Progressive Real-Time Rendering of Unprocessed Point Clouds

Presenter: Markus Schuetz

THE PROBLEM

- Rendering millions of points in real time usually requires high-end graphics cards or the use of spatial acceleration structures.
- We introduce a method to progressively display as many points as the GPU memory can hold in real time to get more visually-pleasing results even on notebooks and low-end GPUs.

METHODS

The basic idea of our method is to reduce the amount of points that are drawn each frame. This is done in three passes:

1. Reproject previous frame.
   Render only points that were visible in the previous frame.

2. Add random points.
   Add a random selection of points to fill gaps that appear after transformations.

3. Generate an index buffer from all currently visible points.

4. Repeat.

RESULTS

Heidentor, 26M points

- GPU: 3.14ms 154fps 293k points 0.14ms
- Reproject: 154fps 293k points 0.14ms
- Add random: 3.14ms 154fps 293k points 0.14ms
- Generate index buffer: 0.1ms
- Total: 3.39ms 154fps 293k points 0.14ms
- Coverage in: 154fps 293k points 0.14ms
- Coverage out: 154fps 293k points 0.14ms
- USE GPU memory:

Reitz, 120M points

- GPU: 69.09ms 7.6fps 120M points 1.2ms
- Reproject: 7.6fps 120M points 1.2ms
- Add random: 69.09ms 7.6fps 120M points 1.2ms
- Generate index buffer: 1.2ms
- Total: 69.36ms 7.6fps 120M points 1.2ms
- Coverage in: 7.6fps 120M points 1.2ms
- Coverage out: 7.6fps 120M points 1.2ms
- USE GPU memory:

Candisari, 250M points

- GPU: 107.70ms 5.7fps 250M points 1.2ms
- Reproject: 5.7fps 250M points 1.2ms
- Add random: 107.70ms 5.7fps 250M points 1.2ms
- Generate index buffer: 1.2ms
- Total: 107.98ms 5.7fps 250M points 1.2ms
- Coverage in: 5.7fps 250M points 1.2ms
- Coverage out: 5.7fps 250M points 1.2ms
- USE GPU memory:

Point clouds can be rendered 3x to 35x faster with a combination of progressive rendering and an incremental shuffled vertex buffer object.

Our Approach...

- Distributes the workload of rendering a single, large block of points over multiple frames, without the need to generate acceleration structures in advance.
- Routes details that were already drawn in previous frames and progresses uniformly towards the finished result, typically in less than a second.
- Is designed to work while points are being loaded or scanned so that users can immediately start seeing results.
- Uses a single, randomly shuffled array of points as its data structure.
- Shuffling happens incrementally while points are loaded.
- Allows users to explore any point cloud that fits into GPU memory in real time.

Related Work

- Futterleib et al. developed a method that accumulates detail when the camera is still and creates a new vertex buffer from visible points in discrete intervals, in order to preserve the accumulated details when the camera moves again [1]. Our method differs in that we create an index buffer every frame, instead of a vertex buffer in discrete intervals.
- Similar to our approach, Pinto et al. reprojects every frame to the next, but they add nodes of a hierarchical structure, instead [2]. As such, it converges faster but in non-uniform way, and it requires a hierarchical structure.

References


https://doi.org/10.1109/ICVR.2017.8007521

Acknowledgements

We would like to thank the following institutions for providing the respective data sets:

- Heidentor: Ludwig Boltmann Institute for Archaeological Prospection and Virtual Archaeology
- Reitz courtesy of NAUL, Laser Measurement Systems
- Candisari courtesy of TU Wien, Institute of History of Art, Building Archaeology and Restoration

Markus Schuetz, Michael Wimmer
TU Wien
http://cg.tuwien.ac.at
mschuetz@cg.tuwien.ac.at
wimmer@cg.tuwien.ac.at

download the full paper at

bit.ly/2JByVBP

code samples:
https://github.com/m-schuetz/siggraph2018