MOMENT TRANSPARENCY

BRIAN SHARPE, SENIOR SOFTWARE ENGINEER
WETA DIGITAL
Gazebo

• Gazebo is Weta Digital’s general-purpose real-time renderer
  • OpenGL rasterizer
  • Consumes movie data directly without change.
• Data is often in non-ideal format for real-time rendering
  • Large data (lots of instances, lots of triangles)
  • Arbitrary transparency
Hardware Constraints at Weta Digital

- A range of GPU’s at Weta Digital’s
- A lot of Kepler’s (Desktop and RenderWall)
- Cannot use frame-interlock (raster order views)
Transparency Algorithms

• Hybrid Transparency [Maule et al. 2013]
  • Issues when camera inside particle system

• Weighted-Blended Order-Independent Transparency [McGuire and Bavoli 2013]
  • Overly transparency look

• Fourier Opacity Map Transparency [Jensen and Bavoli 2010]
  • Ringing issues
  • Precision in Z overly coarse (close objects blend together incorrectly)
Moments for OIT

- Decided to extend/improve “Weighted-Blended Order-Independent Transparency” with a per-pixel moments-based transmittance function.

- Déjà vu?
  - Work developed concurrently to “Moments-Based Order-Independent Transparency” [Münstermann et al. 2018 I3D]
  - From the same group that pioneered the use of moments for shadow mapping.
  - Go read their work, its awesome! :-)
Transmittance Function for OIT

- Adaptive Transparency [Salvi et al. 2011] showed that if a per-pixel transmittance function could be precomputed, then order-independent transparency (OIT) could be achieved in a two-pass algorithm.

- Precomputing this transmittance function is non-trivial.

- Weighted-Blended Order-Independent Transparency approximated it with a constant curve, which gave nice results, but in most cases overly transparent.
Moment Shadow Mapping

- Moment Shadow Mapping [Peters and Klein 2015]

- For each pixel in shadow map, store the corresponding 4 moments \((z, z^2, z^3, z^4)^T\) for a single depth \(z\)

- A filtered sample over a set of these pixels with corresponding weights \(w_i\) is

\[
b = \sum_{i=1}^{n} (z_i, z_i^2, z_i^3, z_i^4)^T \cdot w_i
\]

- Hamburger 4MSM algorithm used to calculate approximate fraction of depth values closer to light than given input depth \(z\)

- Moment shadow mapping uses this for filtered shadows
Moment-based transmittance function

- To adapt transmittance for use with moments and OIT, we move to log space, made easier using terminology from volumetric rendering.

- We can interpret a fragment's alpha \( \alpha \), as *optical depth* \( d \)

\[
d = -\log(1 - \alpha)
\]

- Optical depth is additive, meaning composition of multiple fragments is a sum

\[
d' = \sum_{i=1}^{n} d_i.
\]
Moment-based transmittance function

• With this in mind, we can use optical depth $d$ as the weight when creating a moments sample

$$b = \sum_{i=1}^{n} \left( z_i, z_i^2, z_i^3, z_i^4 \right)^\top \cdot d_i$$

• Noting that total optical depth $d'$ is needed for normalization, we now have a transmittance function stored as 5 floats (4 moments $b$ and 1 total optical depth $d'$)

• Reconstruction is via the Hamburger 4MSM algorithm

$$T(z) = \exp(-\text{Hamburger}(\frac{b}{d'}, z) \cdot d')$$
Moment-based transmittance function

• Results:
  • Transmittance function = black
  • Moment-based = red
  • Moment-based with overestimation = green

• Moments are a good match for 2 and 3 samples only.

• Overestimation [Peters et al. 2017] is an improvement (25% used in this case) although it overshadows in the front.
Moment Transparency

• Integrate into Weighted-Blended Order-Independent Transparency by replacing its empirically defined transmittance function, with new moments-based version.

• Requires an extra geometry pass
  • Allocate at screen resolution two new buffers
  • RGBA_F32 to capture the moments $b$
  • R_F32 to capture total optical depth $d'$
  • Initialize both to zero, and set blend mode to additive

• Replace $w(z,a)$ with moments-based version

\[
b = \sum_{i=1}^{n} (z_i^2, z_i^3, z_i^4)^T \cdot -\log(1 - \alpha_i)
\]

\[
d' = \sum_{i=1}^{n} -\log(1 - \alpha_i)
\]

\[
T(z) = \exp(-\text{Hamburger}(\frac{b}{d'}, z) \cdot d')
\]
Moment Transparency

- Large values in z can cause precision issues when accumulating moments.
- To fix this we apply the following logarithmic warp.

\[ z' = \frac{\log(z/Z_{\text{near}})}{\log(Z_{\text{far}}/Z_{\text{near}})} \]
Moment Transparency

Weighted-Blended
Order-Independent Transparency

Moment Transparency

Ground Truth
Moment Transparency

Alpha Blending  
Weighted-Blended Order-Independent Transparency  
Moment Transparency
MOMENT TRANSPARENCY: Improvements
Low Resolution Moments

- Moments can be filtered
- This allows decoupling of the moments buffer resolution from that of the screen (ie low resolution transmittance function)
- Increases performance and reduces memory
- Results:
  - Thin surfaces can have issues (some bleeding, flickering with animation)
  - Large objects almost no issues at all
  - More forgiving than you might think
  - Recommend a little bit (at least 1/2, at Weta we use 1/8)
Low Resolution Moments

Moment Transparency

Moment Transparency LowRes 1/8
Different Formats for Write & Read

• To improve performance, it is possible to use a different format for moments capture (ie write) than used for moments reconstruction (ie read)

• Non-Linearly Quantized Moments (NLQM) [Peters 2017]
  • A special quantized format for storing 4 moments at 16bit
  • No loss observable loss in quality
  • Faster to reconstruct than regular moments (less bandwidth, less arithmetic instructions)
  • Convert captured 32bit moments to NLQM before reconstruction
Two Moments Passes

• As seen, regular Hamburger can be a poor fit for a transmittance curve with large fragment counts.

• A large part of the problem, is that we’re giving equal weight to fragments in the back, as to those in the front, even though they’ll contribute less to the final pixel.

• This results in poor compression.

• Overestimation can help, but can also overshadow in the front.
Two Moments Passes

• Capture a second set of moments, while using the first set of moments as an estimate for this occlusion.

• Adds bias in that each fragment occluded by twice the amount it should be.

• In practice bias is offset by the loss of detail of moments reconstruction, which tends to under occlude.
Two Moments Passes

• Results:
  • Transmittance function = black
  • Moment-based = red
  • Moment-based with overestimation = green
  • Two moments passes = blue
Two Moments Passes

Moment Transparency

Two Moments Passes

Ground Truth
Two Moments Passes

Moment Transparency

Two Moments Passes
MOMENT TRANSPARENCY: Challenging Cases
Challenging Cases Remain

Overestimation 25%  Overestimation 17%  Overestimation 10%  Ground Truth
THANK YOU