Painterly Fire

Saty Raghavachary
DreamWorks Feature Animation
saty@anim.dreamworks.com

Fernando Benitez
DreamWorks Feature Animation
fbenitez@anim.dreamworks.com

Abstract

This sketch describes how to construct a painterly 'wall' of fire, one which exhibits realistic motion while managing to maintain an artistic look.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Splines; I.3.6 [Computer Graphics]: Methodology and Techniques—Graphics data structures and data types; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

Keywords: effects animation, rendering, natural phenomena

1 Introduction

For our third animated feature "Spirit - Stallion of the Cimmaron", we needed to create a ravaging fire which threatens the life of Spirit and other animals in the forest. This sketch describes our technique used to achieve this objective.

2 Technique

We build upon 'Spryticle', a sprite system created inside Maya by our colleague Doug Ikeler, for use in "The Road to ElDorado"[1]. Spryticle predominantly has flat cards that always face the camera (the cards do allow for some bulge deformation). In contrast, our technique, called 'SpriteRibbons'[2] creates multi-segment flexible sprites analogous to streamers, and the segments are free to be oriented along arbitrary directions.

The gist of our process is as follows. The 'backbone' of each sprite is generated as a space curve by linking together instantaneous particle world-space locations across a series of frames. This backbone gets expanded out to a multi-segment sprite on which traditional hand-drawn texture cycles are mapped.

We begin by emitting particles in Maya near the ground plane. The particles are guided along upward paths by a Maya plugin called 'dwPathField'[3]. These particles have extra attributes such as sprite-scale, sprite-normal, etc.to be used at the ribbon generation stage. The particle data is exported to disk in the form of Dynamation '.pda' files. Next a MEL script called 'SpriteRibbons.mel' is invoked to specify parameters for the sprite ribbon generation. These parameters include the length of the ribbons (in terms of particle frame numbers relative to the current frame), patch type (bilinear or bicubic) and whether or not at each frame, the sprite segments should each have their scale/normal/color/opacity or simply borrow the values of the central segment. A standalone program 'createSpriteData' reads the .pda files and from the ribbon-generation information specified above, creates '.xpda' files (a custom extension of the Dynamation .pda format). The .xpda files contain normal and tangent information, making it easy for the next step (rendering) to generate sprite patch geometry.

The sprites themselves are generated in RenderMan at rendertime, using a procedural modeling DSO[4]. The alternative to such just-in-time geometry creation would be to create them in Maya, with the attendant increase in scene complexity, scene file size and exported RIB file size. In MTOR, a 'RIBBox' is inserted with a call to the DSO ("RManSprites.so"). The DSO needs just the name of the .xpda file at each frame, and MTOR inserts this during RIB export. It reads the .xpda files and creates either bilinear or bicubic spline patches as specified by the user. Standard RenderMan surface shaders can be used to shade the sprites. We color the sprites using ramps, and use hand-painted flame shape texture cycles to matte out the colors. This gives a pleasing appearance to each sprite as it winds up its trajectory, changing flame color, shape and opacity as it does.

By varying the texture-mapped flame shapes, adjusting the length of the sprites, layering sprites with differing lengths and adjusting their widths, we are able to acieve a variety of looks for our fire.

The fire elements were used to color the background and also to warp it, both of which helped to integrate the elements into the scene. Further, smoke and embers were generated using the same particles which were used to create the sprites, and this added an extra touch of realism to the generated imagery.

[1] Doug Ikeler: *The Use of Spryticle in the Visual FX for "The Road to El Dorado"*, SIGGRAPH 2000 Conference Proceedings and Abstracts.

[2] Doug Cooper: internal communication.

[3] initially developed by Patrick Witting, extended and supported by Gigi Yates.

[4] www.pixar.com, PhotoRealistic RenderMan Application Note #23.