

Evaluation of an Appearance-Preserving Mesh Simplification Scheme for Configura AB

Rasmus Hedin <rashe877@student.liu.se>
Erik S. V. Jansson <erija578@student.liu.se>

Contents

Contents

| | | |
|----------|------------------------------|----------|
| 1 | Introduction | 1 |
| 1.1 | Motivation | 1 |
| 1.2 | Aim | 1 |
| 1.3 | Research Questions | 1 |
| 1.4 | Delimitations | 2 |
| 1.5 | Background | 2 |
| | Bibliography | 3 |



1 Introduction

1.1 Motivation

Rendering of large polygonal meshes in real-time consumes valuable computation time that could otherwise be used for other problems. If the amount of polygons can be reduced without affecting the visual quality of the final render significantly, then we can save computation time and memory. Algorithms that reduce the polygon count of a mesh based on some metric are called mesh simplification algorithms. An issue that has been found in industry and research is that the quality of the texture coordinates used for texture mapping are degraded when this simplification algorithm is applied.

1.2 Aim

First to survey the field for state-of-the-art mesh simplification algorithms that preserve the visual appearance of meshes and then integrate these into Configura's graphics pipeline. Thereafter, we evaluate each of these by measuring the algorithm's performance and appearance preservation properties. Our goal is to find the mesh simplification algorithm which suits Configura's requirements best.

1.3 Research Questions

1. How can *mesh simplification* be done without affecting the *visual appearance* significantly?
2. What are the alternatives to achieve *mesh simplification* with *appearance preservation*?
3. Which alternative gives the best effect considering *performance* and *appearance preservation*?
 - a) When measuring the algorithm's *computation time* while targeting an *appearance threshold*?
 - b) When measuring the algorithm's *memory usage* while targeting an *appearance threshold*?
 - c) When measuring the *rendering time* of the simplified mesh?

4. Which alternative gives the best *appearance preservation* when targeting a certain *polygon count threshold*?

1.4 Delimitations

Since there are many mesh simplification algorithms in previous work, a proper literature review would have to be done to find possible candidates for implementation. However, since this thesis is mostly concerned with implementing and measuring the performance, we've decided to base our choices on existing surveys and literature reviews to skip doing a literature review ourselves.

Also, since implementing and doing measurements on all algorithms would take too long, we've decided to only pick a interesting subset of the algorithms presented in the surveys.

1.5 Background



Bibliography

- [1] Paolo Cignoni, Claudio Montani, and Roberto Scopigno. “A comparison of mesh simplification algorithms”. In: *Computers & Graphics* 22.1 (1998), pp. 37–54.
- [2] Jonathan David Cohen. “Appearance-preserving Simplification of Polygonal Models”. AAI9914826. PhD thesis. 1998. ISBN: 0-599-13476-3.
- [3] Jonathan Cohen, Marc Olano, and Dinesh Manocha. “Appearance-preserving simplification”. In: *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*. ACM. 1998, pp. 115–122.
- [4] Michael Garland and Paul S Heckbert. “Simplifying surfaces with color and texture using quadric error metrics”. In: *Visualization’98. Proceedings*. IEEE. 1998, pp. 263–269.
- [5] Hugues Hoppe. “Progressive meshes”. In: *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. ACM. 1996, pp. 99–108.
- [6] Jacob Kroon and Anna Nilsson. “Game Developer Index 2017”. In: *Based on 2017 Annual Reports. Dataspelbrancher* (2017).
- [7] Peter Lindstrom and Greg Turk. “Image-driven simplification”. In: *ACM Transactions on Graphics (ToG)* 19.3 (2000), pp. 204–241.
- [8] David P Luebke. “A developer’s survey of polygonal simplification algorithms”. In: *IEEE Computer Graphics and Applications* 21.3 (2001), pp. 24–35.
- [9] Pedro V Sander, John Snyder, Steven J Gortler, and Hugues Hoppe. “Texture mapping progressive meshes”. In: *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM. 2001, pp. 409–416.
- [10] Jerry O Talton. “A short survey of mesh simplification algorithms”. In: *University of Illinois at Urbana-Champaign* (2004).