A lightweight 3D viewer: real-time rendering of multi-scale 3D surface models

Rui Li
rui.li@sorbonne-universite.fr
Laboratoire Jacques-Louis Lions, Sorbonne University
Paris, France

Figure 1: The rendering result of our lightweight 3D viewer, our viewer is able to fit models [Nouri et al. 2017] [Levoy et al. 2000] of different shapes, attributes, and scales, with optimal visual quality.

ABSTRACT

In this paper, we proposed a lightweight 3D viewer for rendering the large-scale surface model, which allows users to visualize and check the 3D models without booting up the processor-hungry software. The scheme of the viewer includes hierarchy level-of-detail generation (HLOD) and real-time rendering, which avoids LOD-popping and mesh cracks between LOD parts based on vertex interpolation between child and parent. The implementation of the viewer could render scanned models of multiples tens of million triangles at optimal visual quality and interactive frame rate on a computer with integrated GPU or discrete GPU.

KEYWORDS

Real-time rendering, 3D visualization, Multi-resolution modeling

ACM Reference Format:

Rui Li . 2023. A lightweight 3D viewer: real-time rendering of multi-scale 3D surface models. In *Proceedings of 27th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (i3D 2023 posters).* ACM, New York, NY, USA, 2 pages. https://doi.org/https://doi.org/none

1 INTRODUCTION

An efficient and affordable 3D viewer is essential to those who work with 3D files, such as graphic designers, researchers, and engineers. It is also an indispensable tool for 3D model inspection, which does not take up much memory and loading time. In this paper, we present a 3D viewer which can render the different models of different scales. The HLOD construction achieves competitive levels of the order of 1 million input triangles per second per core (on the mainstream computer), and the real-time visualization based on HLOD can avoid LOD popping and cracks with an interactive frame rate and optimal visual effect. The child-parent relation between vertices of successive HLODs is designed to be compatible with parallel preprocessing while also allowing for viewpoint-dependent vertex interpolation, which ensures a smooth transition without any cracks or popping artifacts.

2 THE OVERVIEW OF OUR METHOD

The framework of our viewer is demonstrated in Figure 2. In order to accelerate HLOD construction, parallel mesh simplification and parallel LOD construction algorithms are utilized. The model M is first partitioned (not cut) according to a Euclidean cubic grid, and the HLOD is then built by merging and then simplifying neighboring elements of the partition in an octree-like fashion, fine-to-coarse. Some freedom applies to the simplification algorithm, but it must provide a child-parent relation between vertices of successive LODs (this is certainly the case when using only edge collapses, possibly with relocation). We assume the following simplification constraint, for each edge in M_L has a length of at most $\sigma_0 2^{-L}$, where L is the highest level of LODs and $\sigma_0 > 0$ is the first model parameter; for each $1 \leq l \leq L$ and each vertex $v \in M_l$,

$$d(v, P(v)) \leq \sigma_0 2^{-(l+1)}, \tag{1}$$

where $P(v)$ is the parent vertex of v. A consequence of these two

Proceedings of 27th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (i3D 2023 posters), May 03-05, 2023, Bellevue, WA, United States

ACM ISBN 978-x-xxxx-xxxx-x.
https://doi.org/https://doi.org/none

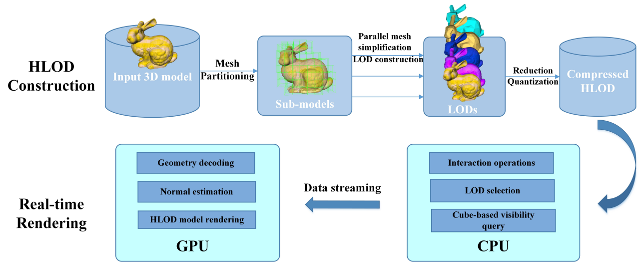


Figure 2: The pipeline of 3D viewer.

conditions is that for $1 \leq l \leq L$, each edge in M_l has a length of at most $\sigma_0 2^{-l}$.

In the rendering stage, the octree-based hierarchy model is traversed coarse-to-fine to select the satisfied cube of the appropriate resolution based on the position of viewpoint, a cube C_l is said to be suitable for visualization if either $l = L$ or

$$d(v_p, C_l) \geq \kappa_0 2^{-l}, \tag{2}$$

where $\kappa_0 > 0$ is a second constant in the model, v_p is the position of the viewpoint.

Vertex interpolation between child and parent between the LODs is combined with suitable constraints (σ_0) in the HLOD construction stage, which guarantees that the actual vertex is visualized depends continuously on the viewpoint position.

For the visualization of $M(C_l)$, we will not draw an arbitrary vertex $\omega \in V(C_l)$ at its original position, but at the position n of viewpoint-dependent and LOD-dependent:

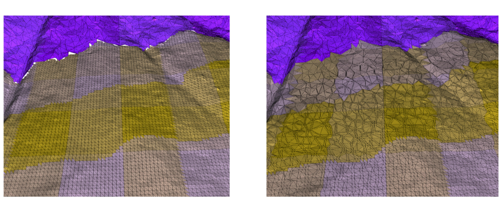
$$n = \lambda\omega + (1 - \lambda)P(\omega), \tag{3}$$

where the interpolation factor $\lambda \in [0, 1]$ is of the form

$$\lambda = b(d(v_p, w)/2^{-l}). \tag{4}$$

For some fixed continuously decreasing blending function $b : \mathbb{R}^+ \rightarrow [0, 1]$.

Therefore no crack will occur at the border between the morphing of the neighbor meshlets $M(C_l)$ and $M(C_{l+1})$, it prevents the LOD popping and boundary cracks by vertex interpolation between LOD. The vertex interpolation result is shown in Figure 3.



(a) Original vertex position (b) Viewpoint-dependent and LOD-dependent vertex position

Figure 3: The viewpoint-dependent and LOD-dependent vertex position produce a blending band to eliminate the cracks.

3 RESULTS

We test our 3D viewer on two different platforms: a desktop equipped with an integrated GPU, and a laptop equipped with Nvidia RTX A4500 (16GB). The test result is based on different scale models as Figure 1 shown. We implemented HLOD construction and viewpoint-based visualization in turn. Table 1 shows the HLOD construction time of large-scale models and quantitative results of HLOD structure, where n_v is the number of vertices and n_t is the number of triangles. It proves that our method is fast and memory efficient.

Table 1: Quantitative results of HLOD construction

Object	M_L		HLOD		Time (s)
	n_v	n_t	n_v	n_t	
David	28184526	56230343	40469139	75553060	24.02
Paint	22548641	45056702	45155593	69596934	21.60
Temple	30001155	60000000	43632901	82369832	35.00
Terrain	55410740	110757478	79843353	149790362	34.24

By controlling the movement of the viewpoint to the model from a distance, real-time rendering based on HLOD was performed. In our accompanying video, the visualization result on two platforms was recorded (the process of real-time vertex interpolation and overdraw analysis), and we also show the comparison with the Nexus [Visual Computing Laboratory 2020], it can eliminate mesh cracks but not avoid the LOD popping. Our viewer can realize real-time rendering of the textured Egyptian temple model (60M triangles) in 170 fps on a desktop and has better performance on a laptop with a discrete GPU. At the same time, we conducted an overdraw analysis to prove our interpolation method will not aggravate the overdraw problem.

4 CONCLUSION

We present a lightweight 3D viewer, which can be used for the efficient construction of HLODs and real-time rendering of large surface models. With its lightweight structure and optimized performance, our 3D viewer enables users to visualize and check 3D models without the need for resource-intensive software or extensive preprocessing times.

ACKNOWLEDGMENTS

This work is supported by the Institute of computing and data science (ISCD) of Sorbonne University. I would like to thank my advisors Didier Smets and Philippe Walter for all their help. We are grateful to the anonymous reviewers for their suggestions and to the Digital Michelangelo project for making data sets available.

REFERENCES

Marc Levoy, Kari Pulli, Brian Curless, Szymon Rusinkiewicz, David Koller, Lucas Pereira, Matt Ginzton, Sean Anderson, James Davis, Jeremy Ginsberg, et al. 2000. The digital Michelangelo project: 3D scanning of large statues. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. 131–144.

Anass Nouri, Christophe Charrier, and Olivier Lézoray. 2017. *Technical report: Greyc 3D colored mesh database.* Ph. D. Dissertation. Normandie Université, Unicaen, EnsiCaen, CNRS, GREYC UMR 6072.

CNR Visual Computing Laboratory, ISTI. 2020. Nexus. https://github.com/cnr-isti-vclab/nexus