InstaFalls:

How To Train Your Waterfalls

Baptiste Van Opstal
DreamWorks Animation
Baptiste.Van.Opstal@dreamworks.com

Youxi Woo
DreamWorks Animation
Youxi.Woo@dreamworks.com

Amaury Aubel
DreamWorks Animation
Amaury.Aubel@dreamworks.com

Figure 1: Shots using the *InstaFalls* system. Left: Caldera. Right: Hidden World.

ABSTRACT

The environments of *How To Train Your Dragon: The Hidden World* are immersed in waterfalls, from lush Nordic islands to mysterious, submerged kingdoms of lakes and crystals. Because of the variety and complexity of these sets (about 4000 waterfalls had to be created), a traditional approach would have consumed too many artistic and rendering resources. We decided to develop a collection of tools named *InstaFalls* with a few goals in mind: streamlining the creative process by minimizing the time spent on simulations, iterating faster thanks to real-time feedback, and managing the large quantity of generated data. Using this system, artists were able to create all the water elements of an environment set, from misty calderas to foamy, aerated ponds.

CCS CONCEPTS

• **Computing methodologies → Procedural animation**.

KEYWORDS

Water effects, distributed processing

ACM Reference format:

 https://doi.org/10.1145/3306307.3328156

1 WATERFALLS SIMULATION

Our previous approach to simulate large-scale waterfalls, called *Flow Field* [Yuksel et al. 2014], relied on a particle simulation with additional forces based on the local distribution of neighbors. While this produced characteristic liquid shapes, it required a number of artistic iterations to achieve the right look. Our new technique builds upon this concept while minimizing the fine-tuning of parameters.

One defining visual characteristic of waterfalls is the peeling effect caused by air friction on the column of water. Droplets on the fringes tend to separate from the column, slow down and eventually vaporize. We achieved this look by using a particle simulation and a number of *OpenVDB* operations such as selectively removing divergence from the velocity field. Thanks to the sparse nature of *OpenVDB*, we retained high performance and low memory usage. Particles that have been isolated for some time are removed from the particles stream and used as emitters for a separate mist simulation.

We ultimately created twelve 1000-frame long simulations for instancing using this approach. The hero waterfalls that made up the Caldera (Fig. 1) were also simulated in this way. They were easily injected into hero FLIP sims in regions where the waterfalls pooled into overflowing basins. Finally, the same technique was used to create whitewater effects on top of FLIP simulations.

2 INSTANCING

At its core, *InstaFalls* is an instancing system. However, because it simulates physics on proxy geometry, instances could automatically be adapted to the terrain in real-time. It enabled the user to concentrate on artistic aspects such as composition and timing while always producing physically-plausible results. In the end, an experienced artist could populate a set containing a network of hundreds of waterfalls and ponds in a day.

2.1 Instancing + Physics + Deformer

The input to the instancing system is a library of cached simulations in a neutral state, i.e. without any collision. When the user paints a stroke on the set, a small number of guide curves are automatically emitted from that stroke. A custom multi-threaded physics solver is run on these guide curves and produces a corresponding set of target curves. In practice, we model gravity and collisions with the set including friction, bounce and

SIGGRAPH '19 Talks, July 28 - August 01, 2019, Los Angeles, CA, USA
ACM 978-1-4503-6317-4/19/07.
10.1145/3306307.3328156
https://doi.org/10.1145/3306307.3328156

stickiness. Because the physics solver always runs in the background, edits to the terrain geometry immediately result in a new set of target curves. These edits could also be used as a tool to shape the waterfalls. Finally, a deformer based on the guide and target curves computes the final position of the points or voxels of the cached simulation. As the deformer can dynamically redistribute its weights, it allows for forking even though the cached simulation does not branch (Fig. 2).

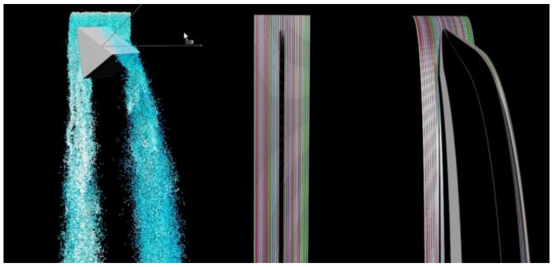

Figure 2: Real-time Adaptive Instancing

2.2 Production Tools

Interactive sessions with creative leadership were paramount to art-direct the vast sets. Tools were all based on painting and real-time feedback. In addition to water sources, boundaries and flow direction, physical attributes could be painted on in space; speed and direction changes could be tuned at any point. The guide curves provided a natural parameterization and modifications to all properties could be easily expressed in that parametric space.

Approved looks could be saved into presets and immediately applied across all instances at the shot or sequence level. The water or volumes were additionally clustered for either art direction or rendering convenience.

The system included an integrated foam and ripple solver. It could detect waterfalls that sourced or landed in proximity to a water surface and inject aeration, ripples and foam into the simulation. Similarly, other parts of the *InstaFalls* package also contained strongly coupled features that could adapt to nearby elements, such as the waterfall guides that aligned to pond surfaces and generated geometry to smooth the transition between water surface and waterfall.

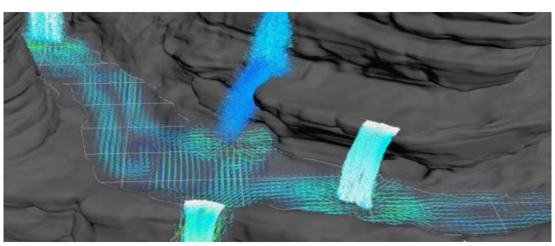

Figure 3: Integration with Sims

2.3 Lighting

The reflection and refraction of the surrounding environment is a defining element of the look of water. With wide panning cameras, it was important to collect the correct color information from all sources - 3d assets, 2d plates, matte painting.

The system automated a shader-based process to describe the environment from all input elements. By organizing the 3D scene assets into a material lookup chart and building a hierarchy of best-fit color choices, it allowed for accurate reflection and refraction reads from all angles, including back sides. Adding to the flexibility of the system, all compositing modifications were picked up, allowing environment look development and lighting to be refined in parallel.

3 DISTRIBUTED POST-PROCESSING

The complexity of our water sets also meant we had huge amounts of data to post process. This included analyzing the data for secondary simulations and building level sets for rendering. While these kinds of processes typically do not need a lot of processing power, they could easily consume hundreds of gigabytes of memory. To optimize our render resources, the data was divided into regions and then distributed across multiple machines with limited memory capacity.

Distribution is not new and having tried a number of existing techniques, the same limitations became apparent: the number of created regions was inconsistent and unpredictable throughout the frame range of a shot. Submitting to a predefined number of machines became hard or even impossible. In addition, because the data came from multiple simulations with different resolutions or simply on account of the shape of the simulation, the amount of data between the regions was not balanced evenly; some machines ended up overburdened while some others were practically idle.

To circumvent these two key issues, we created an algorithm that was highly customizable, could be adapted to any shape and evenly divided the data into a predictable number of slices. It allowed us to process successfully an average of 25 Tb of data per shot and work with level sets with as many as 10 billion voxels, while using only smaller machines on our render farm (4-8 procs per job, 6 Gb RAM per job on average).

4 CONCLUSION

Some set-dressing systems incorporate physics such as RBDs to achieve accurate and natural contact of instances (e.g. *Sprinkles*, our proprietary set-dressing tool). The tight coupling of a physics engine and instancing in *InstaFalls* took that idea a step further. It provided us with a way to quickly populate water sets in an artist-friendly way. For a single sequence, The Hidden World, over 3000 waterfalls were instanced from only 12 pre-simmed caches. For the Caldera sequence, we developed new tools to process huge data sets.

One improvement in the works is to switch from baking instances to disk to a render-time procedural for the waterfalls. This would reduce disk space usage as only the strokes and presets would have to be saved.

While developed primarily for water, some of our physically-based instancing and partitioning techniques could be applied to a wider variety of effects, e.g. instancing foliage elements that could automatically avoid environment collisions.

REFERENCES

C. Yuksel, K. Maxwell, S. Peterson. 2014. Shaping Particle Simulations with Interaction Forces. In *ACM SIGGRAPH 2014 Talks (SIGGRAPH '14)*. ACM New York, NY, USA, Article 42, 2 pages. https://doi.org/10.1145/2614106.2614121