Clustered Deferred and Forward Shading

Clustered Shading

Ola Olsson, Markus Billeter and Ulf Assarsson
Chalmers University of Technology

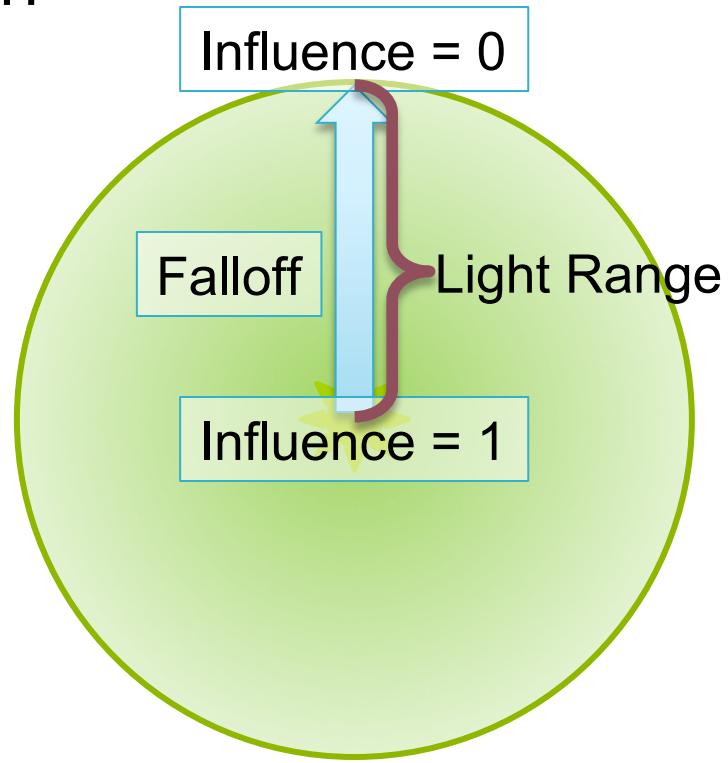
CHALMERS

Presentation Roadmap

- Brief summary of properties
- Tiled Shading recap
- Tiled Shading Problems
- Clustered Shading
  - Algorithm
  - Results

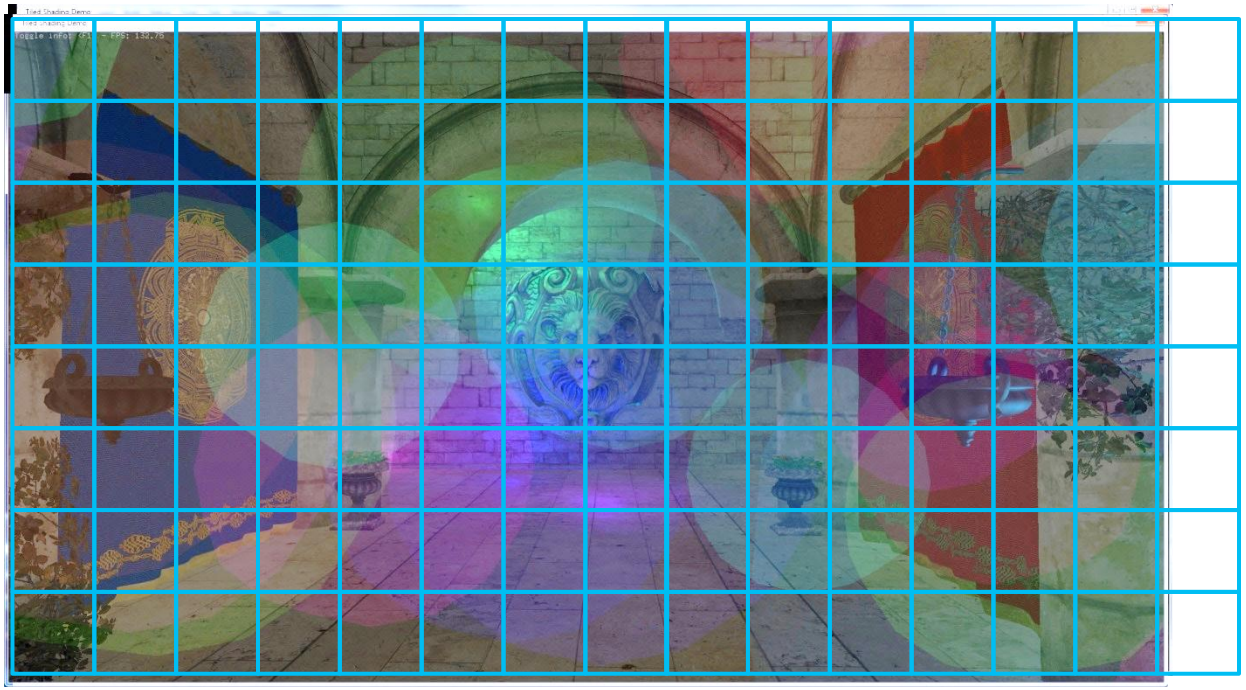
Clustered Shading -What does it do?

- Real-time shading algorithm
  - Thousands of lights
    - Limited range light
    - No shadows
  - Fully dynamic
    - Lights and Geometry
- Robust performance
  - Low overhead
  - Low view-dependence
  - Scales to 1M lights.
  - Handles noisy depth distributions



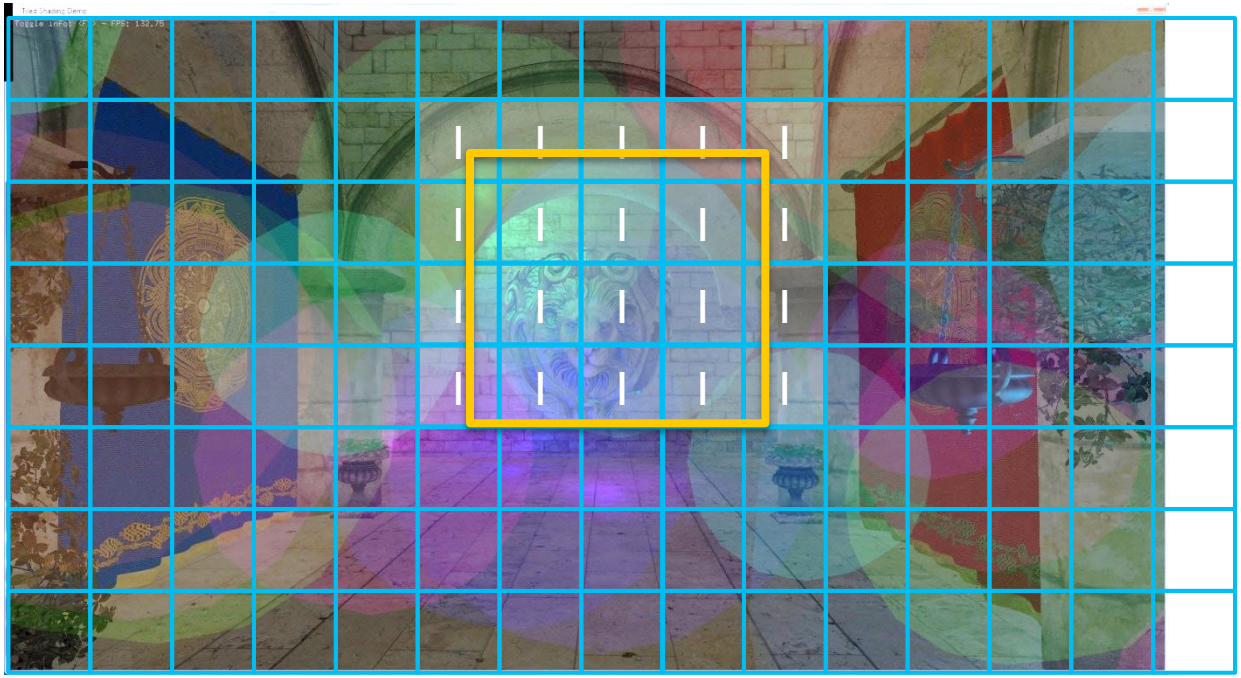
Tiled Shading Recap -The Light Grid

- Screen Space Tiles
  - E.g. 32x32 pixels
  - Each contains list of lights
- For each light
  - Find screen space AABB
  - Add to tiles



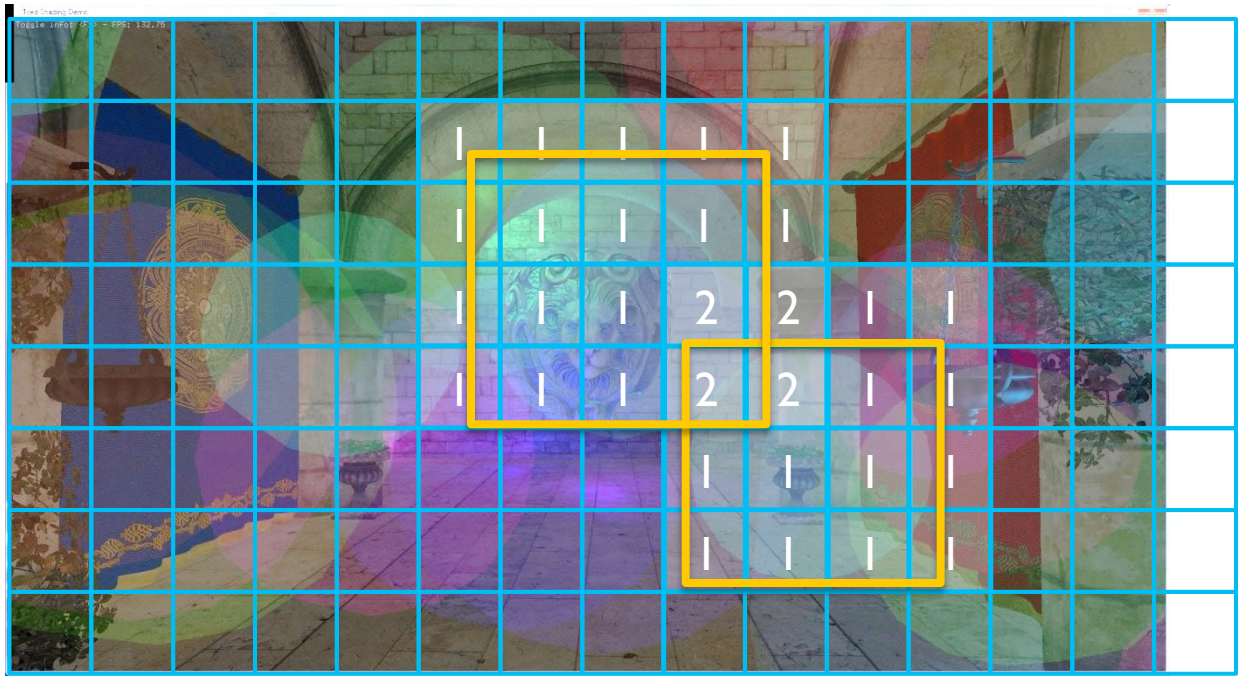
Tiled Shading Recap -The Light Grid

- Screen Space Tiles
  - E.g. 32x32 pixels
  - Each contains list of lights
- For each light
  - Find screen space AABB
  - Add to tiles



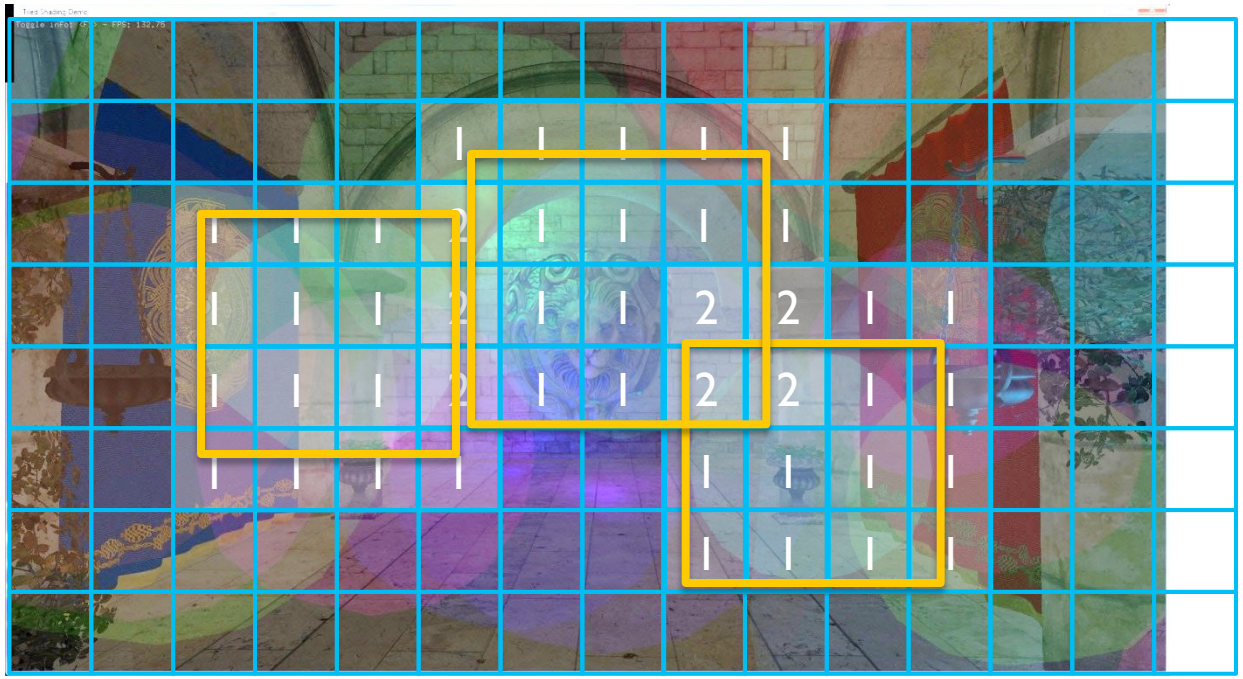
Tiled Shading Recap -The Light Grid

- Screen Space Tiles
  - E.g. 32x32 pixels
  - Each contains list of lights
- For each light
  - Find screen space AABB
  - Add to tiles



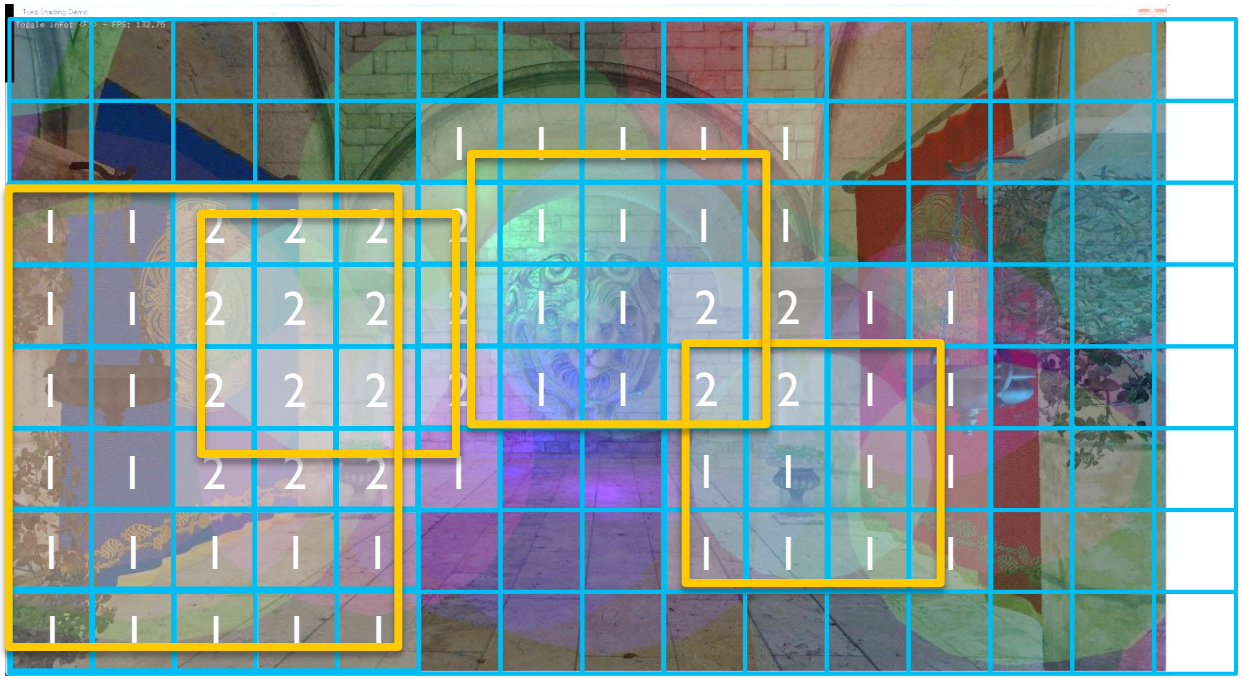
Tiled Shading Recap -The Light Grid

- Screen Space Tiles
  - E.g. 32x32 pixels
  - Each contains list of lights
- For each light
  - Find screen space AABB
  - Add to tiles



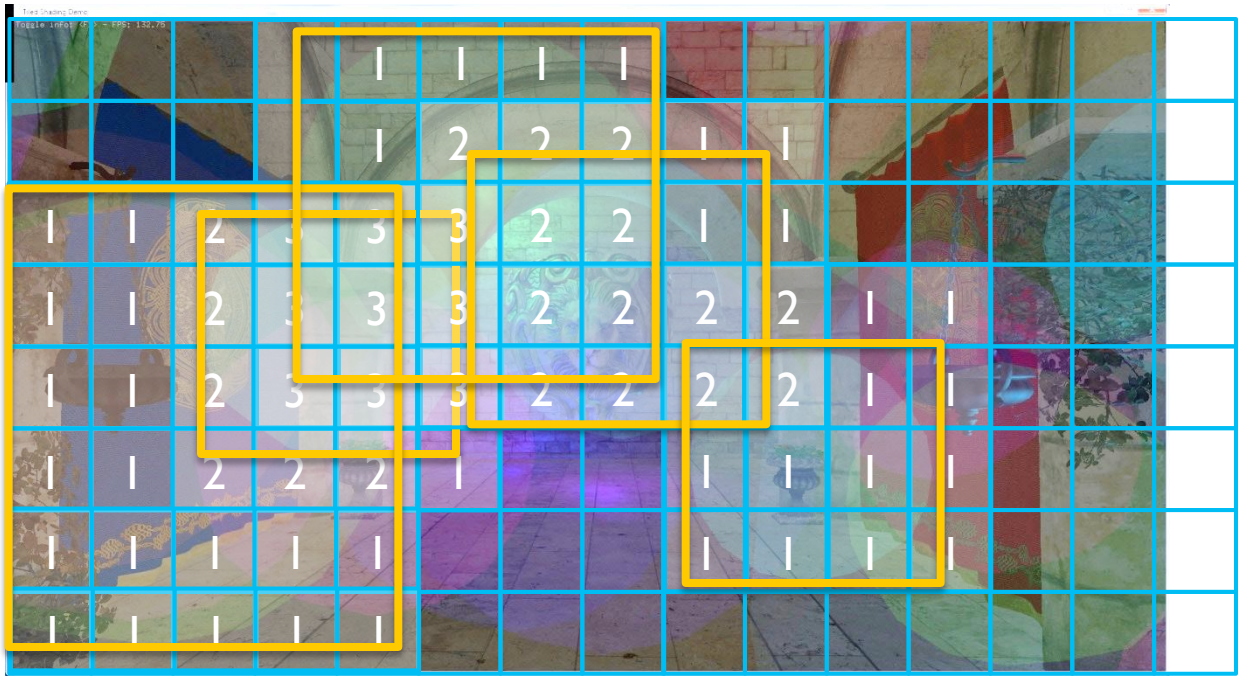
Tiled Shading Recap -The Light Grid

- Screen Space Tiles
  - E.g. 32x32 pixels
  - Each contains list of lights
- For each light
  - Find screen space AABB
  - Add to tiles

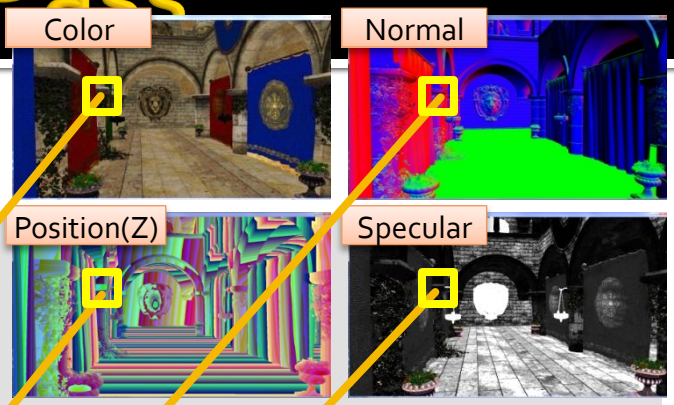


Tiled Shading Recap -The Light Grid

- Screen Space Tiles
  - E.g. 32x32 pixels
  - Each contains list of lights
- For each light
  - Find screen space AABB
  - Add to tiles

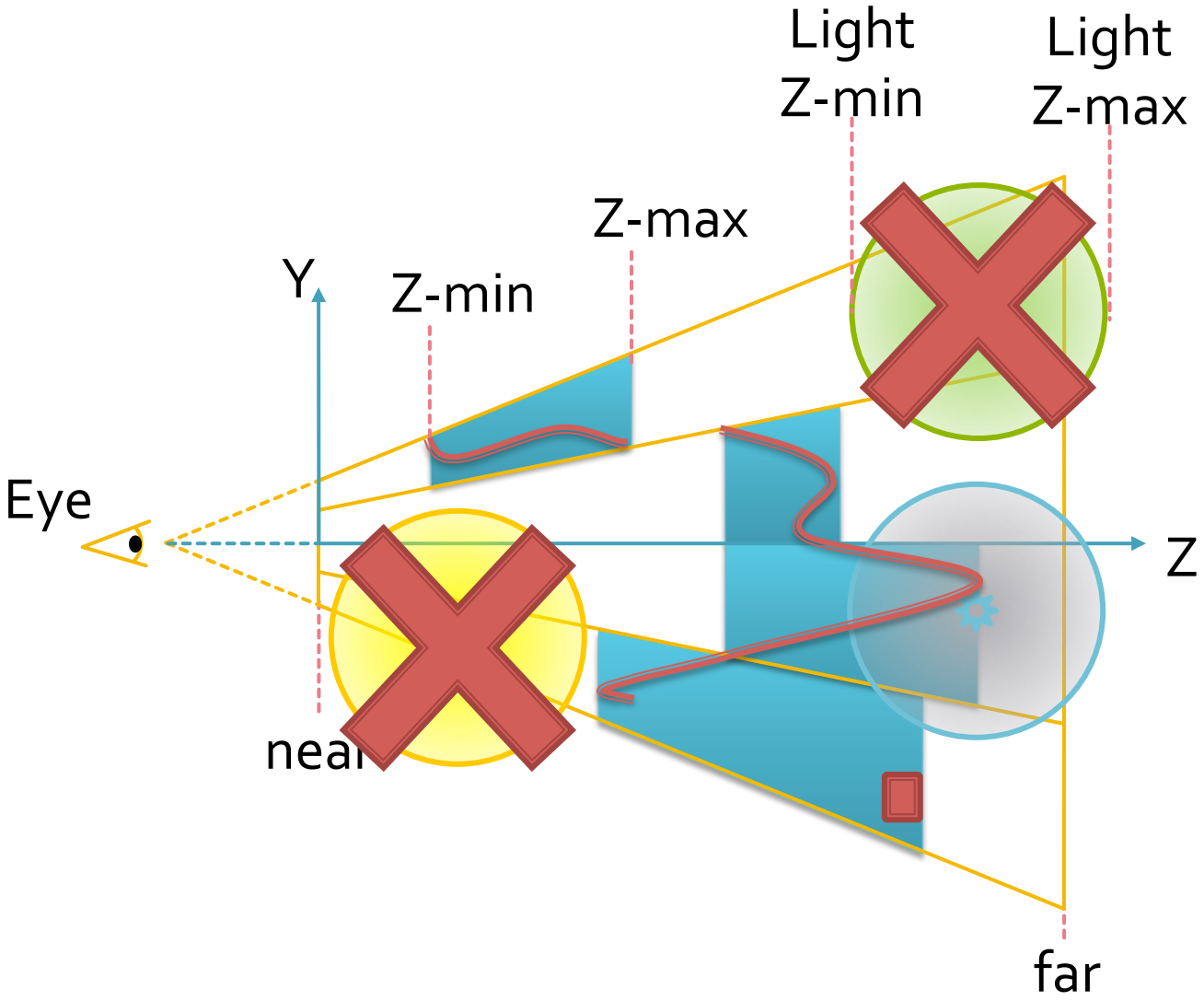


Tiled Shading Recap
- Full Screen Shading Pass



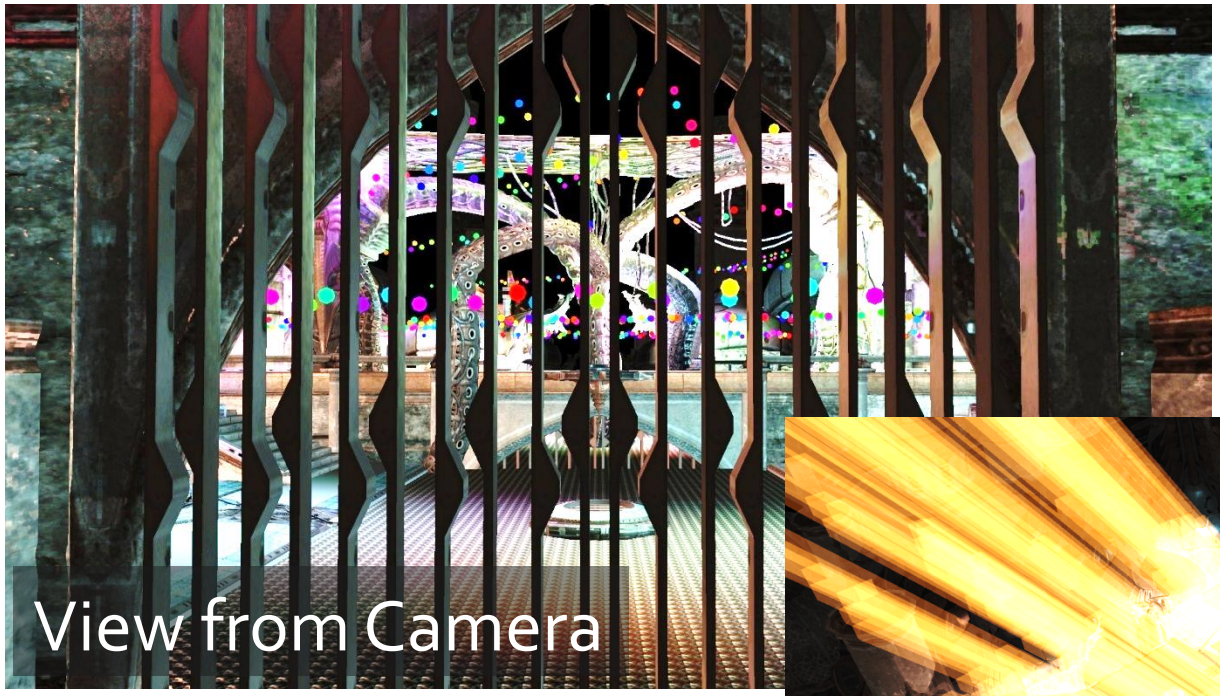

```
out vec4 resultColor;

void main()
{
  vec3 color = texelFetch(colorTex, gl_FragCoord);
  vec3 specular= texelFetch(specularTex, gl_FragCoord);
  vec3 normal = texelFetch(normalTex, gl_FragCoord);
  vec3 position = fetchPosition(gl_FragCoord);

  vec3 shading = accumulate for each light in tile;

  resultColor = vec4(shading, 1.0);
}
```

Tiled Shading Recap
-Depth Range Optimization

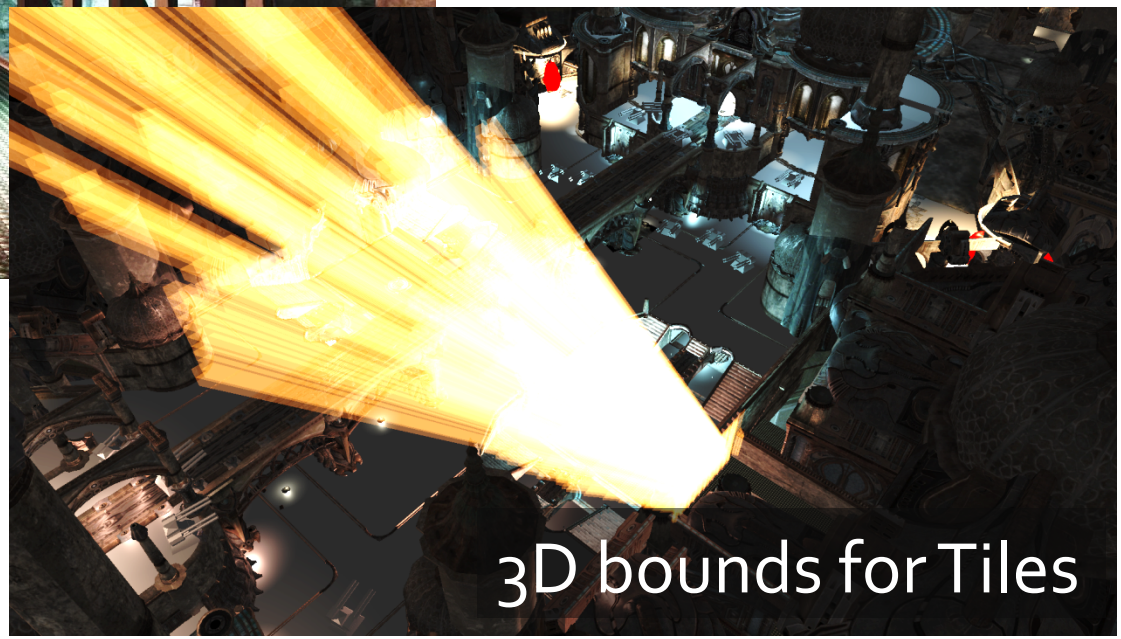


Tiles In 3D® - the movie™
UDK Necropolis CTF

Tiled Shading
-The Discontinuity Dysfunction



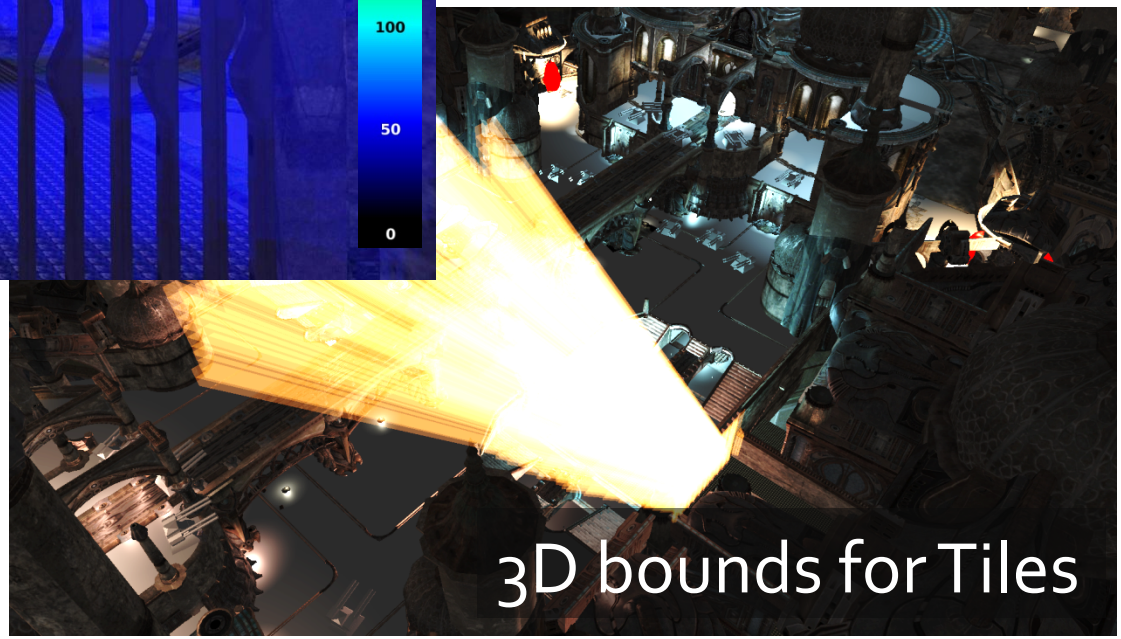
- - Unreal 3 Scene
  - 'Necropolis'
- - Tricky view
  - But not artificial



- 2D tiles with depth bounds
- High light density

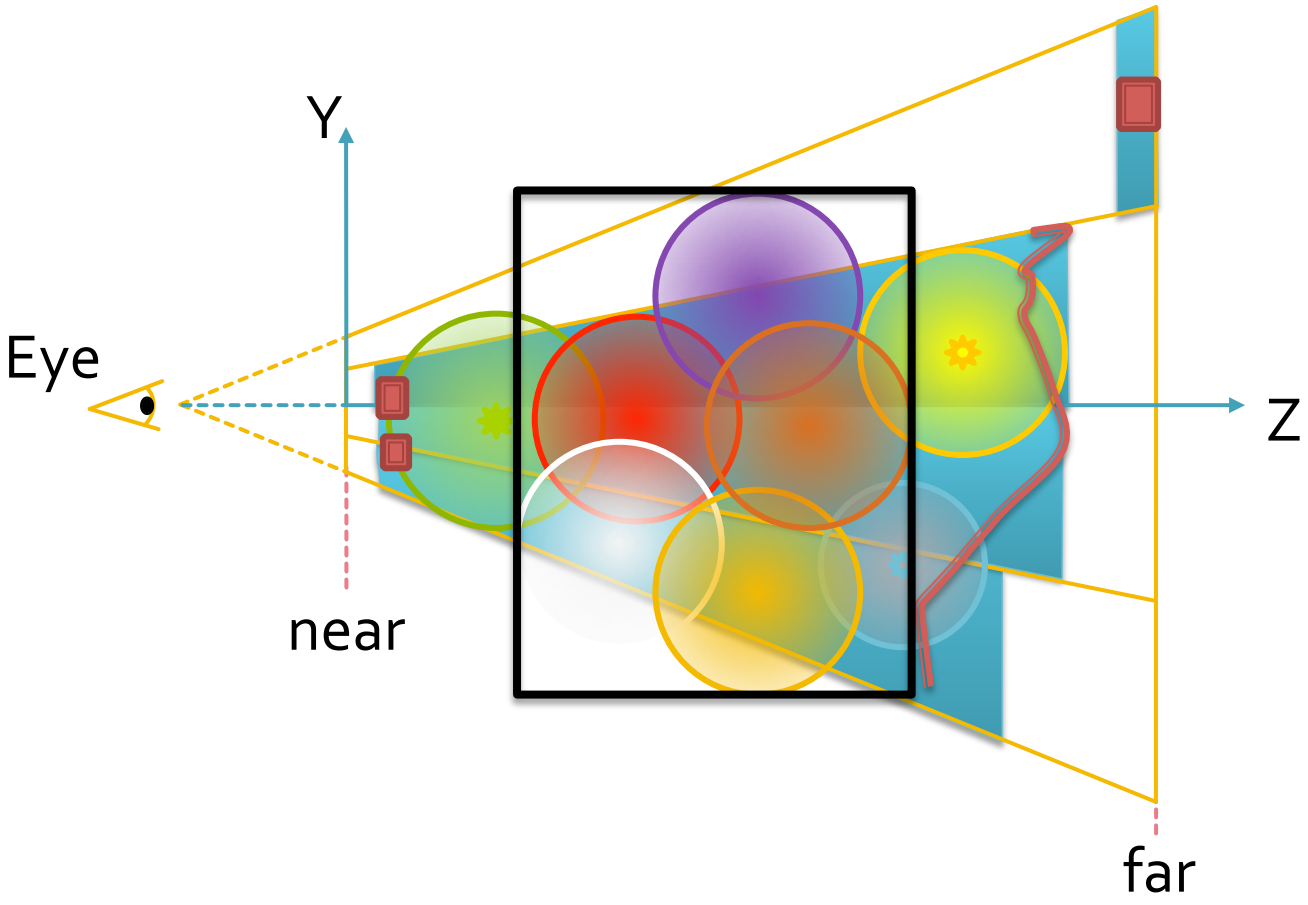
Tiled Shading
-The Discontinuity Dysfunction

- Black = zero
- Green ~ 150
- White > 300

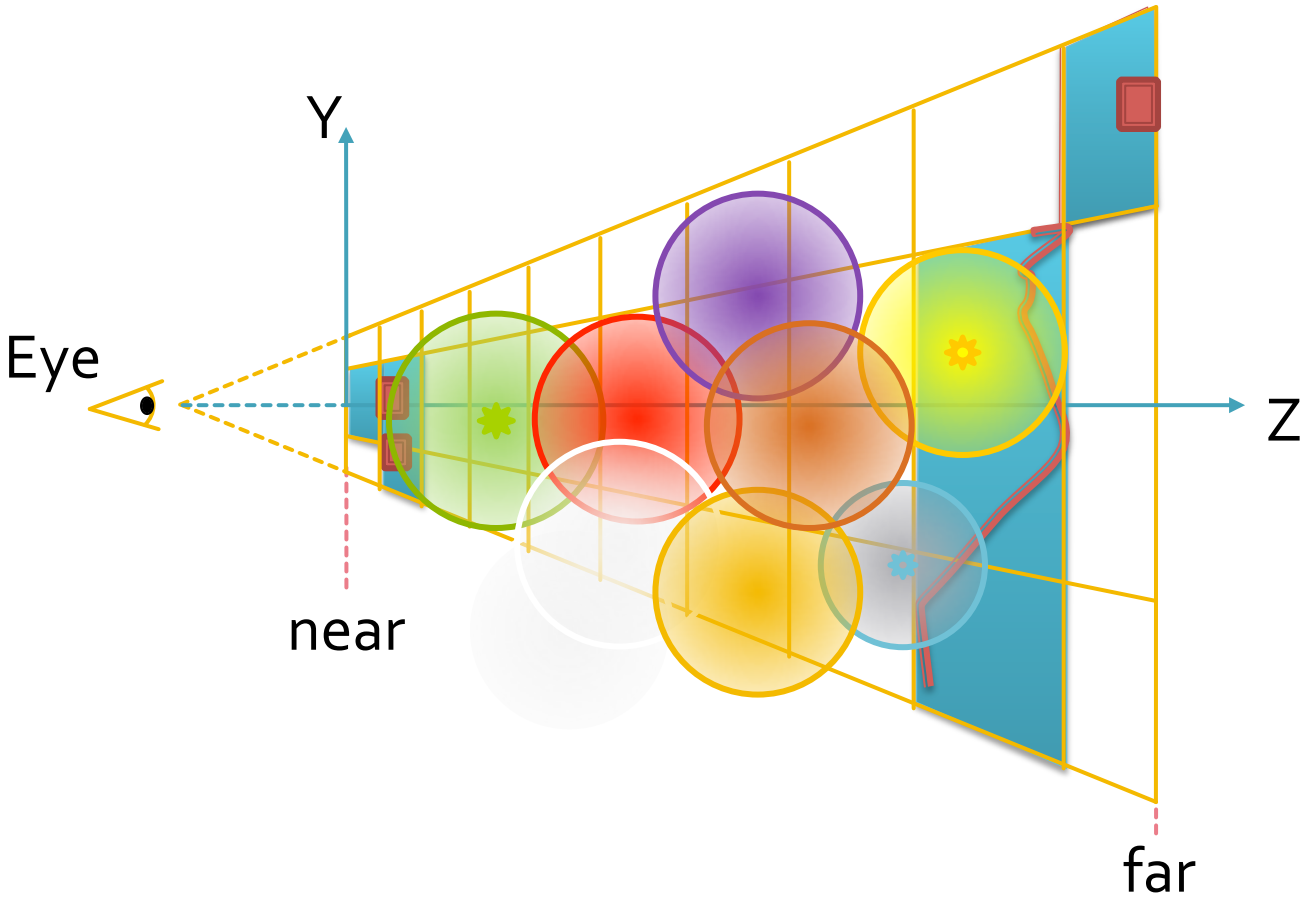


- Tiled: 18.8ms
- Clustered: 9.4 ms

Tiled Shading
-The Discontinuity Dysfunction



Clustered Shading
- Our solution



Clustered Shading -Idea

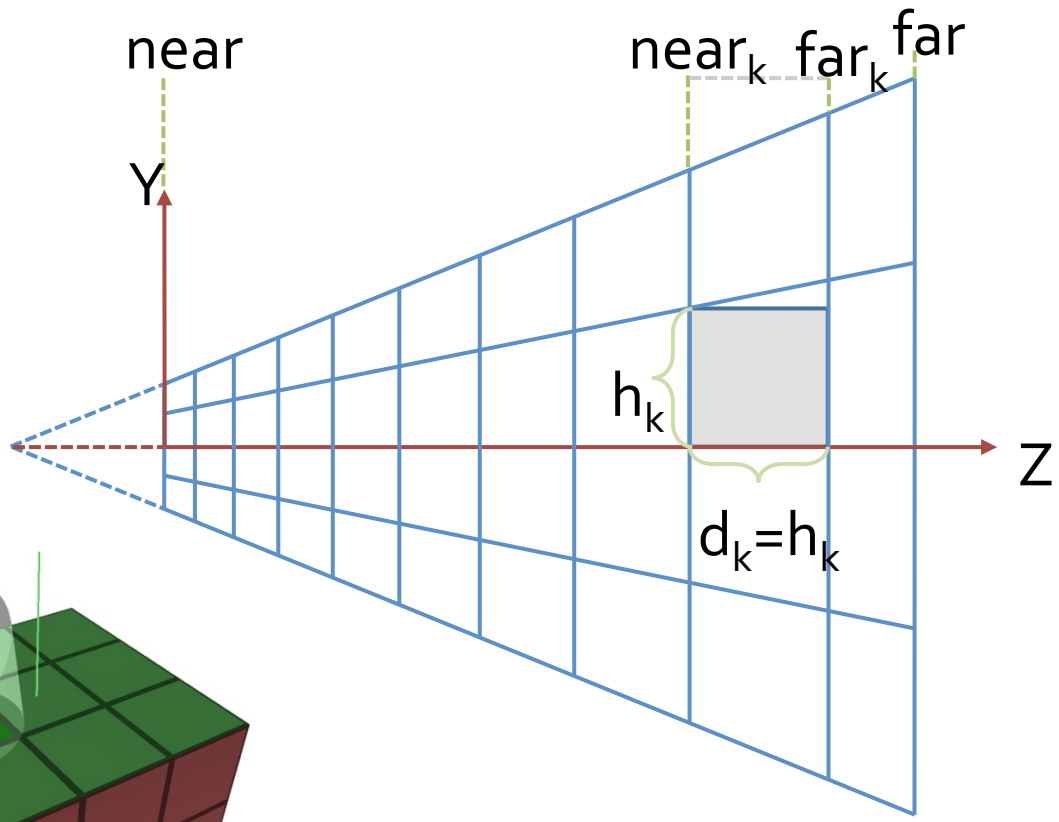
- Add the 3rd dimension
  - Tile also in depth direction = cluster
  - Also > 3 dimensions (e.g. normals)
- Bounded volume around samples
  - Shading cost ~ Light density.
- New Challenges
  - Many more (potential) clusters
    - Must find those actually used
  - Adding lights no longer screen space

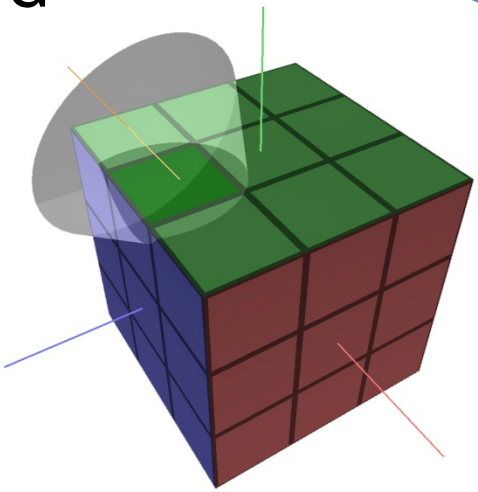
Clustered Shading -Algorithm

1. Rasterize G-Buffers
   - (Forward: pre-z pass)
2. Cluster assignment
3. Find unique clusters
4. Assign lights to clusters
5. Shade view samples

Cluster Assignment

- Cluster key:
  - (*i*,*j*,*k*,*n*)
- *i*,*j* – 2D tile id
- *k* – log(view z)
- *n* – quantized normal



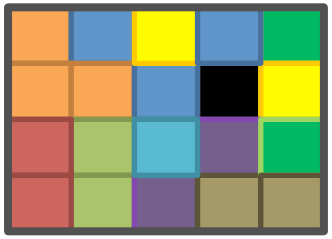


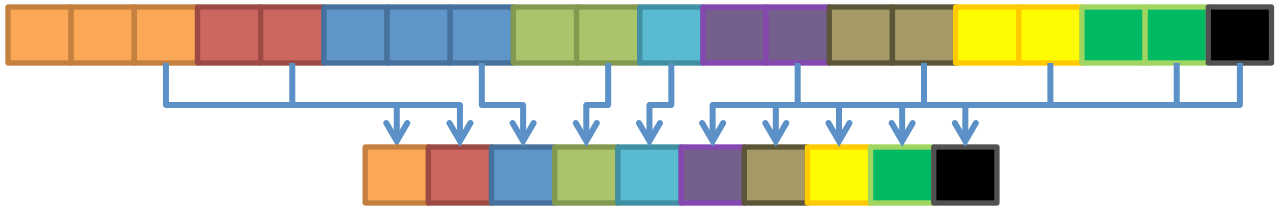
Finding Unique Clusters

- In 2D just use full grid
- With 3D, too many potential cells
  - Especially with normal
  - E.g. $\underbrace{60 \times 34}_{i \times j} \times \underbrace{300}_{k} \times \underbrace{6 \times 3 \times 3}_{n} = 31M$
- Two approaches tested
  - Sorting tiles locally
  - Global page table
    - *Virtual* Grid

Finding Unique Clusters - Tile Sorting

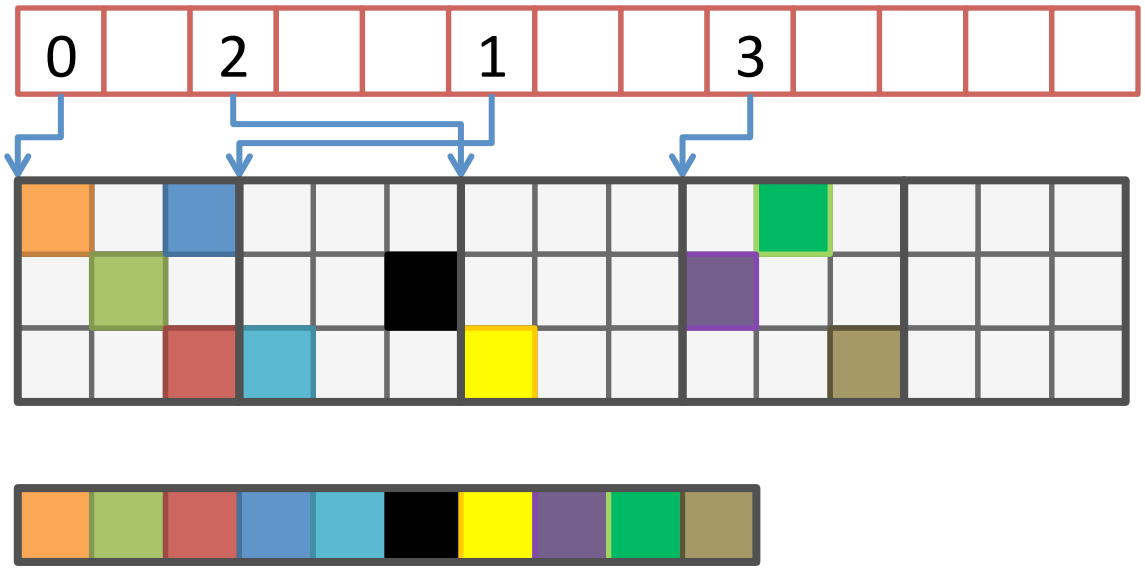
- Local Tile sorting
  - Shared memory
- Global prefix
  - small
- Meta data reduction
  - Normal Cones
  - Aabbs





Finding Unique Clusters - Page Tables

- *Virtual* Grid (or range)
- Very Quick
- 2 Passes
  - Fermi
- 1 Pass
  - Kepler



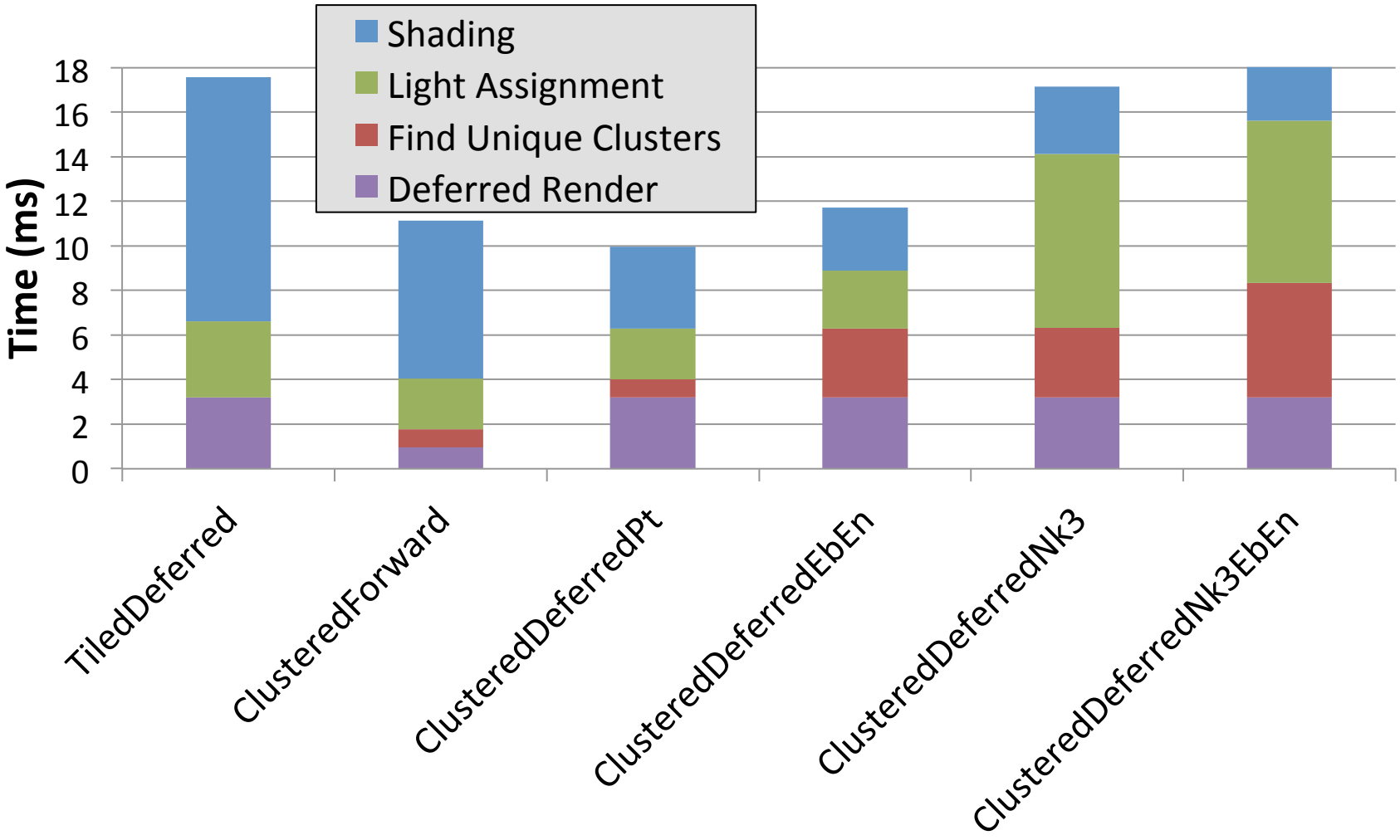
Light Assignment

- More clusters
- More lights
- Hierarchical approach
  - Hierarchy over lights
  - Also possible
    - Hierarchy over clusters
    - Maybe better
  - Or Both
    - Probably best…

Results

- Crytek Sponza
  - +Trees
  - 10k Lights

Results


Shading
Light Assignment
Find Unique Clusters
Deferred Render
Time (ms)
0
2
4
6
8
10
12
14
16
18
TiledDeferred
ClusteredForward
ClusteredDeferredPt
ClusteredDeferredEbEn
ClusteredDeferredNk3
ClusteredDeferredNk3EbEn

Results Animation

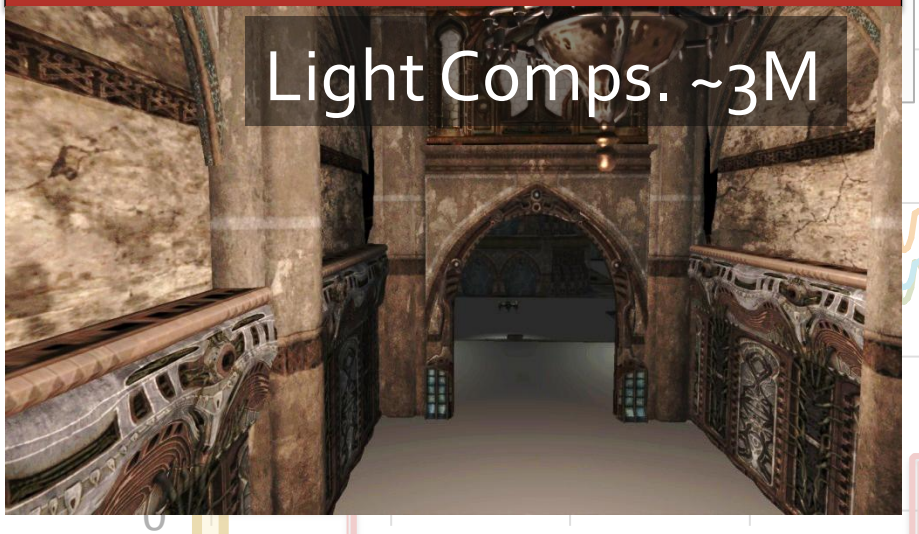
- UDK Necropolis
  - Real game scene (no trees added)
- ~2M polys
- Normal maps
- ~2500 lights
  - In scene ~650 lights
  - Canons adding ~ 1800 more
- Animation on your USB stick

Results

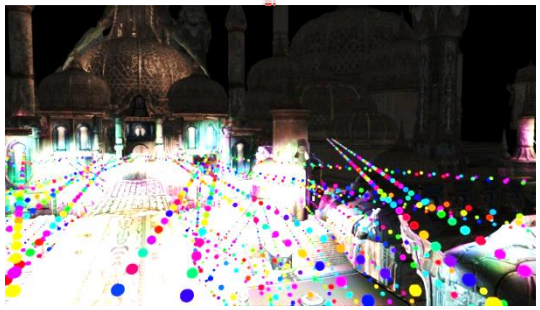

TiledDeferred
ClusteredForward
ClusteredDeferred
Time (ms)
0
5
10
15
20
0
50
100
150
200
250
300
350
400

Results

Tiled Shading

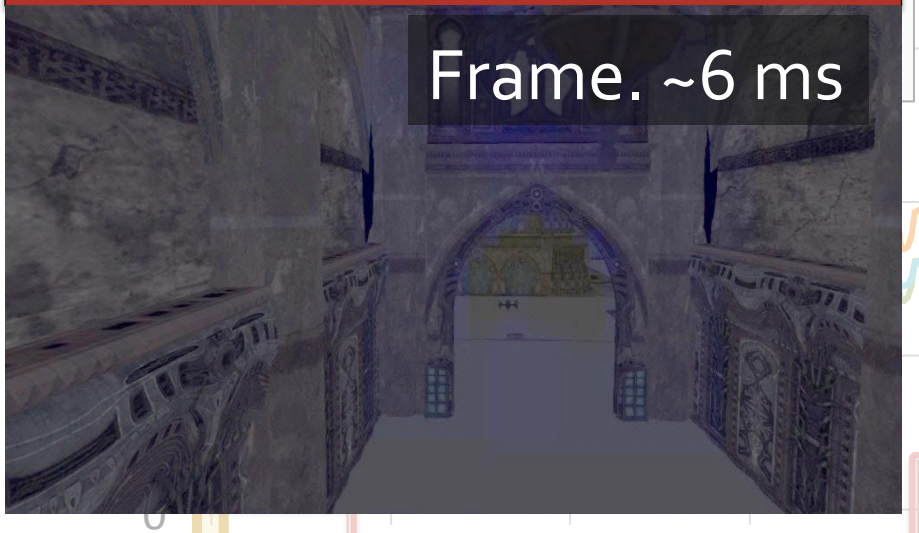


Clustered Shading

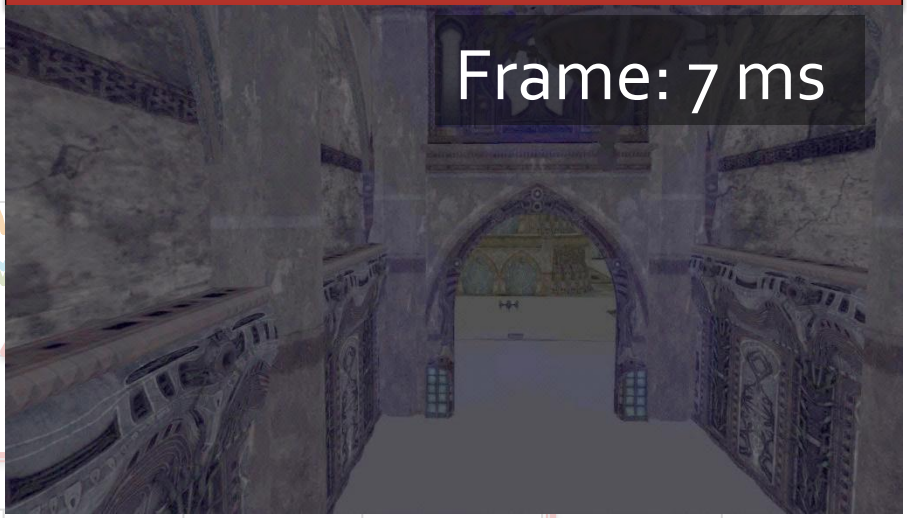


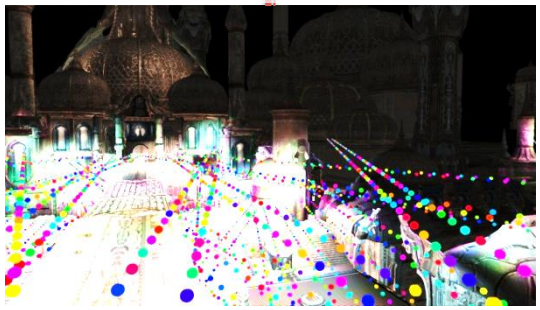
Results

Tiled Shading



Clustered Shading



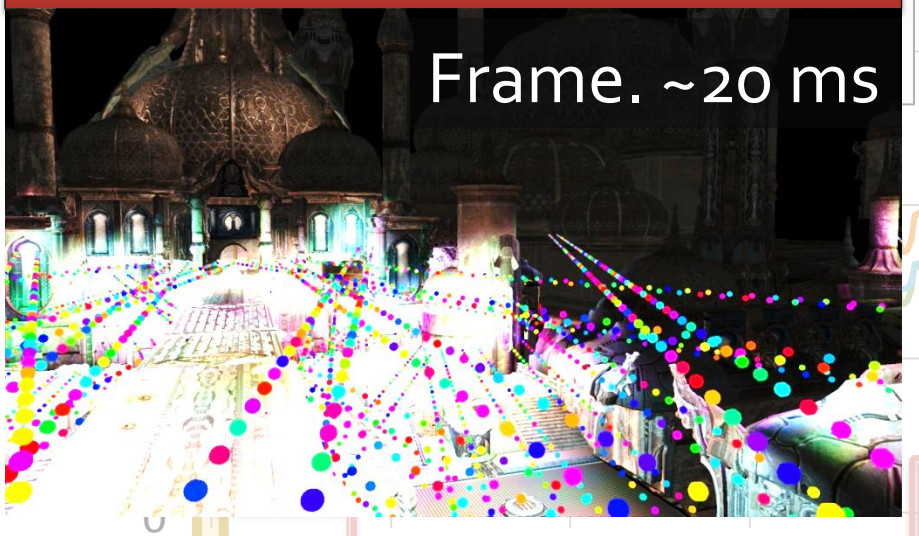


Results

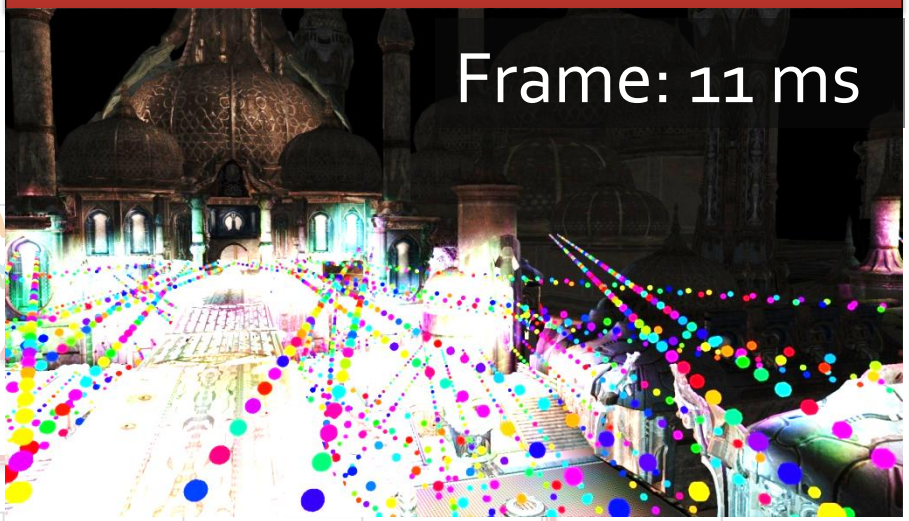

TiledDeferred
ClusteredForward
ClusteredDeferred
Time (ms)
0
5
10
15
20
0
50
100
150
200
250
300
350
400

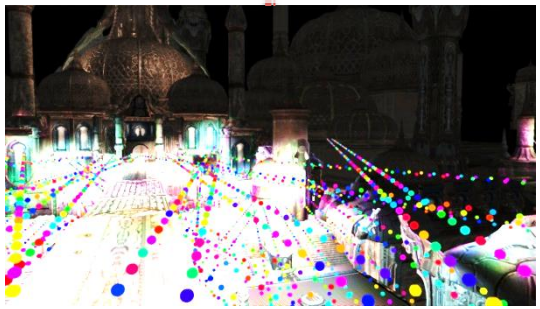
Results

Tiled Shading

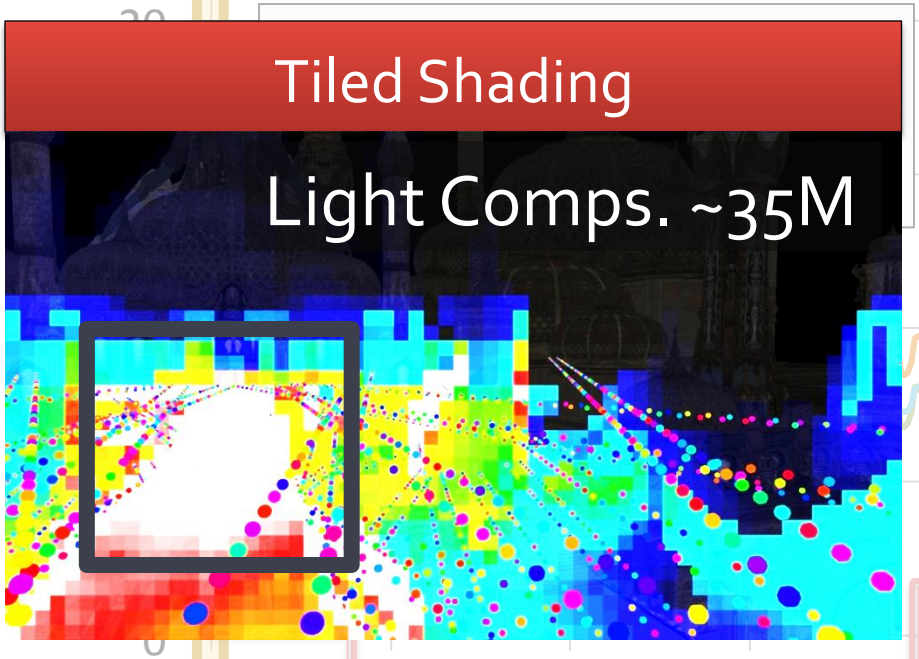


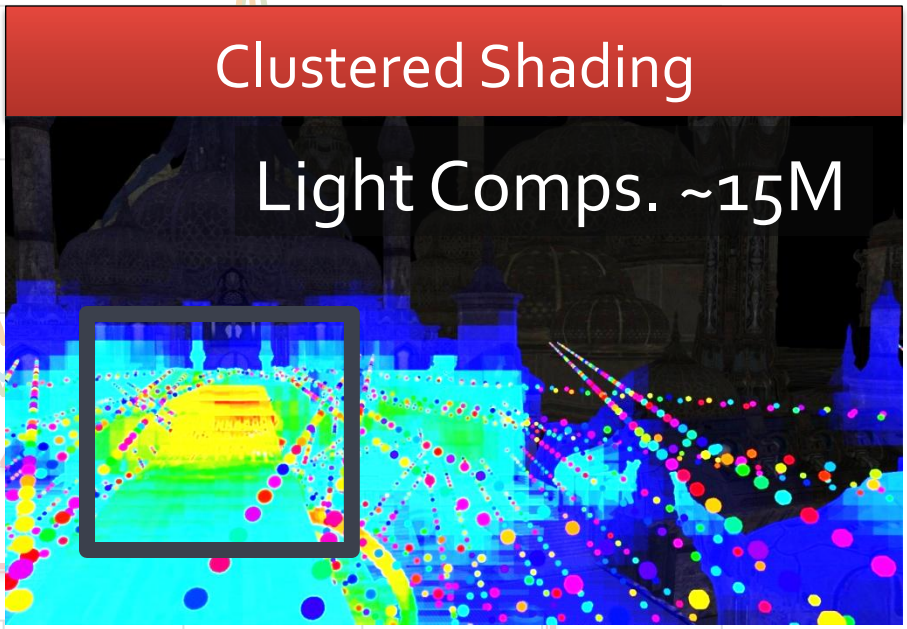
Clustered Shading

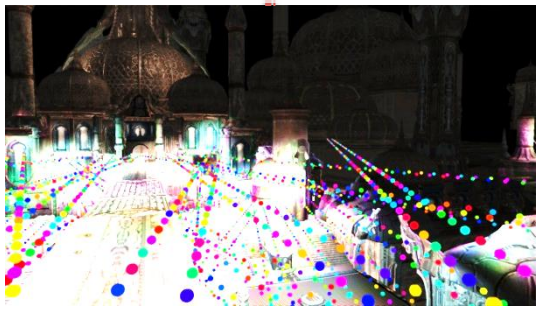




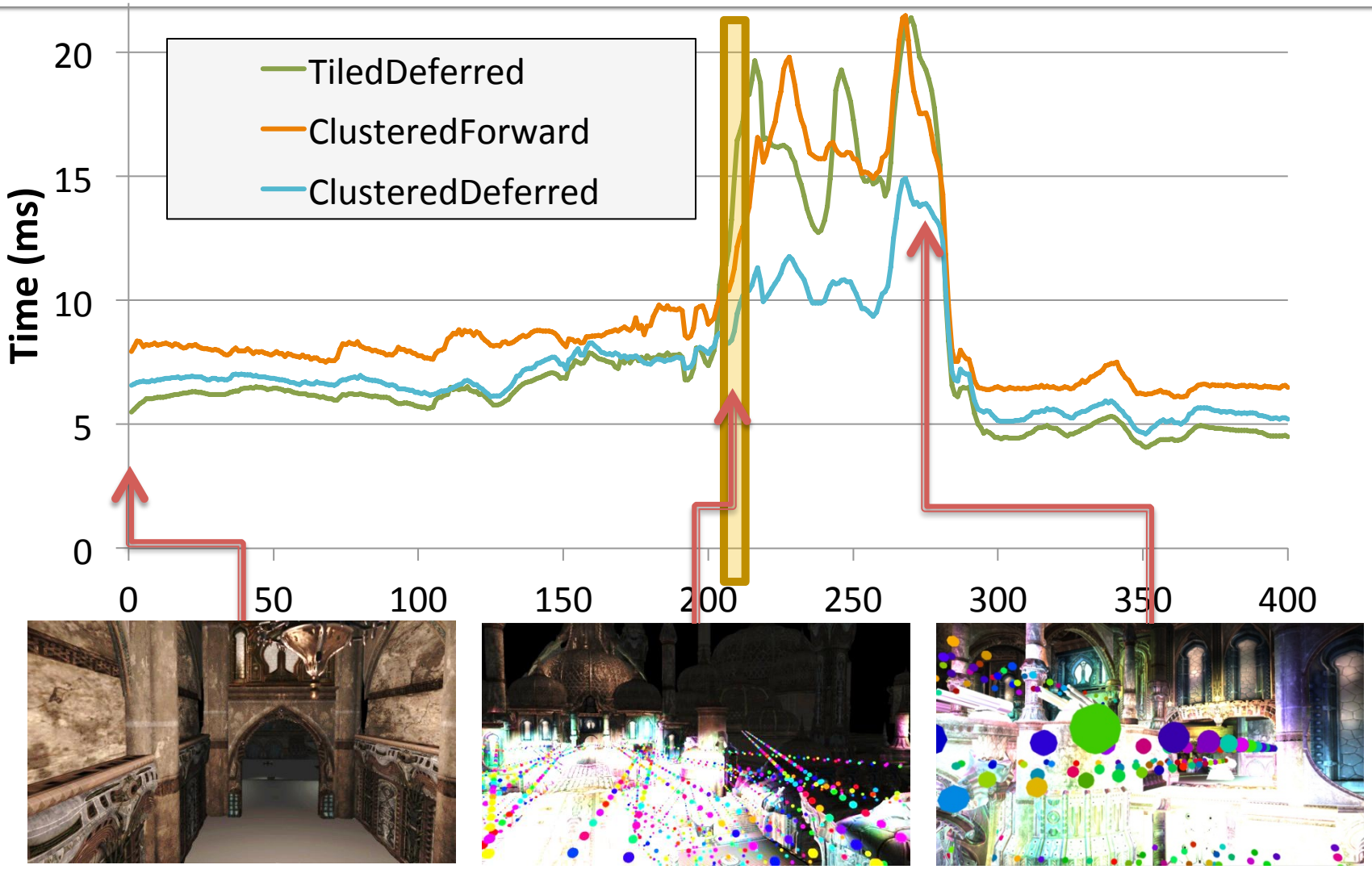
Results


Tiled Shading
Light Comps. ~35M


Clustered Shading
Light Comps. ~15M

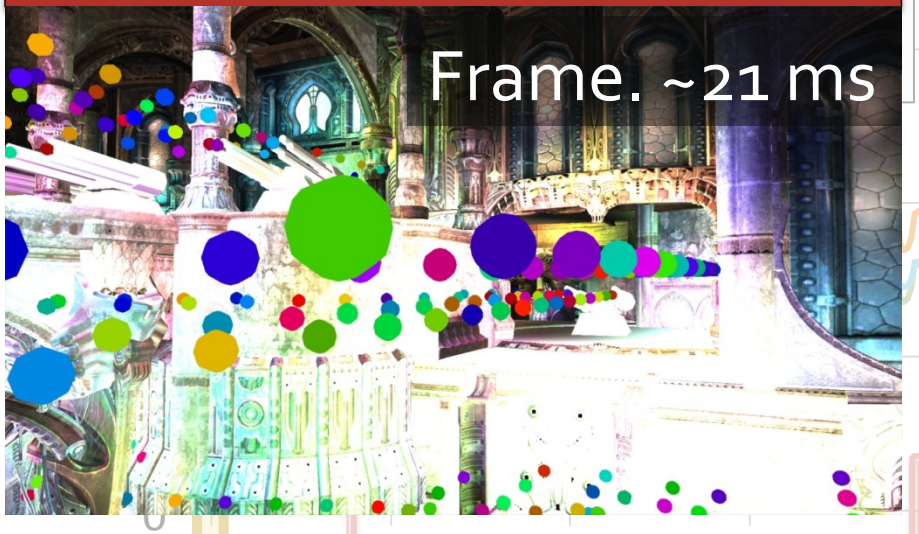


Results

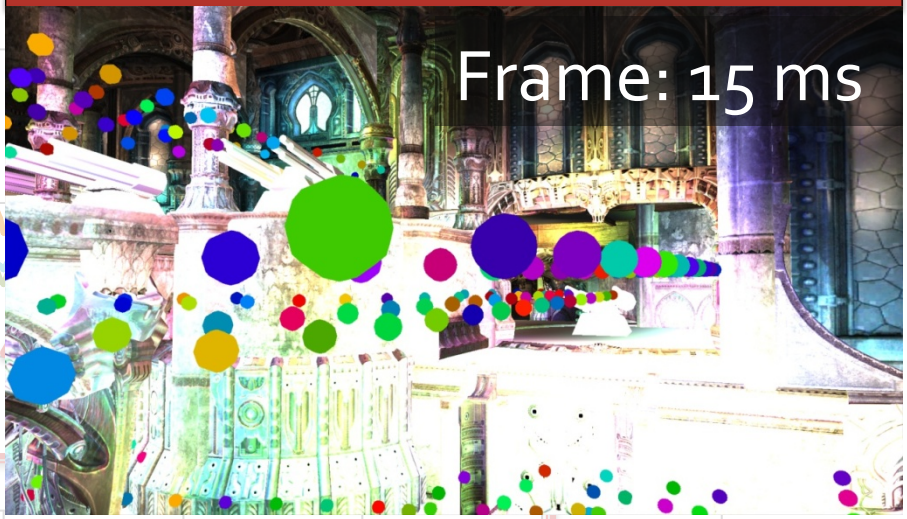

TiledDeferred
ClusteredForward
ClusteredDeferred
Time (ms)
0
5
10
15
20
0
50
100
150
200
250
300
350
400

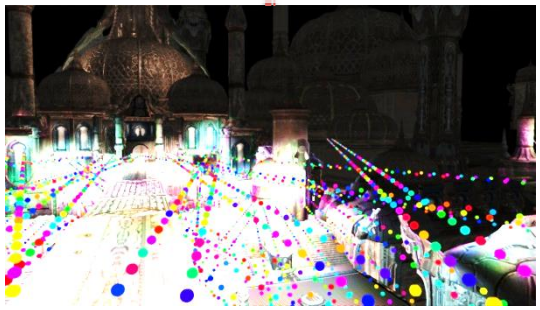
Results

Tiled Shading

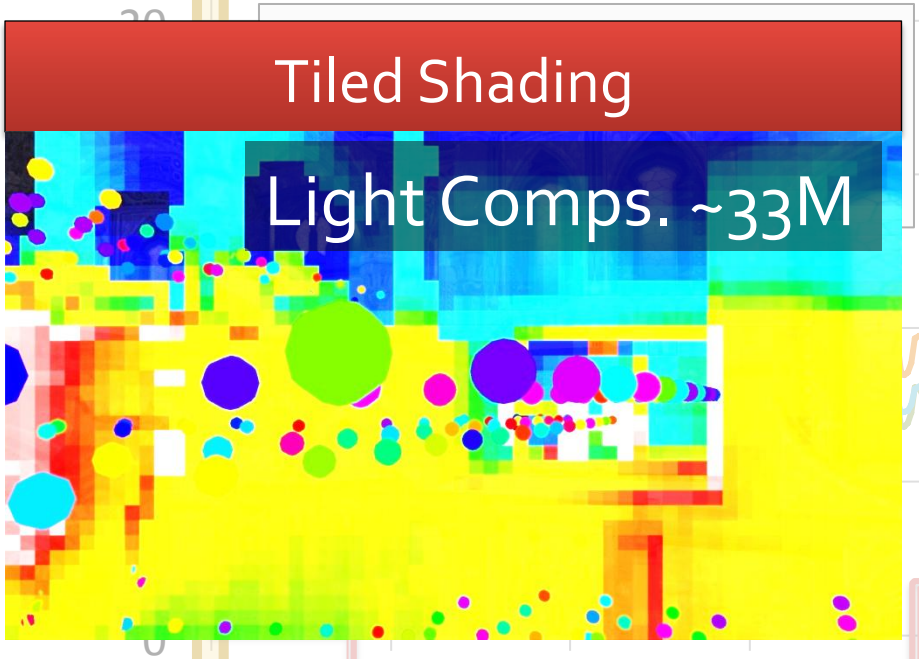


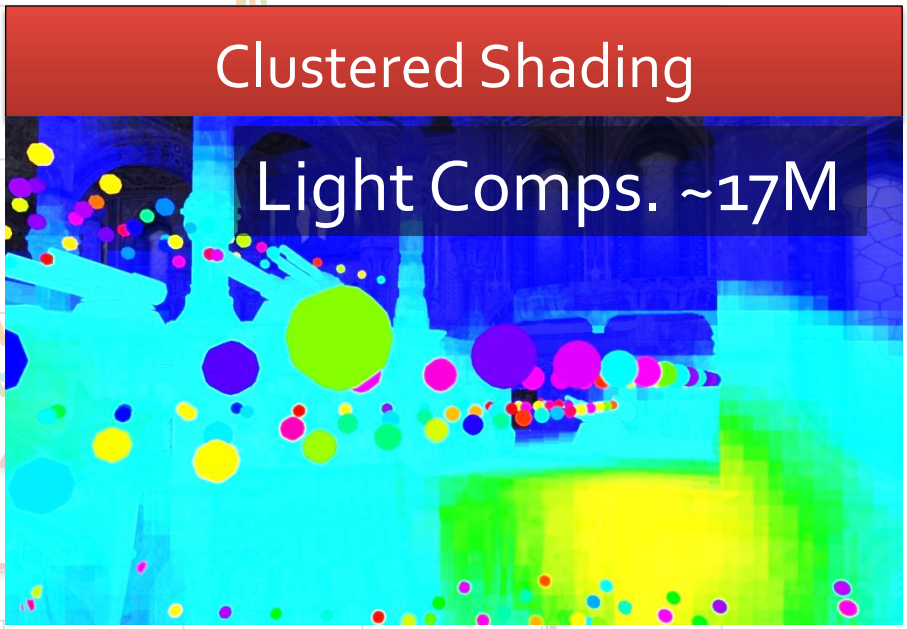
Clustered Shading

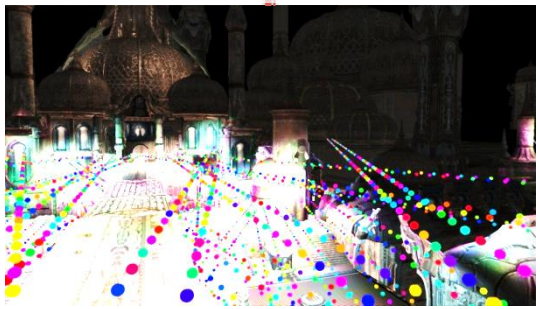




Results


Tiled Shading
Light Comps. ~33M


Clustered Shading
Light Comps. ~17M



Results


TiledDeferred
ClusteredForward
ClusteredDeferred
Time (ms)
0
5
10
15
20
0
50
100
150
200
250
300
350
400

Thanks

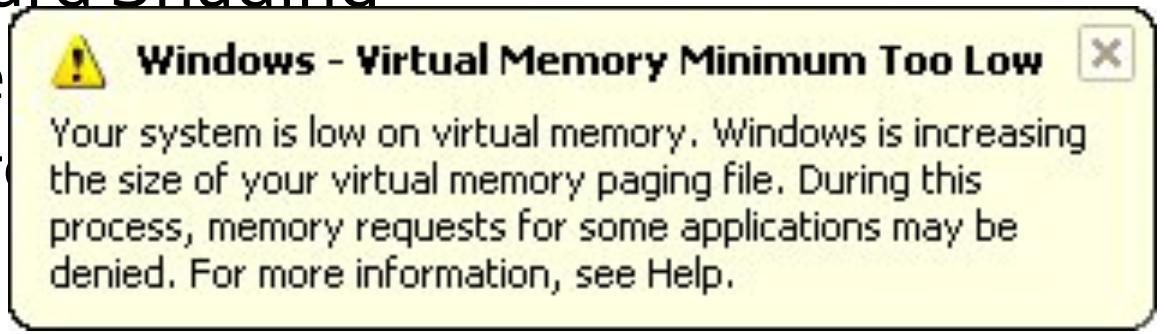
- Questions?
- Demo implementation available soon
  - Well, perhaps not until after summer…

`http://www.cse.chalmers.se/~olaolss`

Culled Slides…

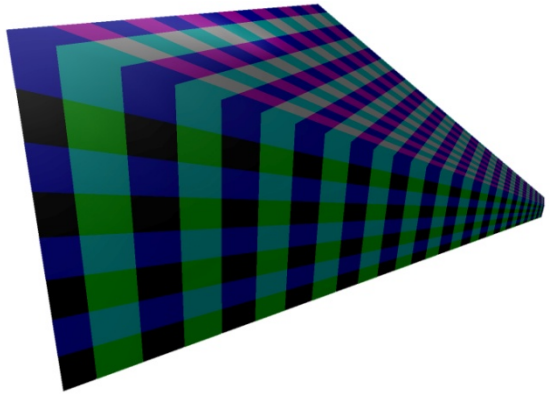
Forward Shading

- Motivation
  - MSAA G-Buffers (each):
    1920 x 1080 x 8 x 16 = 250Mb
  - Custom Shaders.
  - Transparency.
- Tiled Forward Shading
  - ~ 4x vs. Clustered Deferred (necropolis)
- Clustered Forward Shading
  - ~ 2x vs. Clustere
  - Significant impr

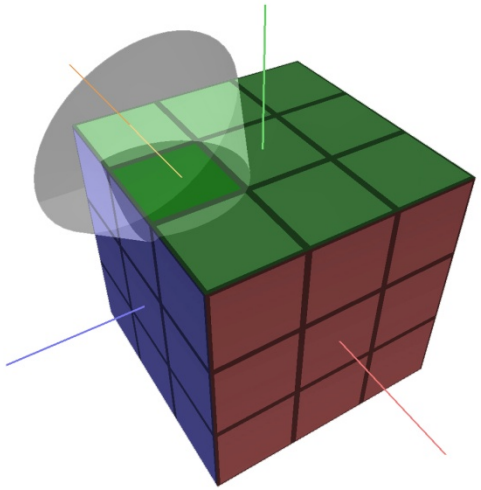


Implicit Cluster Bounds

- Index gives
  - Sub-frustum



- Quantized normal gives
  - Cone
  - Quantized, e.g. 6x3x3



Explicit Cluster Bounds

- Cone for normals
- Aabb for positions
- Extra cost for construction
  - Not always offset
- Not easy with page tables.
  - Lots of atomics with collisions