UCLA

CS174A : Introduction to Computer Graphics

Royce 190
TT 4-6pm

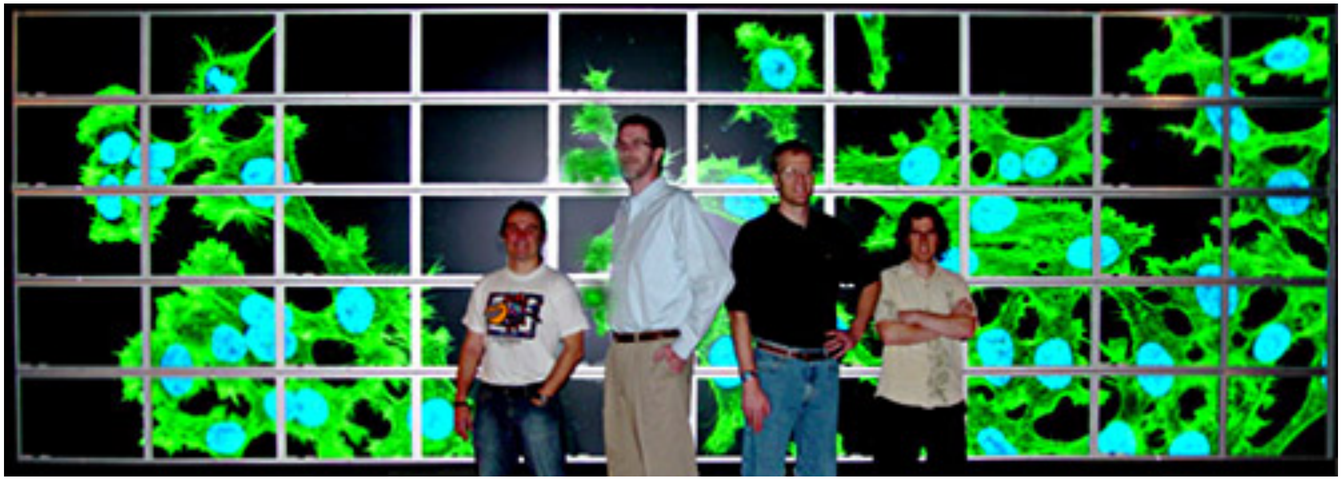
Scott Friedman, Ph.D
UCLA Institute for Digital Research and Education

Parallel Rendering

- Sometimes a single computer is not enough.
- For rendering this manifests itself in at least two ways.
    - Not enough pixels
    - Not enough polygons

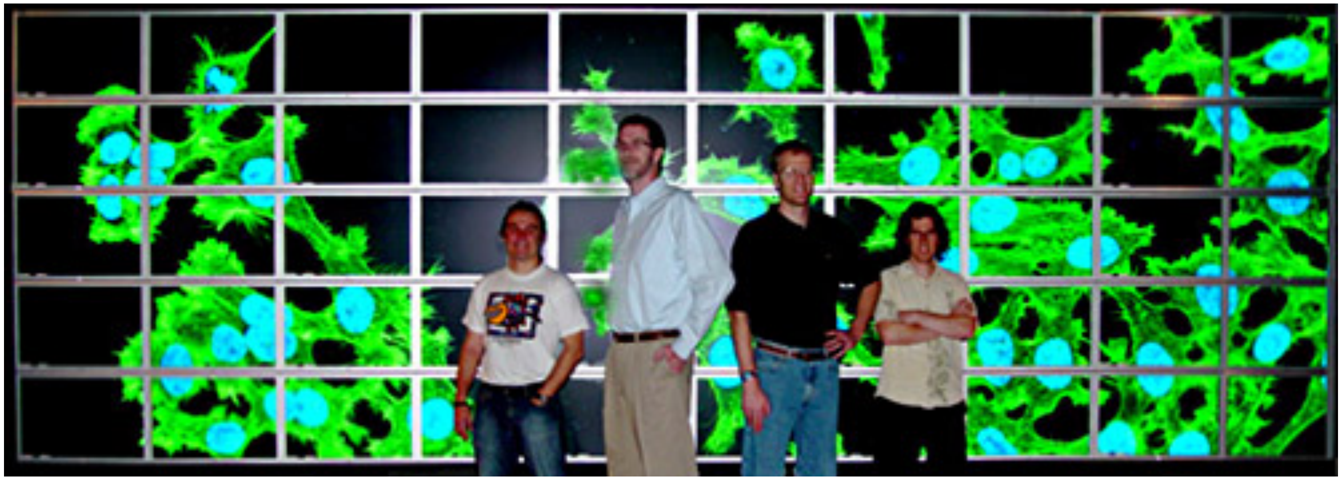
Parallel Rendering

- Not enough pixels
  - Even high-res displays are not high enough.
    - » HDTV is about 2 megapixels
  - Tile high-res display together.
  - Common term is a power wall.



Parallel Rendering

- Not enough pixels
  - These tiled displays can reach 150 megapixels or more.
  - SAGE is a library used to manage these types of walls developed by UIC

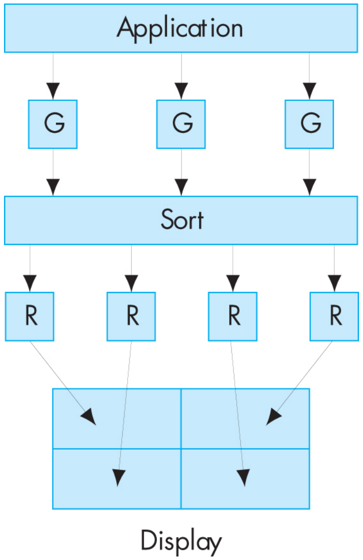


Parallel Rendering

- Not enough polygons
    - Parallelizing increases throughput
    - How we parallelize depends on the problem
    - All graphics parallelization breaks down into three types.
        - Sort first
        - Sort middle
        - Sort last

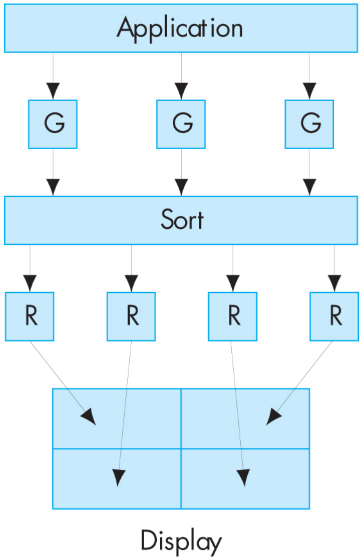
Parallel Rendering

- Not enough polygons – sort middle
  - How all GPUs work these days.
  - Any number of geometry processors (G)
  - Any number of fragment rasterizers (R)



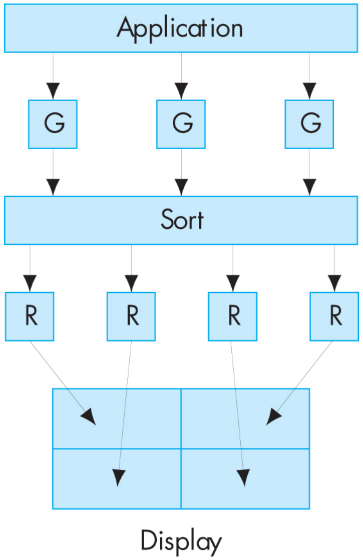
Parallel Rendering

- Not enough polygons – sort middle
  - Each rasterizer is associated with a part of the display
  - Primitives are *sorted* to the rasterizer that corresponds to the projected area of the primitive.



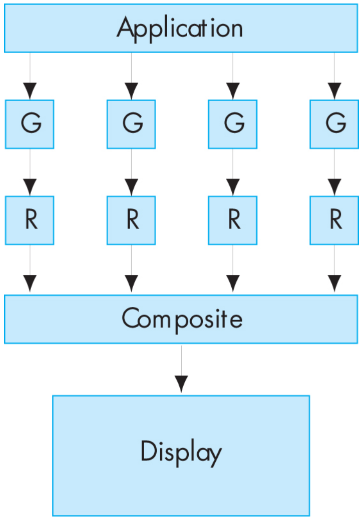
Parallel Rendering

- Not enough polygons – sort middle
  - This solution load balances fairly well on a GPU.
  - Difficult at the application level.



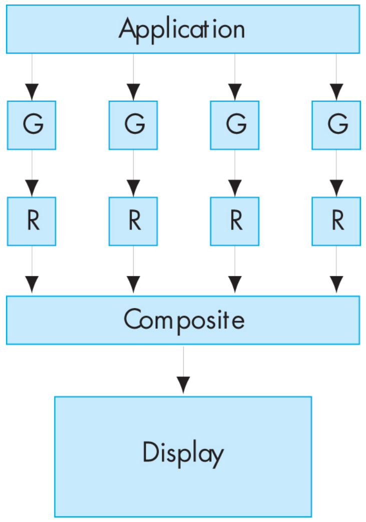
Parallel Rendering

- Not enough polygons – sort last
  - Geometry and rasterization are handled by a single unit.
  - We can load balance across all units.



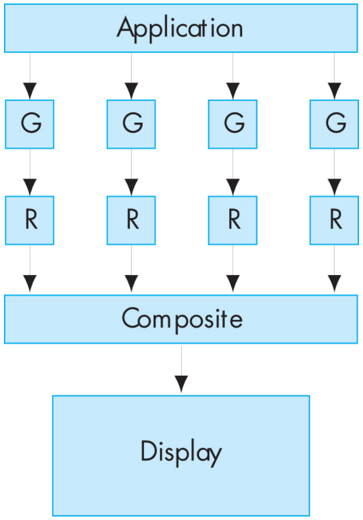
Parallel Rendering

- Not enough polygons – sort last
  - Even though we have very good load balancing for rendering.
  - We have to composite all the pieces back together.



Parallel Rendering

- Not enough polygons – sort last
    - Compositing is a problem.
        - Each rendering system can potentially render to the entire display.
        - To composite the result requires depth information

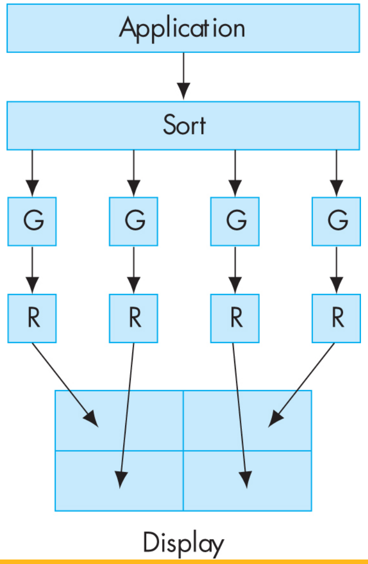


Parallel Rendering

- Not enough polygons – sort last
  - Compositing involves reading the entire color and depth buffer and sending over a bus or network.
  - All these buffers are combined into a final image.
  - A fast network is required.
  - Load balancing is great.
  - Speed can be an issue
    - Readback, network, depth processing, upload

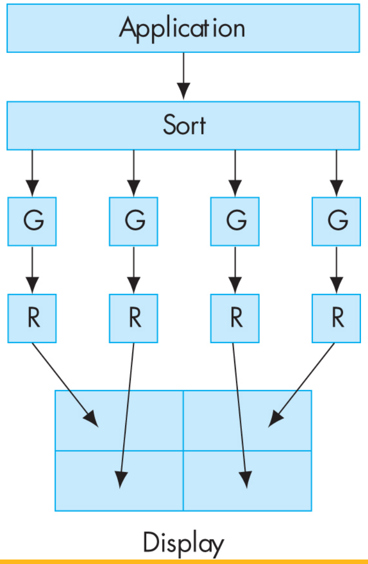
Parallel Rendering

- Not enough polygons – sort first
  - Objects are sorted to the renderer that handles the part of the display it will be projected to.
  - Hard to load balance.



Parallel Rendering

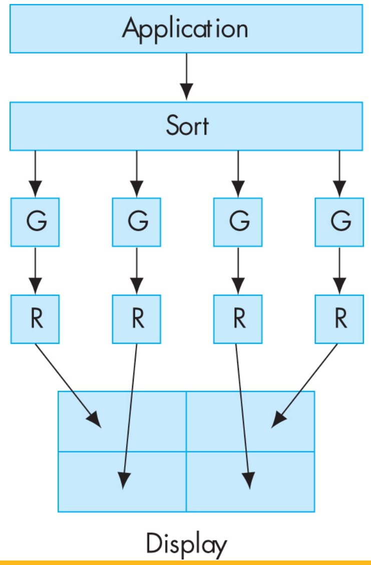
- Not enough polygons – sort first
  - Load balance is hard but compositing is faster.
  - Trick is to quickly determine where objects will land onscreen.



UCLA

Parallel Rendering

- Not enough polygons – sort first
  - One must also have some estimate of how long it will take to render objects in any scheme.
  - One way around this is to adjust the screen partitions.



Parallel Rendering

- Not enough polygons – sort first
    - Videos.