Real time ray tracing of skeletal implicit surfaces

Rouiller, Olivier; Bærentzen, Jakob Andreas

Publication date:
2011

Document Version
Publisher's PDF, also known as Version of record

Real time ray tracing of skeletal implicit surfaces

Olivier Rouiller[a] and Jakob Andreas Bærentzen[a]
[a] Technical University of Denmark

Abstract

Modeling and rendering in real time is usually done via rasterization of polygonal meshes. We present a method to model with skeletal implicit surfaces and an algorithm to ray trace these surfaces in real time in the GPU. Our skeletal representation of the surfaces allows to create smooth models easily that can be seamlessly animated and textured. The ray tracing is performed at interactive frame rate thanks to an acceleration data structure based on a BVH and a kd-tree.

Skeletal implicit surface representation

In our method, surfaces are represented as implicit surfaces defined by a skeleton. The surface is the 0 level set of a potential field F defined by ditance a function to geometric primitives Pi such as points and line segments.

$$F(p) = \sum_{P_i \in P} f(\|p - P_i\|) - T$$

For the distance function, we use degree four functions inside a radius R around the primitives.

$$f(r) = (1 - \frac{r^2}{R^2})^2, r < R; 0, r \geq R$$

This compact and high level representation allows fast editing of smooth surfaces and procedural generation. The surface is ray traced on the GPU by root finding.

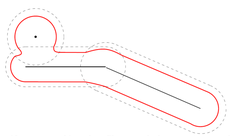

Figure 1 : Implicit surface (in red) defined by a point and two line segments.

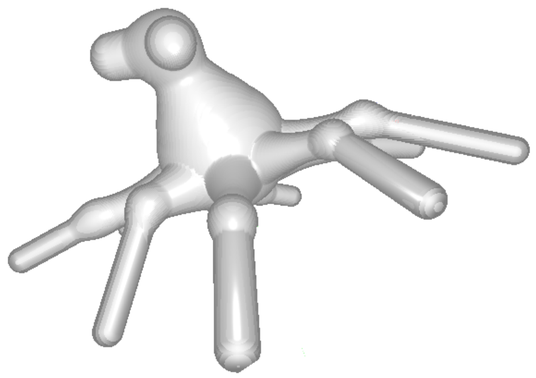

Figure 2 : A monster modeled and rendered with our method.

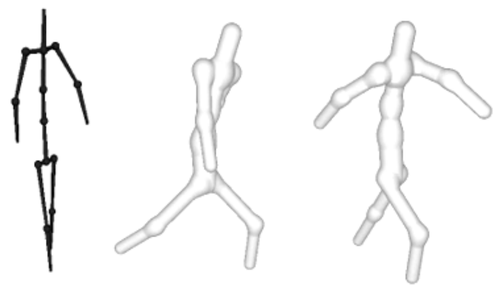

Figure 3 : A skeleton and the corresponding animated surface.

Figure 4 : A textured model.

Ray tracing acceleration

The ray tracing is accelerated using a bounding volume hierarchy fitting the geometric primitives. The BVH is built at each frame on the CPU and sent to the GPU to discard missing rays and to evaluate only the function with the primitives that are contributing to the field along the ray.

The BVH is built from top to bottom and at each level, primitives are allowed to be split as in a kd-tree construction as illustrated in Figure 5.

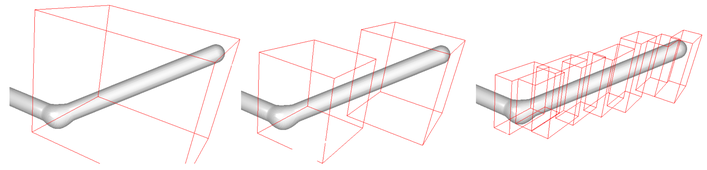

Figure 4 : Illustration of the primitive split done during the BVH creation.

Implementation and Results

The modelling system and the ray tracing algorithm was implemented using OpenGL and the ray tracing is done on the fragment shader. The BVH is stored on the GPU's texture memory. The method allows animation and texturing of the models as well as effects such as relief mapping. Texturing can be done easily per primitive basis.

Future work

We have demonstrated that ray tracing animated models made of skeletal implicit surfaces is possible in real time. It would be interesting to investigate the integration of more primitives and other CSG operations such as union and difference.