

Jack's Frost: Controllable Magic Frost Simulations for 'Rise of the Guardians'

David Lipton
DreamWorks Animation
david.lipton@dreamworks.com

Ken Museth
DreamWorks Animation
ken.museth@dreamworks.com

Ben Sutherland
DreamWorks Animation
ben.sutherland@dreamworks.com

1. Introduction

In DreamWorks Animation's *Rise of the Guardians* Jack Frost has the ability to cover objects in swirling, magical frost. The frost is an extension of Jack's character and its performance and appearance a direct reflection of his mood. We needed a system that not only satisfied the visual criteria of looking magical and frosty, but was also controllable enough to both express the character's emotional state as well as hit hero shapes with specific timing. To achieve this, we built a Houdini based system that allowed for precise control over the frost's animation and behavior.

The frost system is based on an underlying Unstructured Cellular Automata simulation, which triggers points *ON* when a neighboring point, within a given search radius, is also *ON*. From this simulation, a velocity field is generated based on the point's density gradient at its trigger frame. This velocity field is then used to advect thousands of additional points; a trail is left behind each advected point, resulting in the effect's telltale lines.

2. Unstructured Cellular Automata Flow

Cellular Automata is a model for animating growth over time. We use a unstructured version that operates on a point cloud instead of a grid. In our system the artist paints the geometry in a manner that will represent how the simulation travels. Points are then scattered across the geometry, inheriting the painted color. We convert point color to neighbor search radius to define the flow speed; larger radii mean points find *ON* neighbors sooner and therefore drive flow faster in these areas (see Fig 2).



Figures 1 and 2. The frost is meant to be both magical and beautiful.

By painting areas green the artist can increase the speed with which the simulation travels. Blue slows areas down and red prevents points from ever turning on. Key starting points are set

to an on state and the simulation is run. Points colored green have a larger search radius for neighboring points and therefore are triggered on earlier, thus directing the flow through these areas.

3. Curve Generation

At each time step we generate a velocity field from the negative gradient of the triggered point density and store it in a compact VDB data structure [1]. Additionally, the boundary geometry, representing the surface on which the frost effect is propagating, is converted into a sparse narrow-band level set, also encoded into a VDB. Seed-points are then scattered on the geometry and advected in the velocity VDB using high-order Runge-Kutta time-integration schemes that are constrained by the closest-point-transformation derived directly from the level set VDB. This results in robust stream-lines that closely conform to the boundary geometry. Finally the stream-lines are rendered as curves and imported into NUKE as animated mattes for the final ice effects.



Figure 3. Advected points leave a trail defining the shape and flow.

4. Multiple Uses

The system was also used for situations not in the original spec. In the Tooth Fairy's lagoon the velocity field points were used to drive ice growth by converting the *OFF/ON* state to a greyscale image and spatially and temporally filtering the result. This was used as a displacement map and matte on ice below the water.



Figure 4. A flexible setup allowed for a myriad of uses.

References

[1] K. Museth, "VDB: High-Resolution Sparse Volumes With Dynamic Topology", ACM Transactions on Graphics, Volume 32, Issue 3, Pages 23, May 2013.