MSAA-Based Coarse Shading for Power-Efficient Rendering on High Pixel-Density Displays
Pavlos Mavridis  Georgios Papaioannou
Department of Informatics, Athens University of Economics and Business

1. Motivation
Maintaining real-time frame rates at the native resolution of high pixel-density displays is very challenging, especially on power-constrained mobile platforms. Decoupled sampling methods offer a solution by reusing the same shading color across multiple visibility samples, but this ability is rather limited in modern GPUs, where the widely-used MSAA algorithm shades each covered primitive at least once per-pixel, without directly supporting more coarse shading. While various extensions of the graphics pipeline for coarse shading have been proposed [1][2], in this poster we focus on a software implementation for existing GPUs.

2. Contributions
• We present an MSAA-based method that reduces the total number of fragment shader invocations by shading more coarsely pixel blocks that are covered by the same primitive.
• We measure a large reduction of power consumption compared to standard per-pixel shading in a number of scenes. Our forward-rendering approach relies on standard APIs and HW capabilities and is suitable for current GPUs (OpenGL ES 3.1).

3. Method Overview

Step 1. Render at a lower pixel resolution, but with additional MSAA samples to guarantee at least one sample per screen-pixel.
Step 2. A custom resolve shader performs the mapping of sub-pixel MSAA samples to screen pixels.

Centroid Sampling should be used when rendering the intermediate render buffer.

4. Interleaved Sampling
Spatial Interleaving. Adjacent pixels have different sampling patterns. Masks aliasing artifacts at polygon edges with noise. Can be disabled using programmable MSAA sample positions.
Temporal Interleaving. Alternate between two sampling patterns at successive frames. Requires 60+Hz rendering and v-sync.

5. Results

<table>
<thead>
<tr>
<th>Scene</th>
<th>SSIM (%)</th>
<th>PDiff (%)</th>
<th>PSNR (db)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2d Pattern</td>
<td>99.9</td>
<td>0</td>
<td>53.8</td>
</tr>
<tr>
<td>Mansion</td>
<td>95.6</td>
<td>0.01</td>
<td>34.3</td>
</tr>
<tr>
<td>Sponza</td>
<td>81.2</td>
<td>0.05</td>
<td>29.5</td>
</tr>
<tr>
<td>Hairball</td>
<td>97.6</td>
<td>0.08</td>
<td>31.8</td>
</tr>
<tr>
<td>Vol. Smoke</td>
<td>96.5</td>
<td>0</td>
<td>42.5</td>
</tr>
</tbody>
</table>

Table 1: Image quality metrics of our technique compared to standard per-pixel shading (PPS).

6. Discussion
Triangle Size. Less pixels are covered by each triangle, negatively affecting the efficiency of many HW rasterizers.
Compression. Reading from MSAA render buffers might disable compression or trigger a decompression operation on some HW architectures.

7. References

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no 600533