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

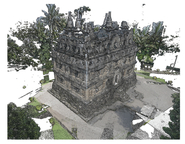
GPU	bruteforce	ours	#add	converges in
1080 GTX	**8.483ms**	**2.154ms**	3M	9 frames / 0.02s
1060 GTX	**13.554ms**	**3.414ms**	2M	13 frames / 0.05s
940 MX	**37.311ms**	**11.281ms**	1M	26 frames / 0.30s

Retz, 120M points

GPU	bruteforce	ours	#add	converges in
1080 GTX	**46.289ms**	**2.892ms**	3M	40 frames / 0.12s
1060 GTX	**59.736ms**	**5.642ms**	2M	60 frames / 0.34s
940 MX	<not enough GPU memory>			

Candi Sari, 250M points

GPU	bruteforce	ours	#add	converges in
1080 GTX	**98.459ms**	**2.744ms**	3M	83 frames / 0.23 s
1060 GTX	<not enough GPU memory>			
940 MX	<not enough GPU memory>			

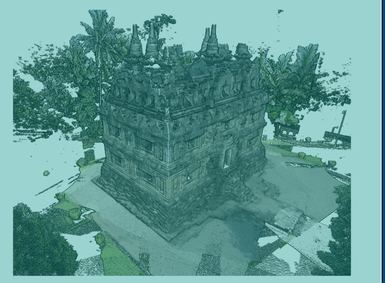


SIGGRAPH 2020 THINK BEYOND
19-23 JULY WASHINGTON DC

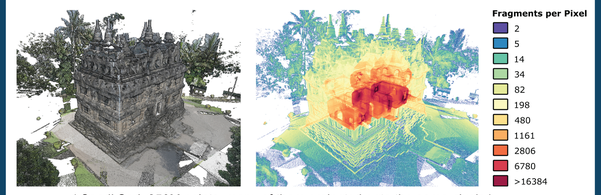
Point clouds can be rendered 3x to 35x faster with a combination of progressive rendering and an incremental shuffled vertex buffer object.

26M points, 3.9x faster

120M points, 16.0x faster



250M points, 35.8x faster



(Candi Sari, 250M points, many of them projected onto the same pixels)

Our Approach...

- distributes the workload of rendering a single, large blob of points over multiple frames, without the need to generate acceleration structures in advance.
- reuses 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 see 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

- Futterlieb et al. developed a method that accumulates detail when the camera is still and creates a new vertex buffer from visible points in discrete intervalls, 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 intervalls.
- Similar to our approach, Ponto 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

[1] Jörg Futterlieb, Christian Teutsch, and Dirk Berndt. 2016. Smooth visualization of large point clouds. IADIS International Journal on Computer Science and Information

[2] K. Ponto, R. Tredinnick, and G. Casper. 2017. Simulating the experience of home environments. In 2017 International Conference on Virtual Rehabilitation (ICVR). 1–9. 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 Boltzmann Institute for Archaeological Prospection and Virtual Archaeology
• Retz courtesy of *RIEGL* Laser Measurement Systems
• Candi Sari 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