Summary on *Hybrid Transparency*

Victor Shu

The paper mainly focuses on an approach for real-time approximation of OIT (order-independent transparency) called Hybrid Transparency to provide a high quality on rendered image without visible artifacts. There are several approaches to this, but some of them requires a huge chunk of memory (A-buffer), or they are not available on the GPUs at the time (adaptive transparency), or they introduce artifacts (stochastic transparency). Other approaches like weighted sums technology makes the pixel color saturates very quickly and weighted averages (WAvg) technique generates poor color approximation. The approach of this paper uses an accurate method to evaluate the visibility of the front-most fragments but uses a less accurate one for the other fragments.

The HT approach splits the fragments into two parts: the core and the tail. To extract *k* core fragments, the paper uses a truncated A-buffer, where *k* is determined based on memory budget or the scene. The tail, on the other hand, must work on limited memory but provides image without overflowing. The paper uses WAvg, which works in a single geometry pass, to get a better color approximation. After that, a final color will be calculated with the core color, the tail color and the background color. The paper also managed to implement the method on GPU with two geometry passes which collects the *k* cores and store the pixel visibility function in a full-screen quad and then blend the cores and tails.

The paper compares the HT approach with the reference method, A-buffer, which produces images with almost no noticeable artifacts, along with other well-known techniques, in terms of memory consumption, image quality and performance. Compared to WAvg, HT takes more memory but produces image with good quality quickly, and compared to AT, HT uses the same memory with an unbounded memory implementation, but it uses much less if it's implemented to run on fixed memory. As for the image quality, HT presents a significant improvement on quality with only 4 slots of buffers compared to the original AT and it's already almost the same with the reference image. And HT also managed to produce most accurate image even with an opacity configuration that is difficult for it while other methods all showing some obvious artifacts. The HT method generates the lower error estimation for different alpha/opacity distributions in most cases, but it generates higher error than AT only on the worst case for HT where HT captures no layers as the core. HT also has a reasonable FPS compared with AT.

The results show that, the HT approach shows some advantages over other methods regarding the quality of the image produced. HT provides image with no visible flickering, which caused by different approximations across frames, or any loss of details, which is caused by incorrect evaluation of visibilities. As the HT approach utilizes the GPU parallelization, it is expected to have better performance compared to AT which requires serialization. For a large scene, the results show that HT is capable of rendering in bigger resolutions while the unbounded memory versions of AT and HT run out of memory. This is because HT has fixed and small memory footprint, while the performance is comparable to the unbounded AT. The unbounded HT is also two times faster than the unbounded AT.

In summary, HT is an approach for OIT that combines different techniques to balance the memory consumption and the image quality. The advantage of HT is that it can handle any number of transparent layers, operating in a fixed memory space and it runs on GPUs. It can in particular process complex scenes in high screen resolutions. Future improvements can be made if the GPU supports 64-bit atomic operation.