Out of Core Simplification

Benjamin Watson

Dept. Computer Science

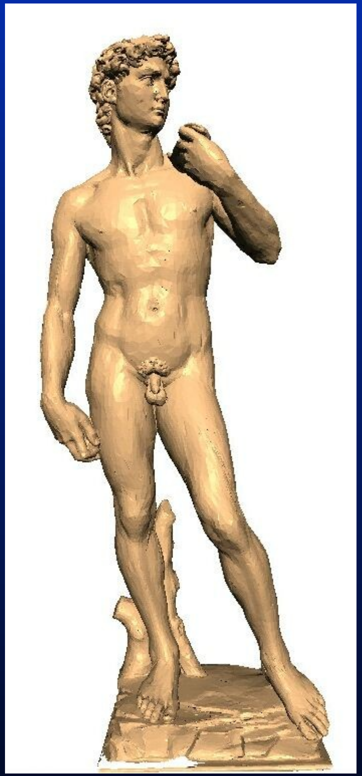
Northwestern University

watson@cs.northwestern.edu

SAN ANTONIO SIGGRAPH 2002

Models are getting bigger

- Models now ride Moore's Law
  - *Major source: 3D scanning*
  - *Example: Digital Michelangelo*
  - *Sizes currently in the 300 millions*
  - *Well beyond most core memories*



Can’t we do “big”?

Maybe, but not “massive”

- *That is, models not fitting in core*
- *Previous limit less than 10 million faces*

What’s the problem with out of core?

- *Requires slow disk access*
- *So must minimize disk access*
- *Most simplification algs don’t*

Out of core strategies

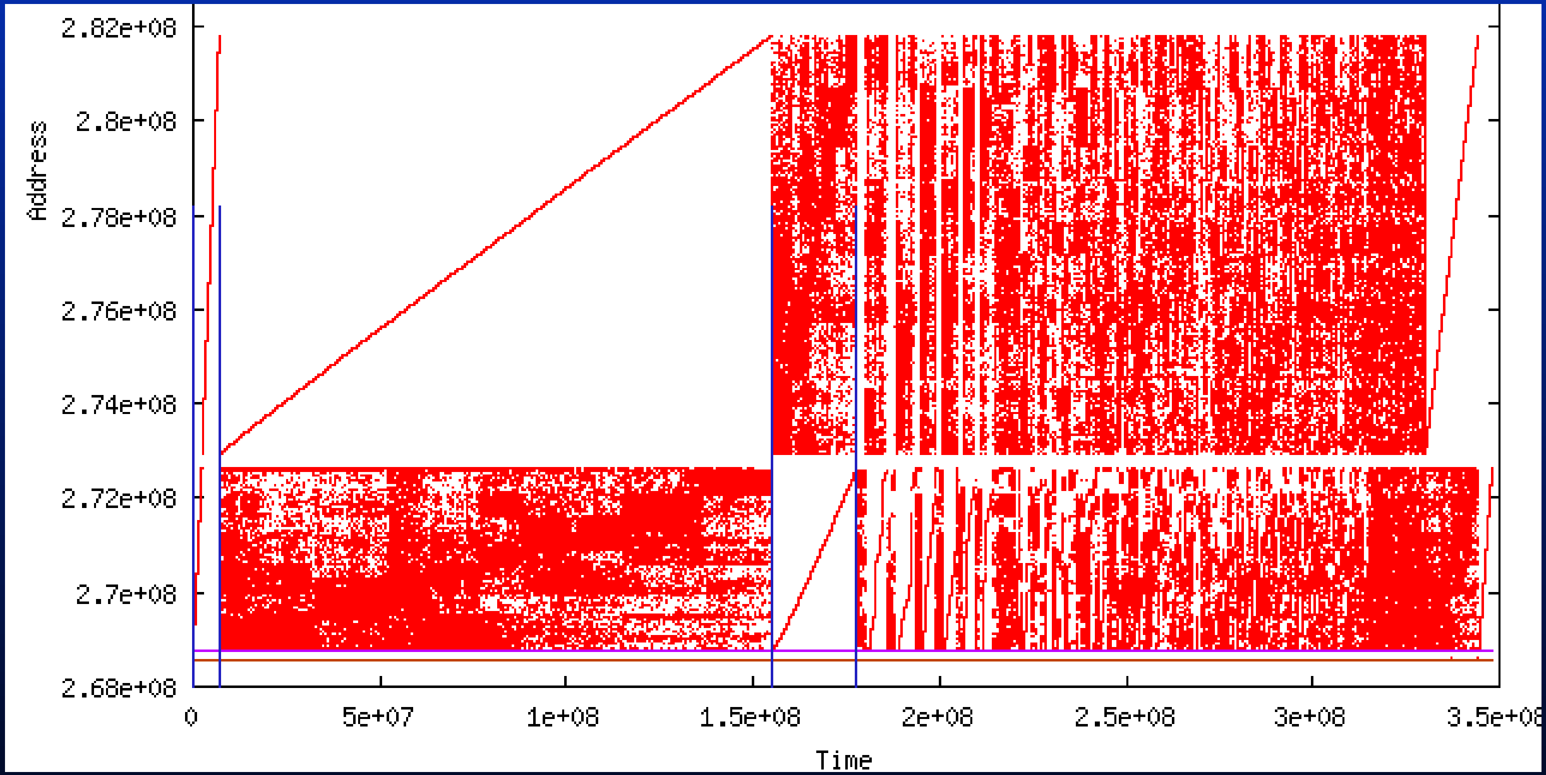
For good out of core performance, use

- *Locality (reducing working set)*
- *Reuse (minimizing swapping)*

Most existing algorithms poor at both

- *Locality not guaranteed in model formats*
- *Most algorithms are greedy -- poor reuse*

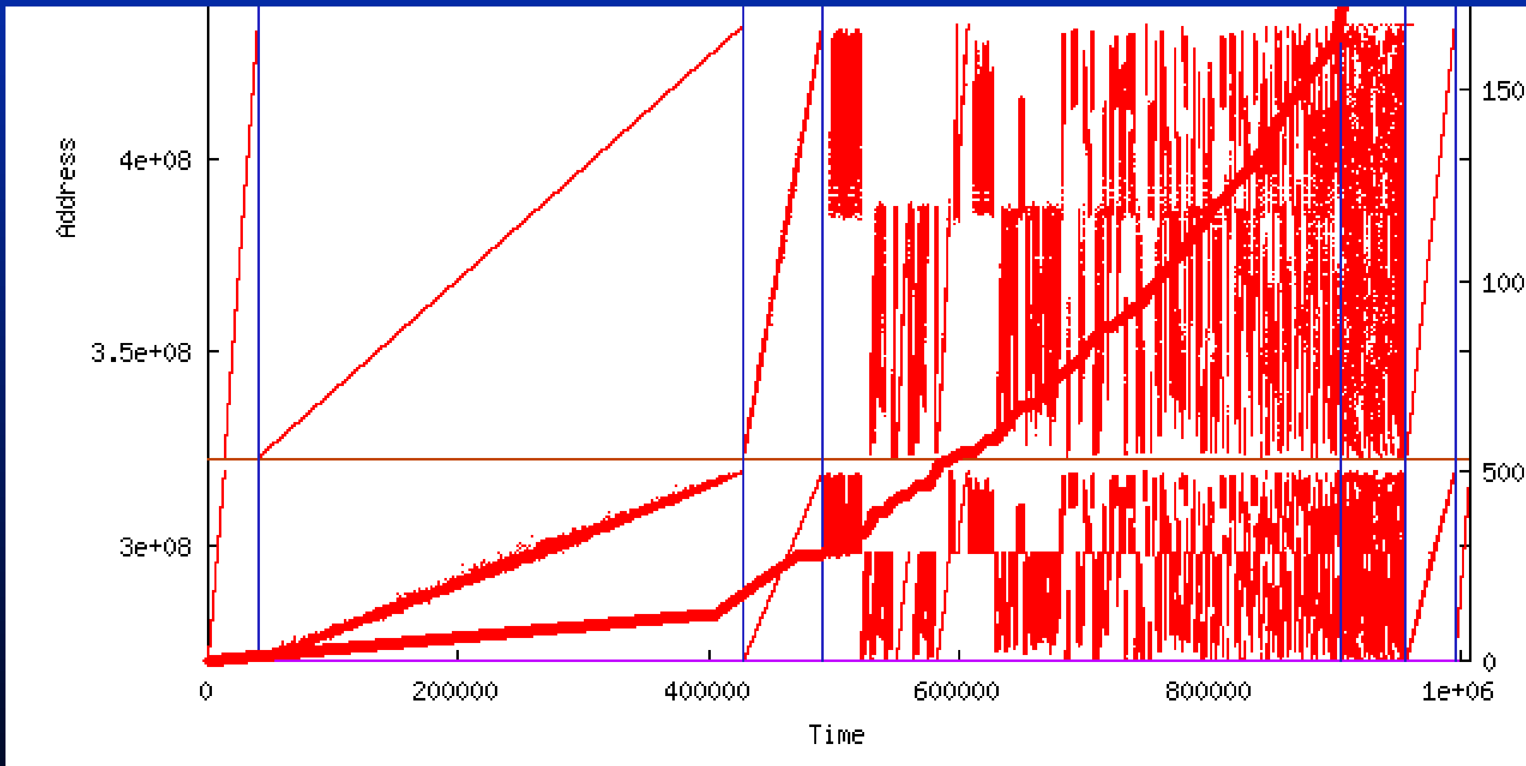
Demo: *bunny vs. dragon*



RSimp on bunny

SAN ANTONIO SIGGRAPH 2002

Demo: *bunny vs. dragon*



RSimp on dragon

Solutions: *Lindstrom*

Modification of Rossignac & Borrel

Adds locality by deref’ing to create “soup”

- *Done w/ little thrashing in linear time*

Hashes vertices on each input face

- *Add normal to quadric in each vertex hash entry*
- *Retain face if 3 vertices hash differently*

Output retained faces, quadric mins

Solutions: *Lindstrom*

Advantages

- *Extremely fast: single linear pass on "soup"*
- *56 million faces in several minutes*
- *Can handle 100s millions of faces*

Disadvantages

- *Poor accuracy: a non-adaptive algorithm*
- *Not sensitive to topology*

Solutions: *Lindstrom*

(2K faces) (20K faces) (200K faces)

Solutions: *Shaffer & Garland*

Addition to Lindstrom's approach

First, apply Lindstrom's algorithm

Resulting model fits in core memory

Then, adaptively simplify

Using refining algorithm similar to `RSimp`

(We discuss `RSimp` *shortly)*

Solutions: *Shaffer & Garland*

Advantages

- *Improved mean accuracy about 35%*

Disadvantages

- *Somewhat slower*
- *Not sensitive to topology*
- *Introduces spurious topological joins*
- *Limited output size*

Solutions: *Shaffer & Garland*

(2K faces) (20K faces) (200K faces)

SAN ANTONIO SIGGRAPH 2002

Solutions: `VMRSimp`

Modification of `RSimp` by Brodsky & Watson

`RSimp` refined toward desired output size by

- *Define a poor 8 patch (vertex) approximation*
- *Repeat*
    - *Choose patch with most normal variation*
    - *Split patch according to normal variation*
- *Until desired number vertices reached*

Solutions: `VMRSimp`

- Modification makes simplification a sort
  - *Each patch a range on input array*
  - *Splitting patch means sorting into subranges*
- Thus locality is built and refined
  - *Allows reliance on virtual memory*
  - *Added modification allows quality/reuse tradeoff*
- 56M faces in 32 bit address space

Solutions: `VMRSimp`

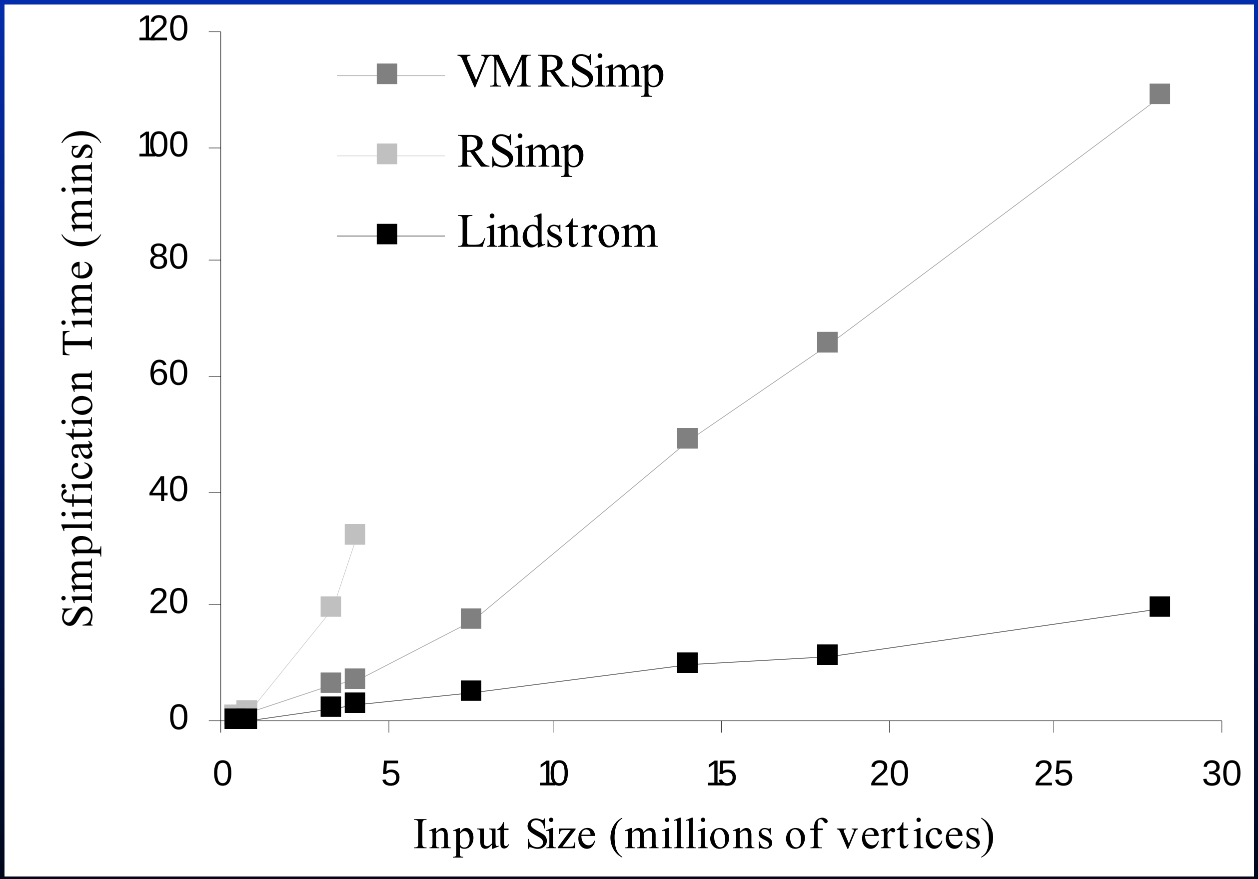
Advantages

- *Mean accuracy improves additional 30%*
- *Maximum error reduced 2-5 times*
- *Topological sensitivity (boundaries, joins)*
- *Very large output sizes (10M+) possible*

Disadvantages

- *About twice as slow as Shaffer & Garland*
- *Can reach 32 bit address space limits*

Solutions: `VMRSimp`



1 GHz PIII
RH Linux 7.1
1 GB Mem

Accuracy control improves times 25%

Solutions: `VMRSimp`

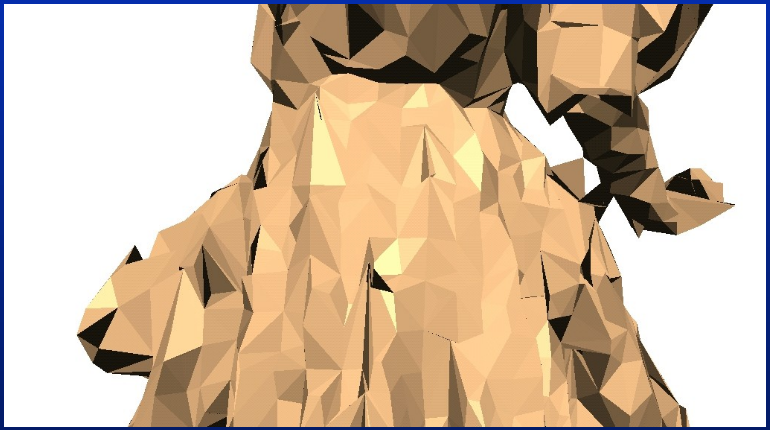
Output Tris	Lindstrom		Schaffer and Garland		`VMRSimp`	
	mean	***max***	***mean***	***max***	***mean***	***max***
1K	0.4549	26.10	0.4821	25.91	0.3450	14.89
10K	0.0986	24.35	0.0946	24.43	0.0598	12.80
100K	0.0266	24.47	0.0164	24.17	0.0119	10.45

Metro error as % of model bounding box

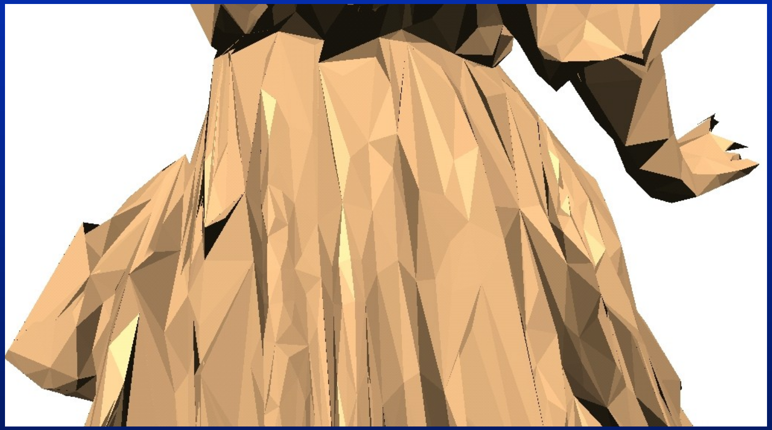
Solutions: `VMRSimp`

(2K faces) (20K faces) (200K faces)

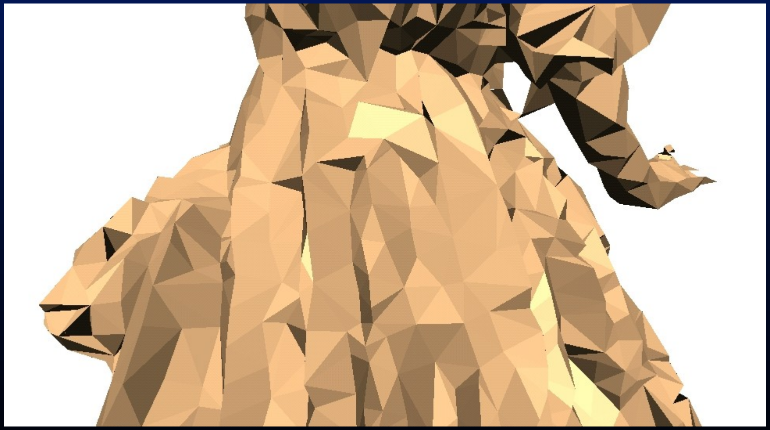
Solutions: *comparison*



Lindstrom



VMRSimp



Shaffer & Garland