical rendering pipeline (deferred or direct?).
- The algorithm must allow as many effect as possible.

How to speed up the ray-tracing?

- Shoot ray only where there is a surface intersection. Involves spatial subdivision techniques such as BVH, Octree, ...
- Speed up intersection. Choice of root finding algo, using geometry.
- Evaluate only the primitives that contributes at the point of evaluation. spatial sorting.

2 Techniques for raytracing Implicit surfaces

Hart gives a good intro to different techniques in Ray-Tracing implicit surfaces.

2.1 Polynomial Root Solving

Works for algebraic surfaces (in our scope).

First work seems to be from Hanrahan [Han83]. His method used ”a symbolic algebra system to automatically derive the equation of intersection between the ray and the surface and then solves this equation using an exact polynomial root finding algorithm”. This implies converting the space function of the surface to a univariate function along the ray and then solve this polynomial equation. Having the coefficients of the ray polynomial, numerical methods for root finding are used.

The same approach is used in [LB06] but the univariate ray polynomial is computed using Bezier tetrahedra.

2.2 Interval analysis

A common method to find root of a function along ray is to use interval analysis, isolating the root in an interval then refining this interval to approximate the root. [Mit90].

2.3 Lipschitz methods

Interval analysis can be improved by using the local Lipschitz constant of the function at each iteration.

2.4 Sphere Tracing

Another method to raytrace an implicit surface is to use sphere tracing [Har96]. This method does not involve root finding but finds the first intersection of the ray with the surface. This relies on knowing the distance of the current point on the ray to the surface. If this distance is less than ϵ, we found the intersection, otherwise we know that we can move forward with the distance to the surface. It seems that in the case of convolution surfaces, we know this distance.

References

[Han83] Pat Hanrahan. Ray tracing algebraic surfaces. In *Proceedings of the 10th annual conference on Computer graphics and interactive techniques*, SIGGRAPH ’83, pages 83–90, New York, NY, USA, 1983. ACM.

[Har96] John C. Hart. Sphere tracing: a geometric method for the antialiased ray tracing of implicit surfaces. *The Visual Computer*, 12:527–545, 1996. 10.1007/s003710050084.

[LB06] Charles Loop and Jim Blinn. Real-time gpu rendering of piecewise algebraic surfaces. In *ACM SIGGRAPH 2006 Papers*, SIGGRAPH ’06, pages 664–670, New York, NY, USA, 2006. ACM.

[Mit90] D. P. Mitchell. Robust ray intersection with interval arithmetic. In *Proceedings on Graphics interface ’90*, pages 68–74, Toronto, Ont., Canada, Canada, 1990. Canadian Information Processing Society.