1 Unified Crowd Data Format

We implemented a simple custom data format not unlike Maya’s PDC particle data container. Specifically, we store per-frame data about root transforms, body rotations, wheel steering and roll angles, brake light intensity, and optionally a pointer to a desired animation clip to layer on top. A header stores a list of agents in each simulation file, their character variants, the superset of all animation clips requested in the simulation, etc.

We integrated this format with Massive, Houdini, and Marionette – our in-house animation system – with a series of plug-ins and scripts.

2 Constraint-Based Posing

One of the more unusual aspects of our pipeline was that we struck a balance between pre-posing animation into caches versus storing “live” animation for the entire crowd. In traditional instancer-based crowds, all animation clips are cached into per-frame geometry, and then sequenced at appropriate locations for the crowds. By contrast, with dynamic pose engines (like Massive’s standard rendering plug-ins), per-frame joint angles are simulated and then posed for each crowd character on each frame of the final shots. This can be useful for simulations needing IK goal modifications or pose-blending, but in the case of playing back discrete keyframe-animated clips it can be inefficient.

Our system stores animation data on a small pool of off-screen characters called “actors” (typically 10 to 60), which is accessed via a static, in-memory array of pose information by the larger crowd of “agents”. We have two modes for this: one where each of these actors represent a single animation clip, and another where each actor has an entire sequence of animation applied via a finite state machine system. For the former case, simulated crowd agents pick which animation clip to play back for a given frame, and look up the pose information from the appropriate actor. This mode was useful for tightly-choreographed shots where agents chose their acting based on events in a simulation.

In the second mode, crowd agents randomly pick a given actor to follow, and mimic their pose information, with a randomized frame offset. This was most useful for more ambient crowds, where we wanted to easily try out a variety of keep-alive animation without much specific choreography.

3 Agent Behavior

We also implemented a collection of higher-order operators on top of Massive’s fuzzy nodes, including signal decay functions to serve as “memory”, accumulators/deaccumulators for integrating acceleration signals into velocity signals, temporal hysteresis to help with commitment to actions, and flip-flops to implement simple stateful operations in the relatively stateless fuzzy environment. We also used a signal processing-based system developed during Wall-E to approximate damped harmonic motion.

Taking a cue from gradient-domain image-processing techniques, we found blending control systems operating in the velocity and acceleration domains provided more robust set of behaviors than using either domain alone. We found it elegant that this algorithmic approach helps with motion hitches in much the same way as it allows for minimizing discontinuities in an image gradient.

4 Render Metrics

Shrinkwraps significantly increased our ability to render crowds. This effect was most noticeable for large crowds or long distances, since cost falls with pixel coverage of each model. For a test shot with a basic lighting setup:

<table>
<thead>
<tr>
<th>models</th>
<th>rib gen (h:mm:ss)</th>
<th>rib gen memory</th>
<th>prman (h:mm:ss)</th>
<th>prman memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>(close) 200 real</td>
<td>0:29:39</td>
<td>1.0 GB</td>
<td>1:45:22</td>
<td>1.9 GB</td>
</tr>
<tr>
<td>(close) 200 shrinkwrap</td>
<td>0:00:31</td>
<td>0.15 GB</td>
<td>0:37:48</td>
<td>0.6 GB</td>
</tr>
<tr>
<td>(long) 200 real</td>
<td>0:31:10</td>
<td>1.0 GB</td>
<td>0:59:24</td>
<td>3.8 GB</td>
</tr>
<tr>
<td>(long) 200 shrinkwrap</td>
<td>0:00:37</td>
<td>0.15 GB</td>
<td>0:00:51</td>
<td>61 MB</td>
</tr>
</tbody>
</table>
5 Limitations

The use of shrinkwraps was not ideal for models with concavities or transparency. Side mirrors of many cars were the main areas we noticed visible artifacts. Heavy displacement can also create render-time artifacts from lighting and texture filtering, but for background characters we found these acceptable. We also avoided some forms of ray-tracing of crowds – in particular, self-reflection. Our displacement-shaded mouths weren’t shaped the same as our full mouths because of the way we parameterized the front end of the various car types.

Because we essentially collapsed the entire model hierarchy down to a box for the body, and four cylinders for the wheels, per-gprim lighting was not possible. In addition, as is the case with any type of LOD scheme, maintenance and preparation of our crowd models entailed some overhead for the characters department. Finally, animation was limited to rigid transforms of the body and wheels, along with facial details.

6 Intended Audience

This talk is targeted to anyone interested in implementing or using a production crowds pipeline, especially for film work.