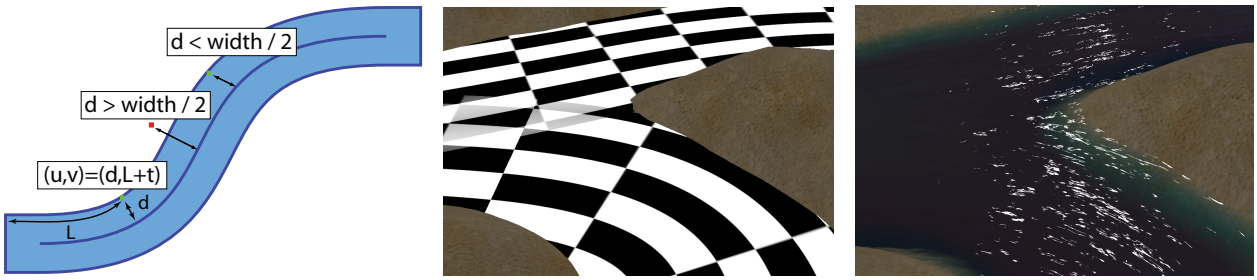
Real-time Rendering of River Networks

Quintijn Hendrickx [1] Ruben Smelik [2] Rafael Bidarra [1]

[1]Computer Graphics & CAD/CAM Group, Delft University of Technology, The Netherlands
[2]Modelling, Simulation & Gaming Department, TNO Defence, Security and Safety, The Netherlands



(a) (b) (c)

Figure 1: (a) Projection onto a river curve, (b) texture mapping on Bézier curves, (c) final result: water flowing through a river

Realistic rendering of water bodies such as rivers and oceans has proven to be one of the most difficult challenges in computer graphics. This challenge can be split into two main problems: simulating the movement of water and simulating the optical properties of water. This poster focuses on the problem of simulating water movement in complex river networks with various kinds of junctions.

Different solutions have already been proposed that vary widely in level of realism versus applicability in real-time systems. Recent work includes several different kinds of particle systems, such as a screen-space particle system [Yu et al. 2009] and an optimized three dimensional particle system [Kipfer and Westermann 2006]. Particle systems are an intuitive approach for simulating water flow but often require large amounts of memory and computation power.

This poster presents an efficient technique for real-time rendering of complex river networks without using any kind of particle system. Instead, Bézier curves and streaming normal maps are used to simulate the flow of water through rivers. The curves represent the geometric features (path and width) of a river. Multiple quadratic Bézier curve segments are linked together to create more complex river curves and junctions.

A commonly used method to visualize Bézier curves is to sample along the curve at a fixed rate, and then tessellate these samples into a geometric structure. However, to achieve smooth results, many samples are needed, resulting in a high vertex count. Because this is often not desirable in real-time rendering, we render the Bézier curves with bounding quads using only four vertices per curve. An implicitly defined distance field is used to project each pixel in the quad onto the nearest point on the curve. Using only quadratic order Bézier curves allows us to define the distance field as a function of the Bézier control points, which does not require any iterative algorithms. As a result this function is, due to its parallel nature, particularly suited for being evaluated on the GPU. Because of the low vertex count, no LOD techniques are necessary for large scale river networks, and rendering performance depends mostly on the total surface of visible water in screen-space.

The distance d to the corresponding projected pixel and the arc length L along the curve are calculated as shown in Figure 1a. The distance from the pixel to the curve is used to discard pixels that are not within the boundaries of the river. This results in an accurate and smooth curve rendered with only a very small number of vertices. Calculating the arc length along the Bézier curve allows us to map a texture onto the river surface. Adding a time-dependent offset t to this mapping will smoothly stream the texture along the curve.

Traditional tessellation methods for Bézier curves are typically unsuitable for junctions of curves. The produced geometry for each curve segment would overlap and not allow for complex blending between curves. Our method is able to visualize complex junctions by grouping overlapping segments into a single larger bounding quad. Each pixel in this quad will be projected onto all of the river segments. If a pixel maps onto multiple curves the final result will be interpolated based on the distance to these curves. Figure Figure 1b shows the result of mapping a texture on a series of linked Bézier curves with a simple junction.

This technique has been implemented as an extension to the open source osgOcean nodekit [osgOcean], which is part of OpenSceneGraph. See Figure 1c for a still of the achieved effect, and the demo clip at `http://graphics.tudelft.nl/~ruben/rivernetworks.wmv` for a better impression. Currently there is no smooth transition between the rivers and the ocean water of osgOcean, but this is an important goal for future work.

In conclusion, the use of Bézier curves to model and render river networks has proven to be an efficient method to produce convincing results of flowing water in complex environments.

References

KIPFER, P., AND WESTERMANN, R. 2006. Realistic and interactive simulation of rivers. In *Graphics Interface*, Canadian Human-Computer Communications Society, C. Gutwin and S. Mann, Eds., 41–48.

osgOcean: An ocean rendering nodekit for openscenegraph. *http://code.google.com/p/osgocean*.

YU, Q., NEYRET, F., BRUNETON, E., AND HOLZSCHUCH, N. 2009. Scalable real-time animation of rivers. *Comput. Graph. Forum 28*, 2, 239–248.