Analytical Motion Blur Rasterization with Compression

Carl Johan Gribel\textsuperscript{1}
Michael Doggett\textsuperscript{1}
Tomas Akenine-Möller\textsuperscript{1,2}

\textsuperscript{1}Lund University
\textsuperscript{2}Intel Corporation
Motivation
Motivation

- Human visual system designed to detect motion
- This fails when presented too few images, or too much motion per image
  - motion gets jumpy
- Motion blur aids the motion detection of the visual system
Talk outline

• Edge Equations and exact exposure intervals
  - analytic inside-test
  - visibility management

• Computing the time integral
  - opaque/translucent resolve

• Compression
Previous work

• Accumulation buffer [HA90][WGER05][DWS88]
  - expensive
  - strobing artifacts

• Stochastic [CCC87] [AMMH07][McGuire et al. HPG2010]
  - correct solution but slow convergence

• Single sample analytical visibility [KB83]
  Extended with stochastic shading by Sung et al. [SPW02]
  - details missing for visibility computations
Edge Equation primer for static triangle

\[ p_0 \quad p_1 \quad p_2 \]

\[ (x_0, y_0) \]
Edge Equation primer for static triangle
Edge Equation primer for static triangle

3D homogeneous space: \[ e_1 = (p_2 \times p_0) \cdot (x_0, y_0, 1) \]
Edge Equation primer for static triangle

3D homogeneous space: \[ e_1 = (p_2 \times p_0) \cdot (x_0, y_0, 1) \]
Edge Equation primer for moving triangle
Edge Equation primer for moving triangle
Edge Equation primer for moving triangle

\[ p_0(t) = (1 - t)q_0 + tr_0 \]

\[ t \in [0, 1] \]
Edge Equations

\[ e_1(t) = (p_2(t) \times p_0(t)) \cdot (x_0, y_0, 1) \]
\[ = (((1 - t)q_2 + tr_2) \times ((1 - t)q_0 + tr_0)) \cdot (x_0, y_0, 1) \]
\[ = (ft^2 + gt + h) \cdot (x_0, y_0, 1) \]
We also derive the analytical depth function $d(t)$ ...
Rasterization

• For each pixel of bounding box, compute 2nd degree solutions
• Insert intervals into per-pixel list
söndag den 27 juni 2010
söndag den 27 juni 2010
Opaque resolve

- Sweep over time and keep track of overlapping intervals
- Pick colors of the closest intervals
- Blend together for final color
Opaque resolve

• Sweep over time and keep track of overlapping intervals
• Pick colors of the closest intervals
• Blend together for final color
Opaque resolve

- Sweep over time and keep track of overlapping intervals
- Pick colors of the closest intervals
- Blend together for final color
Translucent resolve

- Sweep over time and keep track of overlapping intervals

- **Alpha-blend each subsection**

- Blend together for final color

\[
\alpha = 0.6 \quad \alpha = 1.0
\]
Translucent resolve

- Sweep over time and keep track of overlapping intervals
- **Alpha-blend each subsection**
- Blend together for final color

\[
\alpha = 0.6 \quad \alpha = 1.0
\]
Translucent resolve

- Sweep over time and keep track of overlapping intervals
- **Alpha-blend each subsection**
- Blend together for final color

\[ \alpha = 0.6 \]
\[ \alpha = 1.0 \]
Compression

- Will end up with lots of intervals
- But, can expect spatial & temporal coherency
- Proposal: merge similar intervals
Compression: Oracle function

\[ O(\Delta_i, \Delta_j) \]

- Metric for interval similarity
- Merge pairs of intervals deemed most similar
\[ O(\Delta_i, \Delta_j) = h_1 \max(t_j^s - t_i^e, 0) + h_2|\bar{z}_i - z_j| + h_3|k_i - k_j| + h_4(t_i^e - t_i^s + t_j^e - t_j^s) + h_5(|c_i - c_j|) \]

temporal proximity
\[ O(\Delta_i, \Delta_j) = h_1 \max(t_j^s - t_i^e, 0) + h_2 |\bar{z}_i - z_j| + h_3 |k_i - k_j| + h_4 (t_i^e - t_i^s + t_j^e - t_j^s) + h_5 (|c_i - c_j|) \]
$$O(\Delta_i, \Delta_j) = h_1 \max(t^s_j - t^e_i, 0) + h_2 |\bar{z}_i - z_j| + h_3 |k_i - k_j| + h_4 (t^e_i - t^s_i + t^e_j - t^s_j) + h_5 (|c_i - c_j|)$$

temporal proximity
depths
slope
\[ O(\Delta_i, \Delta_j) = h_1 \max(t_j^s - t_i^e, 0) + h_2 |\bar{z}_i - z_j| + h_3 |k_i - k_j| + h_4(t_i^e - t_i^s + t_j^e - t_j^s) + h_5 (|c_i - c_j|) \]

temporal proximity
depths
slope
temporal coverage
\[ O(\Delta_i, \Delta_j) = h_1 \max(t_j^s - t_i^e, 0) + h_2 |\bar{z}_i - z_j| + h_3 |k_i - k_j| + h_4 (t_i^e - t_i^s + t_j^e - t_j^s) + h_5 (|c_i - c_j|) \]

temporal proximity
depths
slope
temporal coverage
color norm
\[ O(\Delta_i, \Delta_j) = h_1 \max(t_j^s - t_i^e, 0) + h_2|z_i - z_j| + h_3|k_i - k_j| + h_4(t_i^e - t_i^s + t_j^e - t_j^s) + h_5(|c_i - c_j|) \]

- temporal proximity
- depths
- slope
- temporal coverage
- color norm
- weights

söndag den 27 juni 2010
Interval merging

- Merge interval pair with lowest Oracle value
- Combine temporal coverage
- Blend attributes
Compression with occlusion

39 intervals

8 intervals

77 intervals

8 intervals
Results
Compression: 8
Compression Rates
Compression: 64
Algorithm summary

- Time-dependent edge equations to compute exact exposure intervals
- Visibility management by using a linear approximation of a cubic rational depth function
- Blur translucent geometry by performing alpha-blending extended in time
- We propose a compression algorithm to address fixed memory requirements
Future work

- GPU implementation (in progress)
- Motion blurred shadows (in progress)
- Apply decoupled shading
- Analytical spatial anti-aliasing
- Higher order motion
Danke !
Danke!

* No bunnies were harmed during the research
Compression failure case

Fast moving fence

Analytical 16 intervals

24 spp

256 spp
Depth approximation error

Relative depth error histogram