Outline

- Related work
- Recap
- Algorithm
  - Assign clusters
  - Find unique clusters
  - Assign lights
  - Shading
  - Forward Shading
- Results
- Conclusion & questions
Related work

● Deferred shading
  ○ solves wasting lighting calculations

● Tiled deferred shading
  ○ solves bandwidth bottleneck of deferred shading
Recap - Deferred shading

- Fill the G-buffer
  - render depth, normal and albedo to different render targets.
Recap - Deferred shading

- Render light source bounding volume

- For all pixels inside the volume
  - read data from G-buffer
  - apply lighting
  - blend with previous lighting

- Results in only doing lighting that matters
Recap - Deferred shading

- Improves speed and SIMD efficiency
- Introduces a bandwidth bottleneck
  - read/write for many textures
  - a lot of overdraw
Recap - Tiled deferred shading

- G-buffer like in deferred shading
- Divide the screen in tiles
  - Find min/max depth
  - Find lights influencing the tile
  - Shade pixels in the tile
Recap - Tiled deferred shading

- Has no overdraw and therefore less bandwidth usage
  - Lower bandwidth allows us to have many light sources and good frame rates

- Still has unnecessary work due to depth discontinuities
  - Performance is view dependant
Tiled forward shading

- Use a depth pre pass in first geometry render
- Find and store intersecting light sources for each tile
- Use light index buffer to do shading in a second geometry render
Clustered deferred shading

- From tiles to clusters (2D to 3D)
  - Make use of the depth to subdivide the frustum
  - Allows use of normal for backface culling of lights

- Solves depth discontinuity issues
  - Results in more stable performance
  - Is usually faster than tiled shading
Algorithm (deferred shading)

- Render scene to G-buffer
- Cluster assignment
- Determine unique clusters
- Assign lights to clusters
- Shade samples
Cluster Assignment

- Map pixels to clusters
  - compute an integer cluster key based on G-buffer

- Pixels that are close together are likely influenced by the same lights

- Subdivide the frustum
  - Easy to do, fast to create and predictable in size
  - Cluster key is index in the subdivision
  - Can also divide over normal direction
Cluster Assignment

- Subdivision is done by exponentially spacing the z-division
  - results in cube-like clusters
  - other divisions result in "non-uniform" division
Cluster assignment - Normal

- Use the normal for more detailed clusters
- Quantize normal direction
- Allow different grid sizes
Cluster assignment

- Example of clustering with and without normal
Finding unique clusters

- Find duplicates and remove them
  - Each cluster should only be considered once

- Use sorting
  - sort samples in the local screen space tile
  - compact samples to obtain unique clusters
  - store links from samples to its cluster

- Use page tables
  - Similar to virtual memory
Finding unique clusters

- Unique clusters have implicit 3D bounds
  - Also holds for normal if calculated

- May prefer tighter explicit bounds by finding min-max of samples
Light assignment

- Matching all lights with all clusters takes too long for many lights

- Create BVH for lights each frame
  - First sort lights based on Z-order
  - Group 32 elements into a higher level node

- Traverse BVH depth first per cluster
  - Test bounding box intersections
  - Use optional normal for rejecting lights
Shading

- Tiled shading uses screen space coordinates for light list lookup
- No direct mapping between cluster keys and light list index
- Sorting method stores index explicitly
- Page tables store these in the pages
Clustered forward shading

- Replace shading stage by second geometry render
- Same as tiled forward shading
Results

- Several implementations of clustering
- Comparison between deferred, tiled deferred and tiled forward with various clustered methods
- Necropolis scene with moving lights, about 2500 in total
- Sponza scene with trees added for depth discontinuities, 10K lights
Results

- Difference between methods in the necropolis scene
Results

- Comparison of the methods in the sponza scene
Conclusions & questions

- Clustered shading performs well with reduced view dependence
- Clustered shading scales better with more lights than tiled shading