

# GLOD: A Geometric Level of Detail System at the OpenGL Driver Level

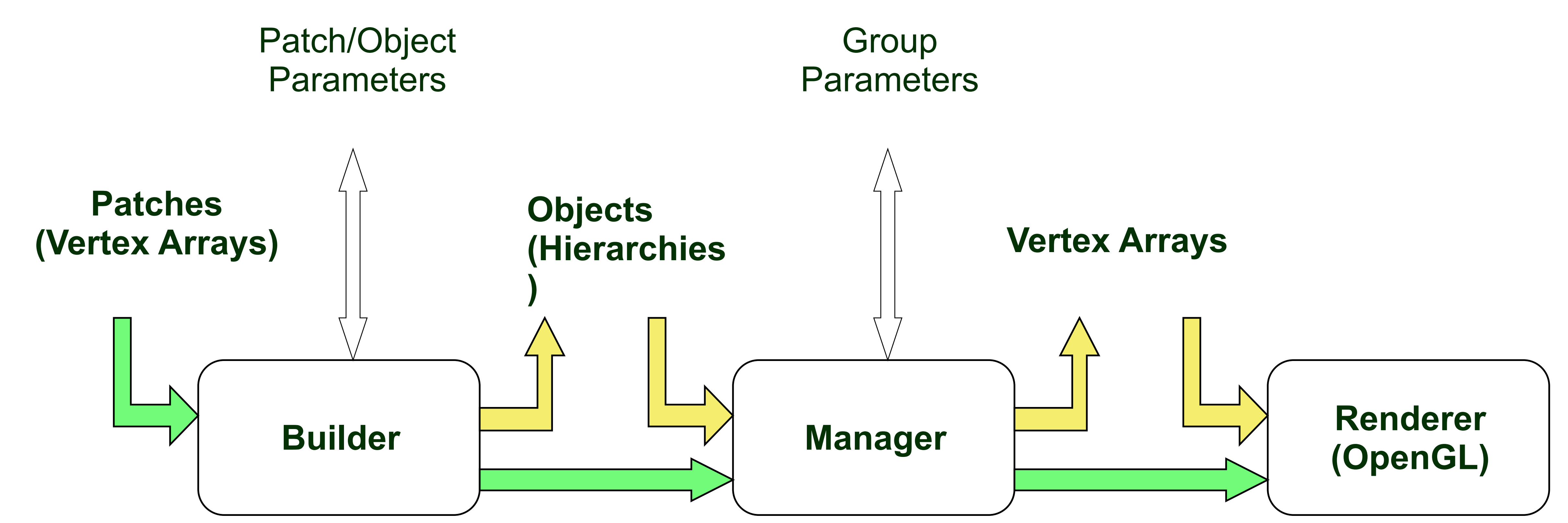
**Jonathan Cohen**  
**Nathaniel Duca**  
 Johns Hopkins University

**David Luebke**  
**Brenden Schubert**  
 University of Virginia

## A Widely-deployable Simplification System

- **Easy to use**
  - Similar semantics to OpenGL driver
- **Powerful**
  - LOD generation, management, and rendering
- **Flexible**
  - Simple switches make dramatic changes in LOD algorithms
- **Incremental adoption**
  - Use just the desired components
- **Extensible**
  - Developers can wrap new simplification algorithms in same API or even add to current open source system

## GLOD Dataflow Model for Incremental Adoption



- Let GLOD handle your data from start to finish
- OR: Build your own hierarchy and hand it to GLOD for management and rendering
- OR: Read back entire hierarchy to avoid build time in later program executions
- OR: Read back adapted hierarchy and do your own rendering

## GLOD Concepts

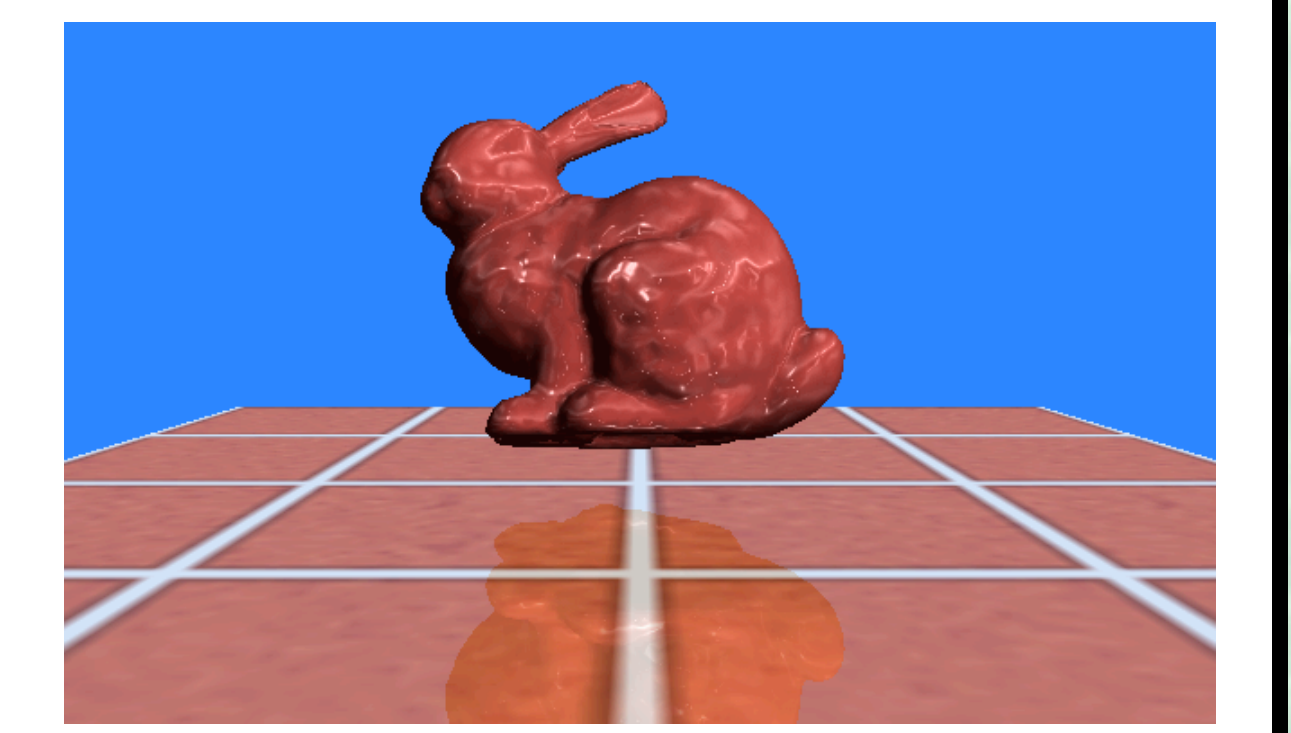
- Patches: unit of rendering**
  - Created via vertex arrays
  - App sets rendering state before drawing
- Objects: unit of hierarchy building**
  - 1 or more patches simplified as an object
- Groups: unit of adaptation**
  - 1 or more objects adapted together

## Easy-to-switch Parameters

- Hierarchy targets**
  - Discrete, continuous (view-dependent)...
- Build operators**
  - Edge collapse, vertex pair, vertex cluster...
- Error metrics**
  - Spheres, error quadrics, texture deviation...
- Adaptation modes**
  - Screen-space error threshold, triangle budget...

## A Matter of Policy...

- GLOD doesn't change OpenGL rendering state**
  - The application retains control
  - Avoids limiting the range of rendering algorithms
  - GLOD doesn't need to know...



Download GLOD today (version 1.0 pre-release 2) from: <http://www.cs.jhu.edu/~graphics/GLOD>