Is It Acid or Is It Fire?

How to Train Your Dragon: The Hidden World

Amaury Aubel
Dreamworks Animation, Glendale, CA, USA
Amaury.Aubel@dreamworks.com

KC Ong
Dreamworks Animation, Glendale, CA, USA
KC.Ong@dreamworks.com

Figure 1: Deathgrippers spitting acid.

ABSTRACT

The animated movie *How to Train Your Dragon: The Hidden World* introduces a new species of dragon in the franchise: the *Deathgripper*. This dragon possesses the ability to spit green acid that both dissolves and sets ablaze objects that it touches. In this talk we present the various challenges posed by this somewhat unique effect from the visual development phase to production shots.

CCS CONCEPTS

• **Computing methodologies** → Computer Graphics.

KEYWORDS

Visual effects, liquid simulation, fire simulation

ACM Reference Format:
Amaury Aubel and KC Ong. 2020. Is It Acid or Is It Fire?: How to Train Your Dragon: The Hidden World. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Talks (SIGGRAPH '20 Talks), August 17, 2020, Virtual Event, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3388767.3407356

SIGGRAPH '20 Talks, August 17, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-7971-7/20/08.
https://doi.org/10.1145/3388767.3407356

1 VISUAL DEVELOPMENT

The FX team was launched on the visual development of the acid fire in pre-production. From storyboards, it was clear that the effect was featured in fast-paced action sequences and that the sets quickly turned into burning infernos as a result. After some experimentation, we settled on a series of beats happening in rapid succession: stream of transparent liquid acid with an immediate release of steam on impact, erosion of material hit by liquid, and finally fire and smoke. We considered making the acid froth upon contact but realized there would not be enough time in shots for the viewer to notice.

While fictional, the world of *How to Train Your Dragon* is usually grounded in physical reality: effects follow the laws of physics. Consequently, the steam, fire and smoke are rendered with the hues one would expect. The green color of the acid (Fig. 1) was picked because it seemed to have an implicit association with "acidity" and reacted well to fire-lit scenarios.

2 SIMULATION

Early storyboards show *Deathgrippers* spitting acid at relatively distant targets. We therefore simulate the stream coming out of the dragon's mouth as high-velocity points that we emit into a regular FLIP simulation. We create artificial viscosity only in the neighborhood of collisions, so that the fluid sticks to colliders to some degree but retains an inviscid behavior elsewhere. To that effect, each point in the fluid simulation is assigned a smoothly

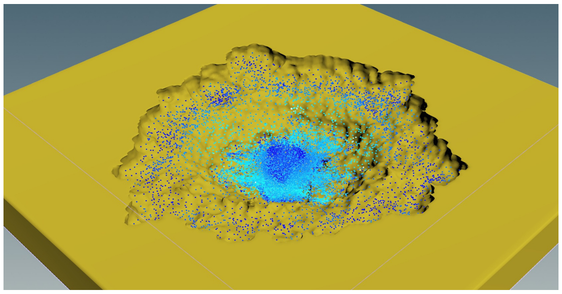

Figure 2: Erosion process: FLIP points (blue) and collider SDF (yellow).

varying *colliding* attribute based on proximity with colliders. A value of 0 indicates that the point is not in the vicinity of a collider and thus is moving freely. For any other value, a *stickiness force* is exerted onto the FLIP point to move it towards the collider. The magnitude of the *stickiness force* is proportional to the *colliding* attribute and its direction follows the gradient derived from the signed distance field (SDF) of the colliders.

The erosion process is simulated by carving holes into the colliders (Fig. 2). We first construct an erosion SDF by rasterizing the colliding points (i.e. *colliding* > 0) in the FLIP sim as spheres. We then subtract the erosion SDF from the SDF of the colliders. Note that prior to this boolean operation, the erosion SDF can be smoothed to minimize the creation of jagged edges.

We add an *acidity* attribute to the FLIP particles as a tool to control the erosion. Each colliding particle consumes a small amount of acidity at each time step. By enabling the erosion only when the acidity is within a certain range, we can delay the start of the erosion as well as decide when it stops. Although this erosion mechanism works well, it sometimes appears that matter simply disappears into thin air. To combat this, we release a subset of points originating from parts of the colliders that are eroded back into the FLIP simulation. In this way, a fraction of the eroded matter turns back into acid. This helps sell the illusion that things "liquefy" before getting dissolved.

We use the set of colliding points in the FLIP sim at each time step to emit into a secondary steam simulation. We analyze the set of colliding points and extract the border of the erosion process using heuristics such as the distance to the original, un-eroded surfaces. Steam emitted from this region will generally not get trapped inside the colliders or the acid. Similarly, we emit fuel and temperature into a fire simulation once the erosion has been occurring for a while. In practice, we paint the maximum possible coverage of the flames on the final frame of the eroded colliders. We then run a simple cellular automaton so that the emission propagates over time starting from hand-placed seeds that are activated based on time. Embellishments such as scorch marks and glowing hot burn patterns are added procedurally through noise-based shaders driven by temperature, surface curvature, and distance to the original, un-eroded surfaces.

3 ERODED ASSET RECONSTRUCTION

All colliding assets are combined into a single SDF before the erosion process. The erosion process removes bits from the SDF over time. To reconstruct a renderable version of the eroded assets, we loop through each asset and intersect the SDF derived from its original un-eroded polygonal representation with the output from the simulation. The resultant SDF is then converted back to polygons for rendering. By doing so, we ensure that UVs are transferred from the correct asset and not from neighboring ones.

As acid eats away at material, internal parts of existing assets (lacking proper surfacing) are inevitably revealed. A simple UV transfer based on proximity worked well in our case: wood, stone and metal were the only materials hit by acid and their surfacing held up relatively well to UV distortion. Additionally, our lighting scenarios generally had a forgiving high dynamic range: eroded material would look either fairly dark or almost blown out.

4 PRODUCTION SHOTS

Production shots presented additional challenges: *Deathgrippers* were spitting acid at moving props like a falling beam and a crumbling suspended metal staircase (Fig 1). In the simple case of the falling beam, we calculated frame-by-frame transforms and applied them to the beam collider. For the complex case of the crumbling staircase, we first ran a rough pass of the acid spitting (without erosion) to determine the regions of the staircase to be eroded. We then ran a rigid-body dynamics (RBD) sim to break and collapse the staircase in those regions. Finally, we refined and re-ran the acid pass with erosion enabled on the RBD pieces. This layered approach gave us more control over each step.

We used the *acidity* attribute to stop unwanted erosion in order to sculpt the final eroded shape. This was achieved by setting *acidity* to 0 when within close proximity of areas we wanted excluded from the erosion. To simulate different materials eroding at different rates, we set the *acidity* value differently when the FLIP points are near wood vs. stone. For shots that required eroded surfaces for continuity, we set the *acidity* value such that the erosion happens right away on contact during pre-roll.

To simulate the metal grates melting, we took the released points originating from the grates during erosion and rasterized them as a density volume with incandescence ramping from acid green to black to very bright yellow as they age. This gives the illusion that the temperature of the melting metal increases over time.

Occasionally we found dangling mesh islands that were remnants of the erosion process. These were cleaned up post-sim by removing island manifolds of small sizes. To have better control, we often ran separate erosion simulations in different parts of the set. In one shot, we had three separate erosion sims and their results were combined when reconstructing the eroded floor.

Last, we created a library of falling debris with trailing fire and instanced them as needed. Flying embers that were advected by the fire sims and characters added another dimension of realism.

5 CONCLUSION

Our effect adds a new twist to the traditional fire-breathing dragon and empowers the *Deathgrippers* as a menacing species. It is featured in 20 final shots across three sequences. While the erosion is often covered by steam, fire and smoke, we believe the effect successfully conveys the sense of immediate danger and threat.