Point Rendering for Impostors

Andreas Bærentzen IMM

What is so great about points?

PRO:

- Very simple to render
- Faster than triangles in some cases
- Generated by acquisition devices
- Lead to a natural LOD scheme

CON:

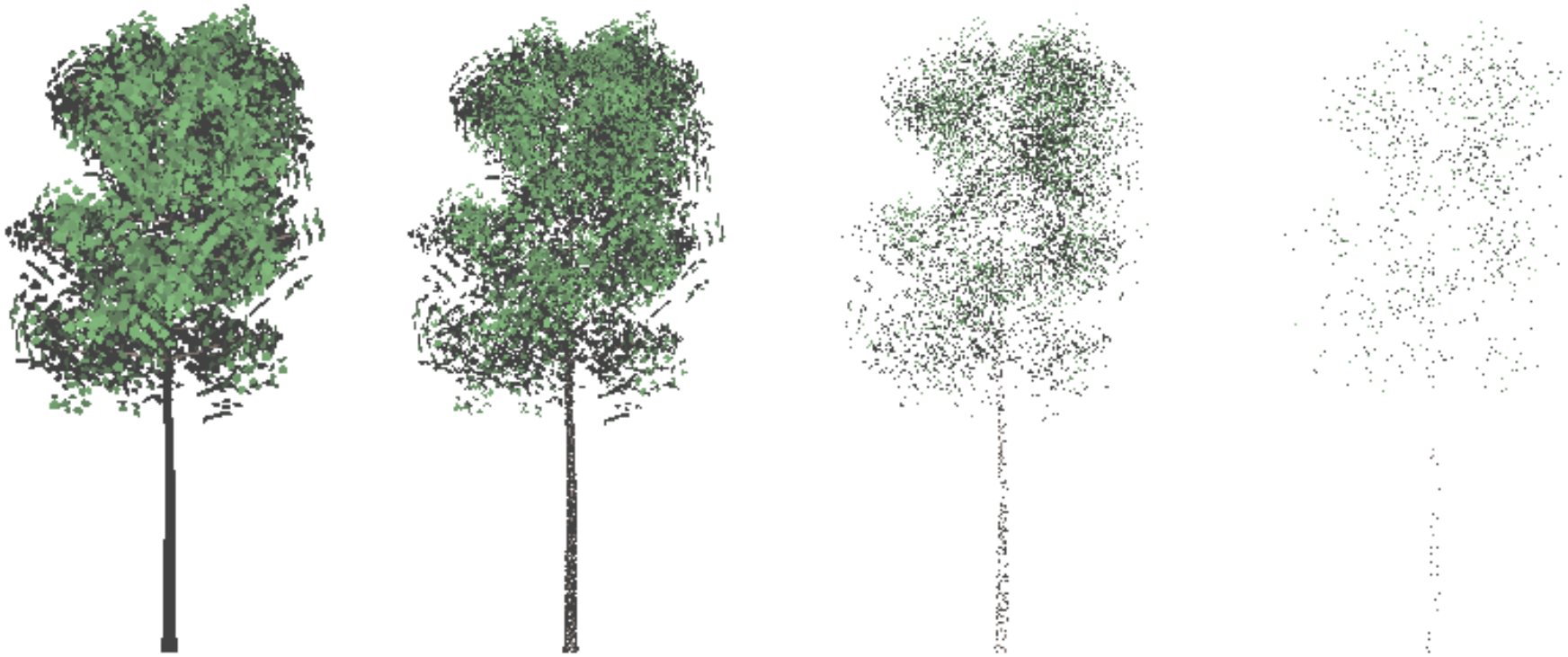
- Present hardware not optimized for points … yet
- Point cloud should be dense

[Alexa et al.]

Idea #1: Point Impostors

The **same** set of points are used to represent all LODs

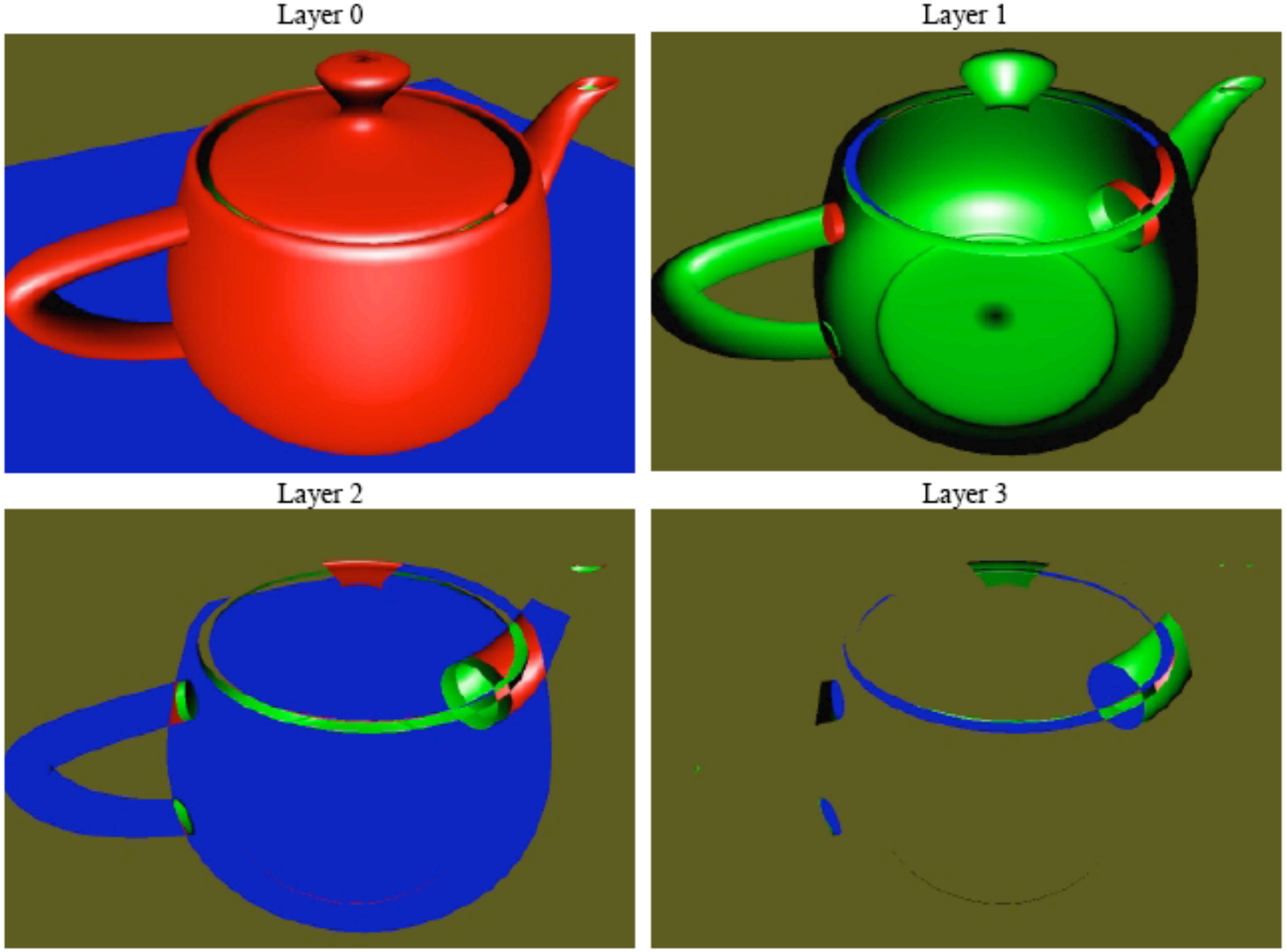
Vertex Buffer Objects make it fast(er)



Idea #2: Generate Points using Graphics HW

- A Z-buffer makes it possible to extract first visible layer of an object
- With two Z-buffers it is possible to render all layers in an object.
- The process is called depth peeling [Everitt et al.]
- Implementation is possible thanks to shadow mapping extensions.

Depth Peeling



[Cass Everitt, NVIDIA]

The little details

- Each (non-background) pixel in each layer represents a point.
- For each layer, we extract z-buffer and attributes (e.g. color, normal, texture)
- We can extract as many attributes as we like.
- Repeat from X, Y and Z directions to get even representation
- Use normals to decide what points to use from each of X, Y, and Z directions.

DEMO

Conclusions

- Points yield very nice continuous LOD impostors
- Point sets are easily generated using depth peeling.
- Technique is possible now.
- Its time is *perhaps* yet to come.