Other Worldly Crowds in Coco

Jane Yen
Stephen Gustafson
Pixar Animation Studios

Aaron Lo
Pixar Animation Studios

J.D. Northrup
Lana Sun
Pixar Animation Studios

ABSTRACT

Coco, Pixar's largest human-based crowds film to date, was ambitious both visually and technically. Over a third of the film contains crowd scenes, ranging from a mansion-filled dance party to thousands of skeleton families journeying across a bridge, to a colossal cheering stadium. This complexity required vast amounts of both animation specificity and look variation in our characters.

Asset management, animation directability, and rendering would have been extremely difficult with our previous pipeline for human crowds at this scale. An array of techniques were developed to tackle these challenges, including crowd asset and workflow improvements; a new skeletal rigging and posing system to procedurally control animation; more automated, aggressive shading and geometric level of detail; and optimized geometry unrolling in Katana to significantly reduce scene processing time and file IO.

CCS CONCEPTS

• **Computing methodologies → Procedural Animation**;

KEYWORDS

crowds, pipeline, procedural animation, rendering

ACM Reference Format:
Jane Yen, Stephen Gustafson, Aaron Lo, J.D. Northrup, and Lana Sun. 2018. Other Worldly Crowds in Coco. In *Proceedings of SIGGRAPH '18 Talks.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3214745.3214796

1 INTRODUCTION

Traditionally for crowds, Pixar has used a sequenced geometry cache pipeline to best preserve animation fidelity. This allows fast iteration and is easily interchangeable with a full animation rig when additional specificity is needed. MURE [Gustafson et al. 2016] is then used for agent-based simulation to apply flocking motion. This pipeline integrates nicely with Pixar's proprietary animation suite, Presto, and allows for smooth interaction with other departments like layout and animation.

This technique has been successful for several films, however, without scale improvements for *Coco* our cache system would quickly become unwieldy. Even more significantly, geometry caches severely limit directability. All animation transitions must be explicitly pre-baked and common agent transformations, like head angle and eyeline, require promotion to full hero rigs. When previewing *Coco*'s story reels, it was clear that many sequences would be problematic with only the existing geometry cache pipeline.

Utilizing a two-pronged approach, we developed a new procedural, skeletal-based system for crowds requiring complex directability, and optimized the traditional pipeline for simpler crowds.

2 ASSET MANAGEMENT

Some 35,000 crowd caches were generated to achieve the necessary look variation and animation actions for *Coco*, requiring careful management and build improvements. Procedural mirroring, geometry trimming, key reduction, rigid transform, decimation, and mesh combining reduced cache data down to 34T with a 35 hour bake time - roughly a 2.5x improvement.

Needing nearly 70 action types, ultimately more than 1,500 unique animation clips were created for 12 retargeting body types. Traditionally, our crowd clips have been short in length (~100 frames) and cyclic, reducing animation overhead and cache sizes. The use of motion capture allowed for longer clips (~400 frames) which we found to be more organic and less repetitive across shots, especially as the start frames of longer clips can be varied in shorter shots. Furthermore, because of this increased diversity, fewer clips were needed per body type as compared to previous films.

A vital tool in managing and visually accessing our clip inventory was the use of automated videos of the retargeted animation clips. We could previsualize and iterate with Crowds Animation to see the results across our entire character pool before populating sequences, saving us significant time rebuilding clips or fixing animation in shots. We also found these videos extremely helpful in determining if we had sufficient clip coverage.

SIGGRAPH '18 Talks, August 12-16, 2018, Vancouver, BC, Canada

ACM ISBN 978-1-4503-5820-0/18/08.
https://doi.org/10.1145/3214745.3214796

3 IMAGE-BASED WORKFLOWS

Working with crowds in our standard production tools can be cumbersome because of the time it takes to open heavy shots, load and find models. As we already iterate by evaluating renders, we realized we could take advantage of existing image IDs to select models and perform common operations that did not require a 3D viewport. We created a plugin for Tweak Software's RV to perform deactivations, promotions to full rigs, and retrieval of crowd asset information; reducing user time from 20 minutes or more to a few minutes. These batch operations could be applied using screen space metrics or manual user selection. The biggest win, though, was creating a fast process that enabled the Simulation department to handle crowds independently. This workflow greatly reduced our time assisting with promotion. Further, we provided visualization features highlighting simulated characters versus non-simulated cloth-warped characters, which also proved crucial in helping them manage large crowd shots.

4 PROCEDURAL PIPELINE

The stadium sequences feature crowds of 20k skeletons, and called for timed transitions, look-ats, and many different styles of animations. It would have been prohibitively expensive to handle such sequences with our geometry caching pipeline, both from a clip-generation and farm IO perspective. For these reasons and the need to extend our directability, we developed simple, FK skeletal rigs for all of our characters. These rigs were encoded in Universal Scene Description (USD) for interchange, using a precursor to the current open-source 'UsdSkel' schemas. As an extension to the MURE crowd system, we developed a portable, procedural rigging and manipulation system, inspired by *Zootopia* [El-Ali et al. 2016]. Like the skeletons, cloth was deformed at render time using linear blend skinning, so re-simulating was not required. For hair, we deployed rigid deformations in the *Coco* time frame, but have since expanded joints to deform hair and fur at render time.

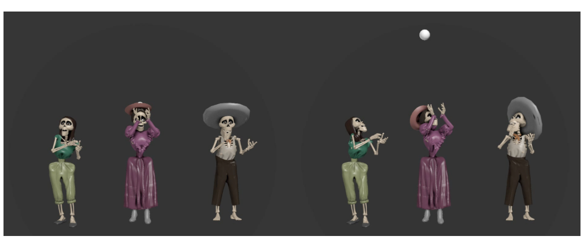

Figure 1: Original animation (left) is procedurally modified to control the look-at direction (right). ©Disney/Pixar

This new system was designed for multi-package integration, the primary target being a custom 'rig prim' type within SideFX's Houdini. With rig prims, the common pattern of fully-featured 'super rigs' is spurned in favor of minimalistic, additive rigging, catered to the needs of each crowd. Rig behavior is defined using sets of nodes inside of a custom node context. Compiled into an in-memory procedural, this node graph is bound to each primitive – either for immediate or deferred execution. The procedural is influenced using both generic attributes on the primitive and exported rigging controls, taking the form of additional primitive vertices.

4.1 Clip Blending

One of the primary uses for rig prims was animation blending, which allowed us to attain the myriads of animation variety and transitions between actions (e.g. watch to cheer, sit to stand) needed in the stadium sequences. In our performance capture pipeline, it was easiest for artists to produce lengthy animations. To maximize variation, these long animations were dynamically divided into chunks of about 4 seconds each, which were then sequenced at random, with cross-fade blending between clips. All clips were annotated, so that clip sequencing would tend to pick similar clips and avoid overly fidgety crowds – an illusion given by characters too frequently altering their primary pose, such as transitioning from arms folded in front to arms at the side.

To achieve precise reaction timing required the ability to smoothly transition into any clip at any time. This was done using a multi-layer, stack-based sequencing algorithm in which clips lower on the stack are subsumed by clips higher on the stack, but only once playback of the clip higher on the stack moves beyond its input blending window. This approach was both trivial to implement, and also made it possible to begin smoothly blending into a target animation at any time – even if a character was already in the middle of blending between other clips. Although there was no limit, 3- or 4-way animation blending was most common.

5 RENDERING OPTIMIZATION

With our traditional methods of particle-system data driving rendered instances, scene processing of our largest 20k-character shots took upwards of 2 hours per render and 90+ GB of render memory for crowd models. We found that combining common LOD techniques, along with a data-driven optimization effort that steered us to some counter-intuitive decisions, brought this down to around 20 minutes and less than 20GB for even the largest shots.

The Foundry's Katana proved to be a useful testbed for reconfiguring our crowds rendering pipeline. One successful technique was "nested" unrolling of our skeletally-posed crowds. For each of around 200 character body types, we "cooked" a single hierarchical representation in Katana; underneath each leaf-level primitive mesh we created children that were copies with skinning deformations applied as a render-time Katana Op. This meant the IO and scene processing overhead of reading character meshes, material definitions, etc. was done once, and then efficiently shared by potentially hundreds or thousands of discrete deformed copies.

This "nested" order also implicitly sorted geometry by material binding, making the handshake with RenderMan even more efficient. Without this form, renders of 20k characters read upwards of 10 GB of cache data per render host. With the nested version, the scene cooked about 40% faster, and reduced IO by 10x versus our traditional instancing pipeline, easing storage bottlenecks.

REFERENCES

Moe El-Ali, Le Tong, Josh Richards, Tuan Nguyen, Alberto Luceno Ros, and Norman Moses Joseph. 2016. Zootopia Crowd Pipeline. In *ACM SIGGRAPH 2016 Talks (SIGGRAPH '16)*. ACM, New York, NY, USA, Article 59, 2 pages. https://doi.org/10.1145/2897839.2927467

Stephen Gustafson, Hemagiri Arumugam, Paul Kanyuk, and Michael Lorenzen. 2016. MURE: Fast Agent Based Crowd Simulation for VFX and Animation. In *ACM SIGGRAPH 2016 Talks (SIGGRAPH '16)*. ACM, New York, NY, USA, Article 56, 2 pages. https://doi.org/10.1145/2897839.2927396