European Study Croup with Industry'113 (ESGI'113)
September 14–18, 2015, Sofia, Bulgaria

Problem 1. Spline intersection improvement

Chaos Group Ltd., www.chaosgroup.com

Yordan Mandzhunkov, yordan.madzhunkov@chaosgroup.com

Company's overview. Chaos Group creates physically-based rendering and simulation software for artists and designers. Founded in 1997, Chaos Group is a Bulgarian company that has devoted the last 18 years to helping artists advance the speed and quality of one of their most important tools. Today, Chaos Groups photorealistic rendering software, V-Ray®, has become the rendering engine of choice for many high-profile companies and innovators in the design and visual effects industries.

A task that we typically perform is intersection of spline curves with a ray. We already have developed several models how to intersect such primitives with rays. However, we are looking for a way to improve our current model in terms of accuracy, without sacrificing too much computation speed.

Problem. We model spline curve primitives with 4 control points in 3D space – p_0; p_1; p_2; p_3. Each control point has it's own width of the curve – w_0; w_1; w_2; w_3. Spline curve center as function of curve's evolution parameter $u \in [0, 1]$ is described with

$$\vec{p}(u) = \vec{p}_3 u^3 + 3\vec{p}_2 u^2(1-u) + 3\vec{p}_1 u(1-u)^2 + \vec{p}_0(1-u)^3. \tag{1}$$

The width of the primitive as function of the same evolution parameter is given by:

$$w(u) = w_3 u^3 + 3w_2 u^2(1-u) + 3w_1 u(1-u)^2 + w_0(1-u)^3. \tag{2}$$

The point that lies on the surface of the primitive has to satisfy the system:

$$\begin{aligned} &|\vec{s}(u) - \vec{p}(u)|^2 = w(u)^2 \\ &(\vec{s}(u) - \vec{p}(u)) \cdot \frac{d\vec{p}(u)}{du} = 0. \end{aligned} \tag{3}$$

For a given ray

$$\vec{r}(t) = \vec{o} + t\vec{d} \tag{4}$$

find $t > 0$ and $u \in [0, 1]$ such that

$$\vec{r}(t) = \vec{s}(u). \tag{5}$$

In case of multiple solutions, we are only interested in the one that has minimum t. We are interested in numerical method that is programmable in c++ and can find accurate solution at the lowest computational cost.

Motivation. The spline curve primitives we described above are widely used for hair in V-Ray. In scene that has one human like character there are typically several million hair strands. Each hair can be composed by many spline primitives. Therefore we need to be able to intersect hair as fast as possible. We have already implemented acceleration data structures as bounding volume hierarchy (BVH) and k-d tree to accelerate intersection. We will use the spline intersection method that you provide, as last phase of the intersection process – after the ray already intersected the bounding box of the spline. We also included a rendered image using our current intersection model: