Fast Solutions to Gas Volumetrics in Matrix Revolutions

Johnny Gibson
Digital Domain
jmgibson@d2.com

Davy Wentworth
Tippett Studio
davy@tippett.com

1 Introduction

The "real world" effects in *Matrix Revolutions* escort the audience through a vast network of misty underground piping, the dense sludge inside the turbid cloudy sky, and into the gaseous Machine City, the domain of the Deus Ex Machina and his bug-like court of denizens. Each of these settings required the Tippett team of artists to populate environments with thin banks of gaseous sludge or mist, giving ambient life to the environment and good depth cues for scale.

2 Basic Technique

The shots were each populated with card-grids using a Mel script in Maya in order to mock what would normally be the steps in a ray-march for the calculus of a volumetric. The spacing between cards was a known quantity and was used as the distance between faux march steps. Populating each march step with a piece of geometry was important to Renderman since it is very good at motion blurring geometry whereas if a raymarcher were used the motion blur would have to have been calculated by the shader handling the march: a bookkeeping task better left to the time-tested renderer. This represented the volume of gas across which the view for that step needed to be integrated.

Since the cards were NURBS surfaces they had a built-in parametricity which could be exploited by the Renderman shader in order to determine the contours of the gases and its gross motion. A combination of classic fBm and turbulence patterns were used to construct the gross features in the cloudy volume but the majority of the motion came from an fBm (fractional Brownian motion) warp of each octave of density individually, allowing noise features at different scales to separate and rejoin over time and giving the impression of motion complexity. Each fractal setting (the number of octaves, the roughness, etc.) had a pair of settings which were blended between the top of the volume and the bottom so that the gasses could appear denser at their bases and wispier at their top edges. Also vortices could be positioned to "swirl" the texture coordinates when disturbed by passing objects.

3 Lighting

A few counterintuitive observations about thin gas photography added the finishing touches to the gases. The first is that due to low light film and vision sensitivity, shadowed gases appear more transparent than strongly lit gases. So gas opacity was shaded relative to how strongly the gas was lit. This point lended itself well to the contrasty and backlit world of the Wachowski brothers' vision since generally naively composited raymarch steps tend to "milk out" and neutralize the available light very quickly, even at shallow depths.

A normal was evaluated in the style of classic hypertextures and a custom illumination loop was built into the surface shader for the gases. Most the light was allowed to permeate and saturate the gas but there was a hint of pseudo-Lambertian diffusion such that the thinner the gas was being shaded, the more light was allowed to spill past the terminator toward the shadowed face.

4 Matte Holdouts and Object Intersections

The most difficult parts of the thin gas renders were not the generation of the visuals of the gases themselves but the integration with other objects. The shot complexity was so high that not all the geometry could fit into one Renderman pass and certainly couldn't be counted on for proper matte holdouts. Also a proximity solution had to be factored in that automatically softened the edges of gases that touched objects since said intersections tend to betray the missing proper object interaction.

Both problems were solved using a jittered sample solution against pre-composited Renderman z-depth imagery. But in order to antialias the results of the depth-holdout, instead of naively sampling together a filtered depth and pushing that through the rest of the occlusion or proximity code, the shader supersampled the end-results of the view occlusion and the edge softening.

The only differences, then, between view occlusion and edge softening were the range of the jitter region and how the results factored into the output alpha channel and Renderman opacity. The view occlusion results were only subject to the pixel-sized region around a depth sample and were treated like matte object holdout results, setting the opacity so that occluded gases weren't rendered. This invalidated the default Renderman alpha outputs so those were ignored in favor of multistreamed alpha channels controlled by the shader. This yielded the alpha channels for the gases. The edge softening regions used a distance user-settable in world units and didn't occlude anything. It was just used as a thinner for the gas densities.