Transparency unit test

Alpha blended transparency

This sample uses basic alpha blending. It sorts objects based on the distance from the camera and renders them back to front. It does the same for the particle systems and the particles. It doesn't sort the particle systems against the other objects as they're rendered using a different draw call.

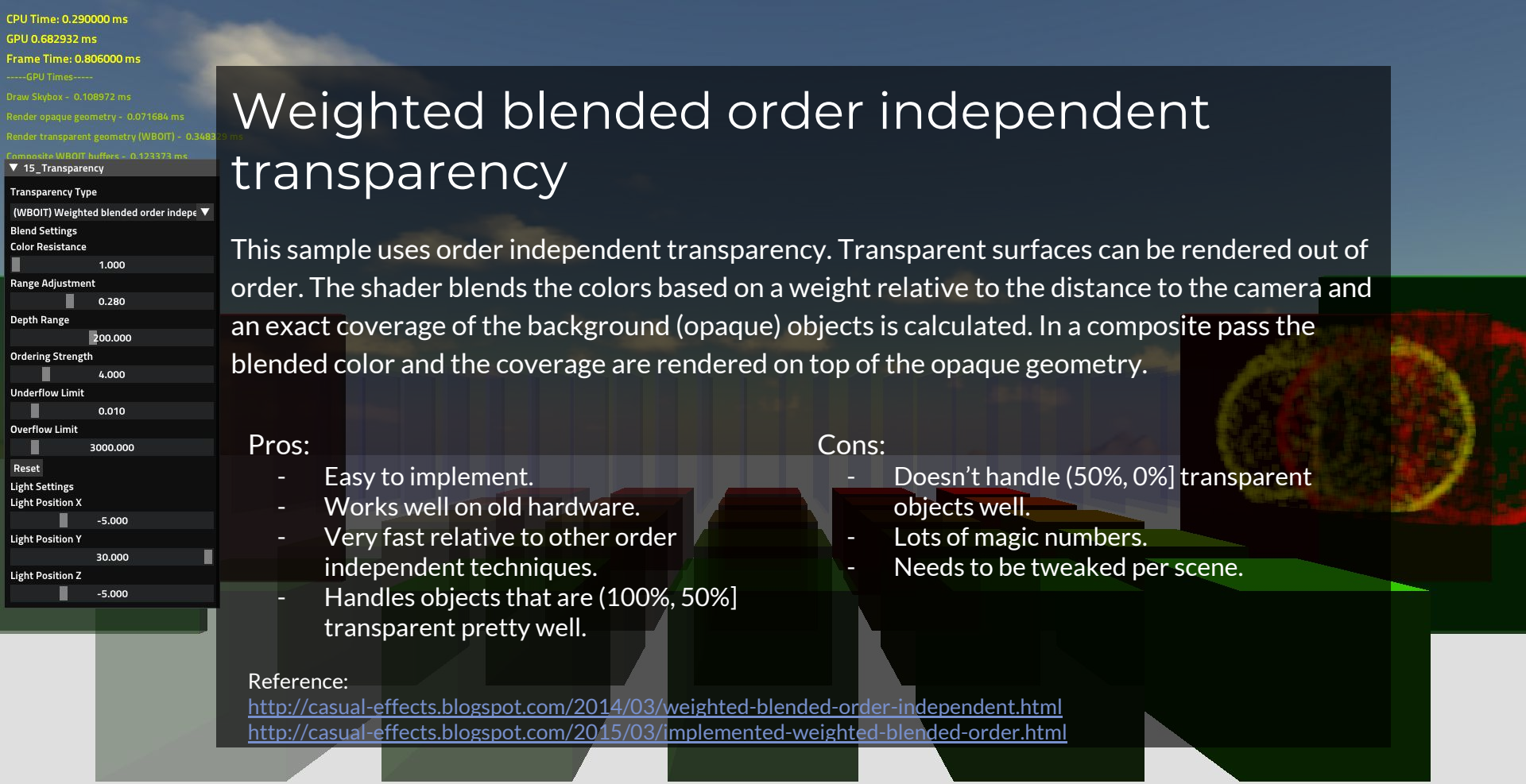
Pros:

- Cheapest of the techniques evaluated.
- Works on all hardware.

Cons:

- All transparent geometry needs to be sorted otherwise you get hideous artifacts (Particle systems)
- For best result needs to be sorted per triangle or even per pixel.
- Sorting increases CPU usage.

Reference:
http://www.opengl-tutorial.org/intermediate-tutorials/tutorial-10-transparency/



Weighted blended order independent transparency

This sample uses order independent transparency. Transparent surfaces can be rendered out of order. The shader blends the colors based on a weight relative to the distance to the camera and an exact coverage of the background (opaque) objects is calculated. In a composite pass the blended color and the coverage are rendered on top of the opaque geometry.

Pros:

- Easy to implement.
- Works well on old hardware.
- Very fast relative to other order independent techniques.
- Handles objects that are (100%, 50%] transparent pretty well.

Cons:

- Doesn't handle (50%, 0%] transparent objects well.
- Lots of magic numbers.
- Needs to be tweaked per scene.

Reference:
http://casual-effects.blogspot.com/2014/03/weighted-blended-order-independent.html
http://casual-effects.blogspot.com/2015/03/implemented-weighted-blended-order.html

Weighted blended order independent transparency by Volition

This sample uses weighted, blended, order independent transparency with the changes proposed by Volition in their GDC 2018 presentation. Blending of emissive transparent objects has been improved by adding a render target for the additive intensity. In the original technique emissive object lose their intensity relative to their transparency. The also changed to weight function to reduce the chance at under/overflow of the framebuffers at FP16 per pixel.

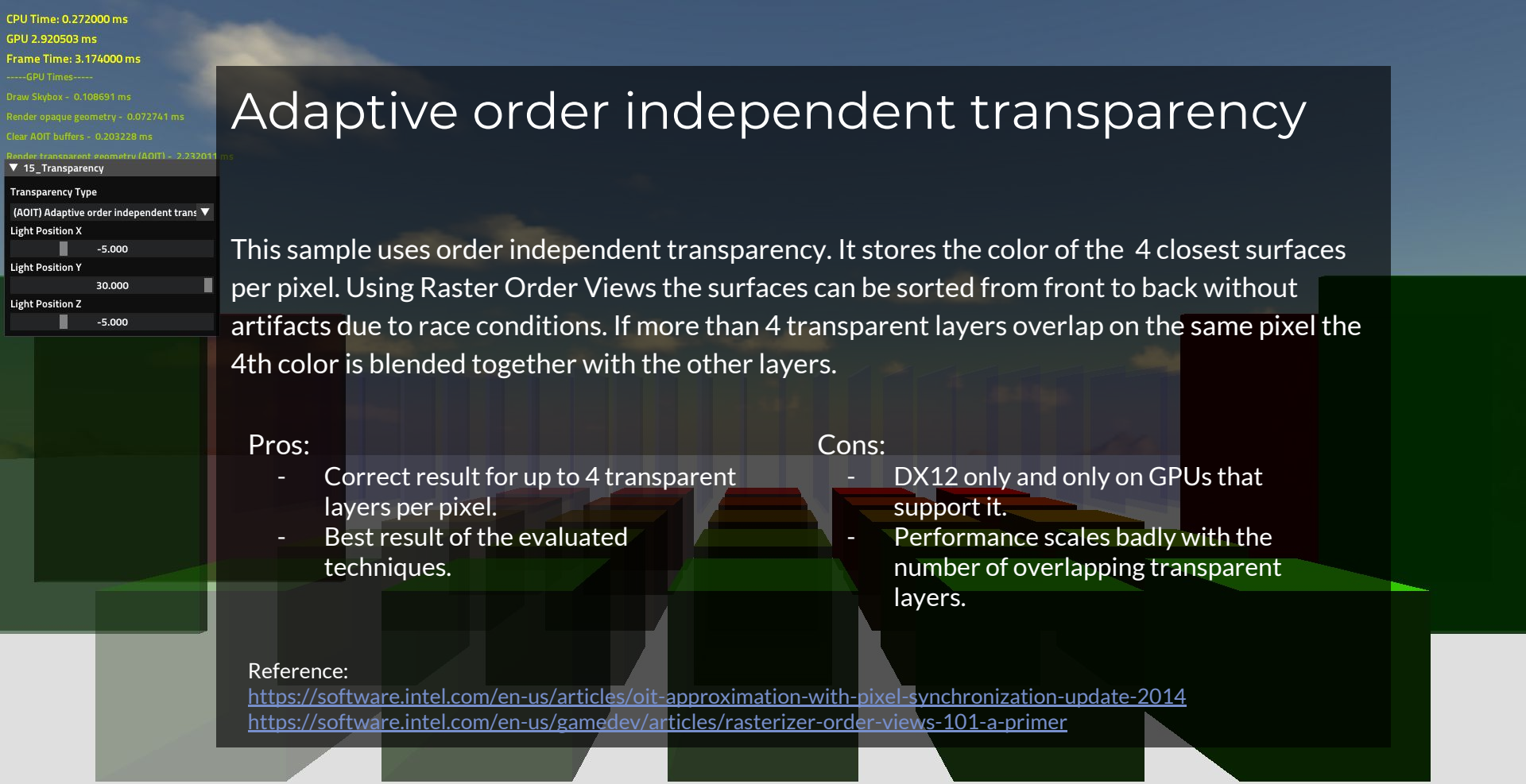
Pros:

- Easy to implement.
- Works well on old hardware.
- Very fast relative to other order independent techniques.
- Handles objects that are (100%, 50%] transparent pretty well.

Cons:

- Doesn't handle (50%, 0%] transparent objects well.
- Lots of magic numbers.
- In our sample it looks worse than the original technique.
- Does not maintain correct color value for a single layer of transparency.

Reference:
https://www.gdcvault.com/play/1025400/Rendering-Technology-in-Agents-of



Adaptive order independent transparency

This sample uses order independent transparency. It stores the color of the 4 closest surfaces per pixel. Using Raster Order Views the surfaces can be sorted from front to back without artifacts due to race conditions. If more than 4 transparent layers overlap on the same pixel the 4th color is blended together with the other layers.

Pros:

- Correct result for up to 4 transparent layers per pixel.
- Best result of the evaluated techniques.

Cons:

- DX12 only and only on GPUs that support it.
- Performance scales badly with the number of overlapping transparent layers.

Reference:

https://software.intel.com/en-us/articles/oit-approximation-with-pixel-synchronization-update-2014

https://software.intel.com/en-us/gamedev/articles/rasterizer-order-views-101-a-primer

What’s next?

Phenomenological Transparency: Successor to Weighted blended order independent transparency. Adds diffusion, refraction and shadows for transparent objects.


Moment based transparency: Based on moment shadow mapping. Real time performance for many layers of transparent objects with high quality results.

Reference:
https://casual-effects.com/research/McGuire2017Transparency/McGuire2017Transparency.pdf
http://cg.cs.uni-bonn.de/en/publications/paper-details/muenstermann2018-mboit/