Real-Time Procedural Volumetric Fire

Alfred R. Fuller* Hari Krishnan† Karim Mahrous‡ Bernd Hamann§ Kenneth I. Joy¶
Institute of Data Analysis and Visualization and Department of Computer Science,
University of California, Davis, Davis, CA 95616

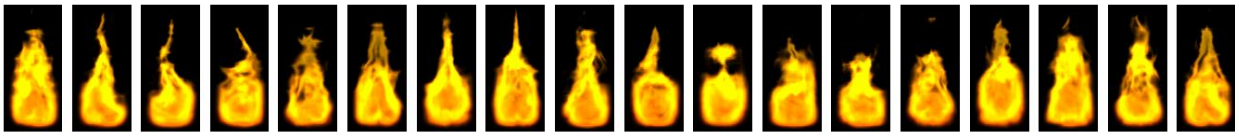

Figure 1: The method presented uses hardware acceleration, a realistic particle simulation and Improved Perlin Noise to create real-time procedural volumetric fire that supports macro-scale bi-directional environmental interactivity. The fires shape and color is modified via a two-dimensional fire profile texture. The motion of the fire can be pre-animated for scripted events or interactively generated by the particle simulation. The method creates visually convincing real-time three-dimensional fire, without the artifacts commonly seen in sprite systems or the computational costs associated with true photo-realistic fire simulation and rendering.

We present a method for producing real-time procedural volumetric fire. Through hardware acceleration, "Improved Perlin Noise" and a realistic particle simulation, we are able to render a uniquely animated vibrant volumetric fire that supports bi-directional environmental macro-interactivity. Our system allows a content artist to easily modify the shape and color of the fire by manipulating a two-dimensional fire profile texture.

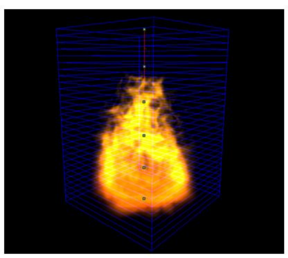

Figure 2: The lattice surrounding the fire is extremely flexible while maintaining texture coordinates for easy rendering. Four corner splines are generated having a distance of $\sqrt{2} * r(t)$ units away from the center spline. The center spline is sampled by length to find the associated position on the corner splines. This produces both the positions and texture coordinates of each corner in our lattice.

The fire is animated both on the macro and micro levels. Macro changes in shape and size are controlled by a four-dimensional central curve. The central curve is deformed by either a pre-scripted sequence of movements, or by a realistic particle simulation that takes into account movement, wind, high-energy particle dispersion and thermal buoyancy. The central curve is used to define a free-form volumetric deformation that is implicitly applied during rendering. Micro fire effects such as individual flame shapes, location, and flicker are generated in a pixel shader using four-dimensional Improved Perlin Noise.

The volume is rendered by constructing an arbitrarily detailed lattice around the central curve, slicing this lattice into quadrilateral planar polygons, and rendering these polygons with appropriate texture coordinates into a unit fire volume. The pixel shader uses Improved Perlin Noise in conjunction with the two-dimensional fire profile texture to generate the micro changes that "bring the fire to life." The constructed lattice supports quick and efficient collision detection which, when combined with a sufficiently intelligent particle simulation, enables real-time bi-directional interaction between the fire and its environment. The result is a truly three-dimensional procedural fire that is easily designed and animated by content artists, supports dynamic interaction, and is rendered in real-time.

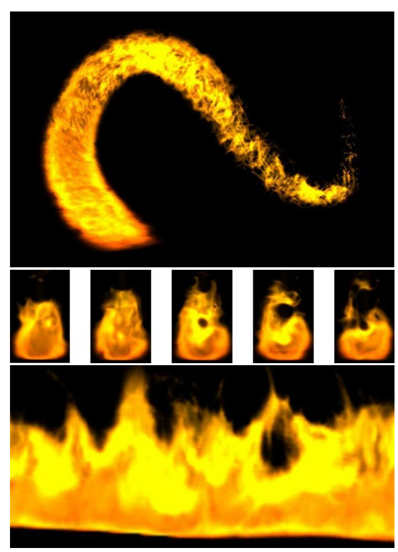

Figure 3: Improved Perlin Noise is used to animate the micro details of the fire. The level of detail of the Improved Perlin Noise is scaled to world coordinates in the pixel shader to eliminate stretching and scaling artifacts. Additionally, the perceived burning speed of the fire can be passed as a parameter to the pixel shader.

*e-mail:arfuller@ucdavis.edu
†e-mail:hkrishnan@ucdavis.edu
‡e-mail:kmmahrous@ucdavis.edu
§e-mail:bhamann@ucdavis.edu
¶e-mail:joy@cs.ucdavis.edu