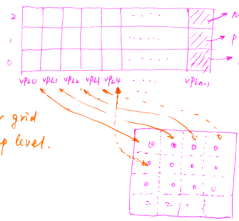
Stencil-Based Multiresolution Splatting [1]

1. Generate RSM

- Frame buffer: [width x height], 3 render Targets: [0] albedo [1] normal [2] position
  - albedo → mip-map
  - normal, position → light space
- could use lower res model

2. Extract VPLs

- frame buffer: [#VPLs x 3]
- compact VPLs in a [#VPLs x 3] map
- no quasi-random sampling & importance sampling, just grab the VPLs along a regular grid from the RSM at proper mip-map level.



3. G-Buffer

- frame buffer: [width x height]

4. Calc depth derivative of G-Buffer

- frame buffer: [width x height]
- 3x3 kernel
- out: depth deviation

out = max{depth values in the 3x3 kernel} − min{depth values in the 3x3 kernel}

5. Custom Mip-Map generation: max depth deviation Mip-Map

- 3x3 kernel
- out: max depth deviation
- down sampling

out = max{depth deviations in the 3x3 kernel}

6. Create Normal threshold buffer

- 3x3 kernel
- out: [x | y | z | w]
  - x, y: max $\vec{N}$.xy
  - z, w: min $\vec{N}$.xy

out.xy = max{$\vec{N}$.xy in the 3x3 kernel}

out.zw = min{$\vec{N}$.xy in the 3x3 kernel}

7. Custom Mip-Map generation: min-max normal Mip-Map

- 3x3 kernel
- out: [x | y | z | w]
  - x, y: max $\vec{N}$.xy
  - z, w: min $\vec{N}$.xy

out.xy = max{$\vec{N}$.xy in the 3x3 kernel}

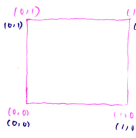
out.zw = min{$\vec{N}$.xy in the 3x3 kernel}

(the same as step 6)

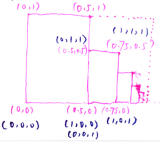
* Post process & Multi-Res post process

① post process

(x,y) - screen pos
(x,y) - tex coord
(x,y,z) - 3D tex coord



② Multi res post process



8 Generate Multi-resolution subplanes [Multi-res post-process]

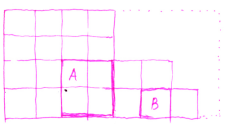
- frame buffer: [2 x width, x height]
- only set the stencil.
- a fragment with texcoord (x, y, i) is set ~~if~~ { not discontinuous (x, y, i) / discontinuous (x, y, i+1)

discontinuity detection:

```
bool discontinuous (float x, float y, int level) {
    float depthDeriv = texture2DLod (MaxDepthDeriv, float2(x,y), level).r;
    if (depthDeriv > ε) return true;
    float4 minMaxNorm = texture2DLod (MinMaxNorm, float2(x,y), level);
    float normDeriv = max( minMaxNorm.x - minMaxNorm.z, minMaxNorm.y - minMaxNorm.w);
    return normDeriv > δ;
}
```

Example:

if not discontinuous (A)
and discontinuous (B)
then set A's stencil
else discard A



multi-res indirect light buffer

9 Gathering indirect lighting from VPLs [Multi-res post process]

```
foreach frag in stenciled fragments:
    foreach l in VPLs:
        do_lighting (l, frag)
```

10 Upsampling

- accumulate the indirect lighting in multi-res indirect light buffer
- the total indirect lighting is stored in Level 0.

11 LightingPass (direct lighting)

12 Final Shading

- shade the scene with direct lighting & indirect lighting.