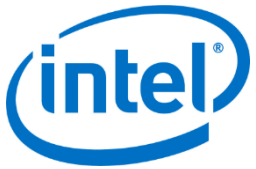
Coarse Pixel Shading

Karthik Vaidyanathan, Marco Salvi, Robert Toth, Tim Foley,
Tomas Akenine-Möller, Jim Nilsson, Jacob Munkberg, Jon Hasselgren,
Masamichi Sugihara, Petrik Clarberg, Tomasz Janczak, and Aaron Lefohn



Motivation

27”
2560 X 1440 Pixels

6”
2560 X 1440 Pixels


The Sword
& Shield

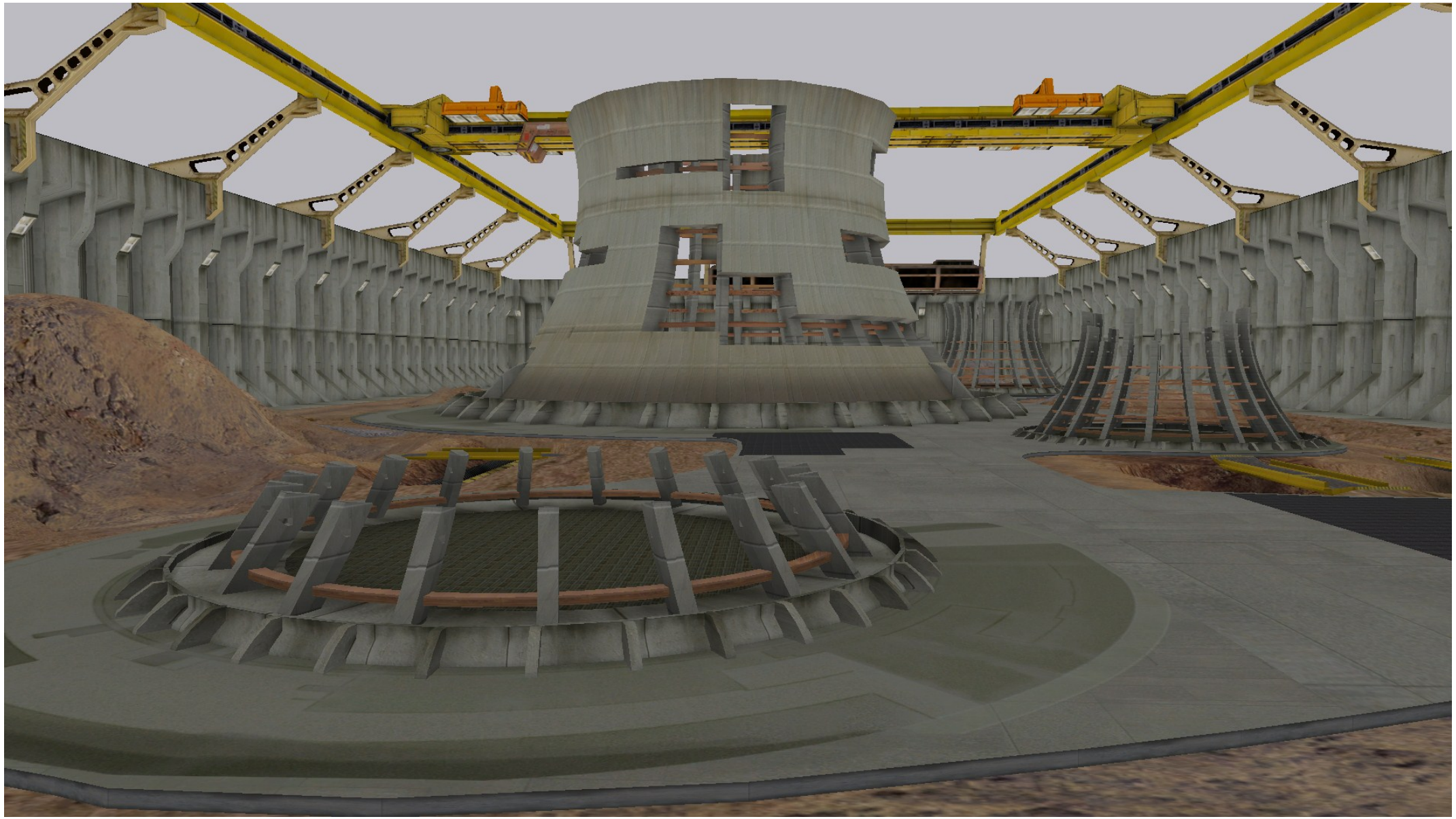

The Sword
& Shield

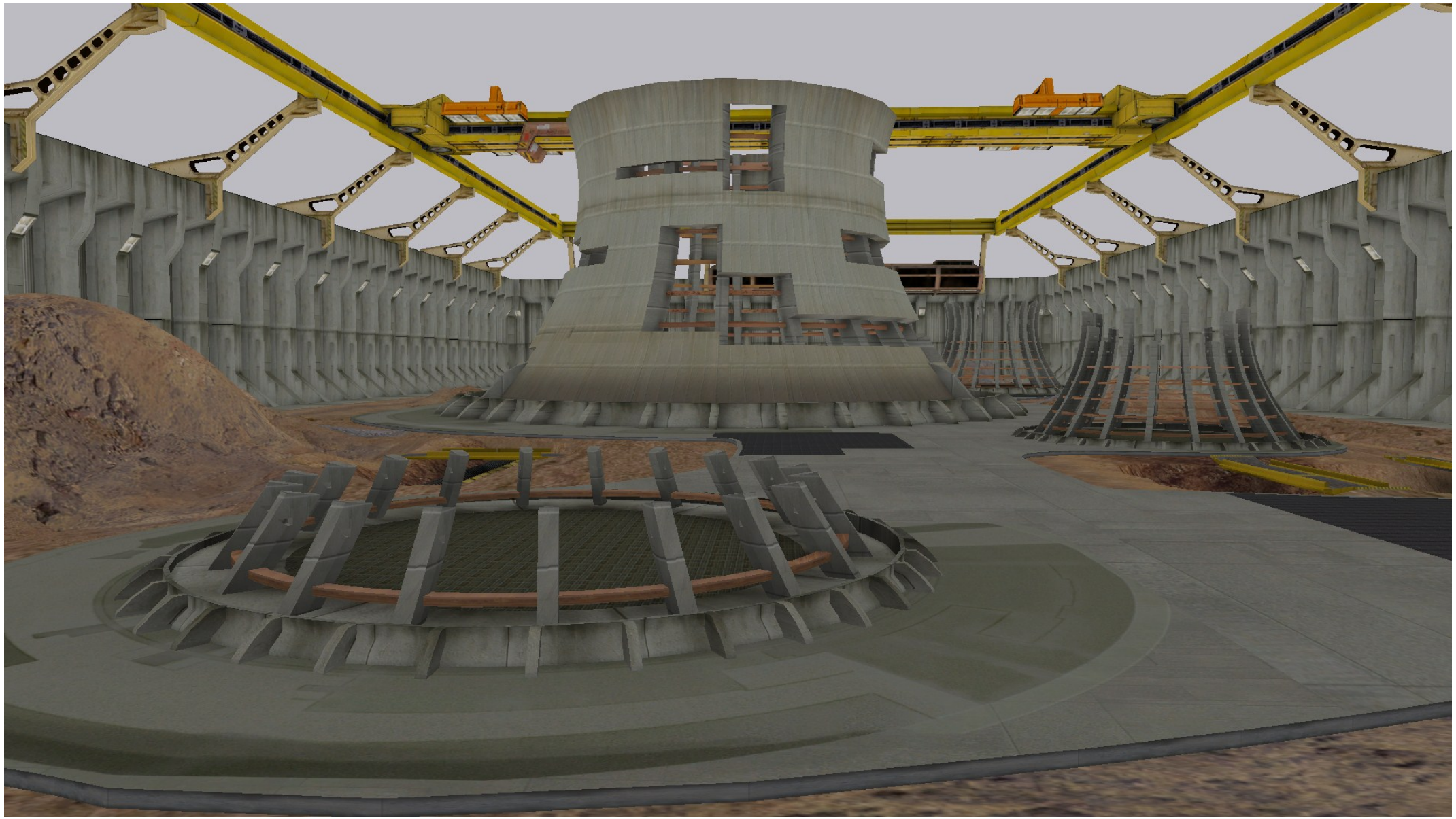
Shade Less Than Once Per Pixel

- High Density Display
- Diffuse Indirect Illumination
- Low Detail Materials
- Motion and Defocus Blurred Regions
- Foveated Rendering
- …


The Sword
& Shield


The Sword
& Shield

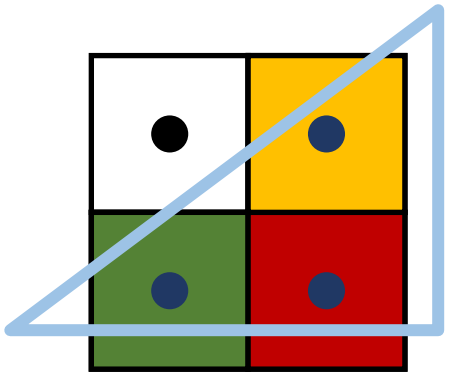




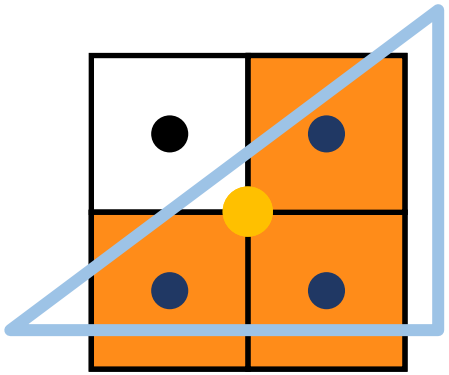
New Capabilities

- Programmable control of shading rates
  - e.g. shade once every 2x2 pixels
- Different components at different rates
  - Multi-rate shading
  - e.g. Ambient occlusion at a reduced rate, other terms at pixel rate

Coarse Pixels



- **Typically**
  - 1 visibility sample, and 1 shading sample per pixel



- **CPS**
  - 1 visibility sample per pixel, but 1 shading sample per NxM pixels (**coarse pixel**)

APIs

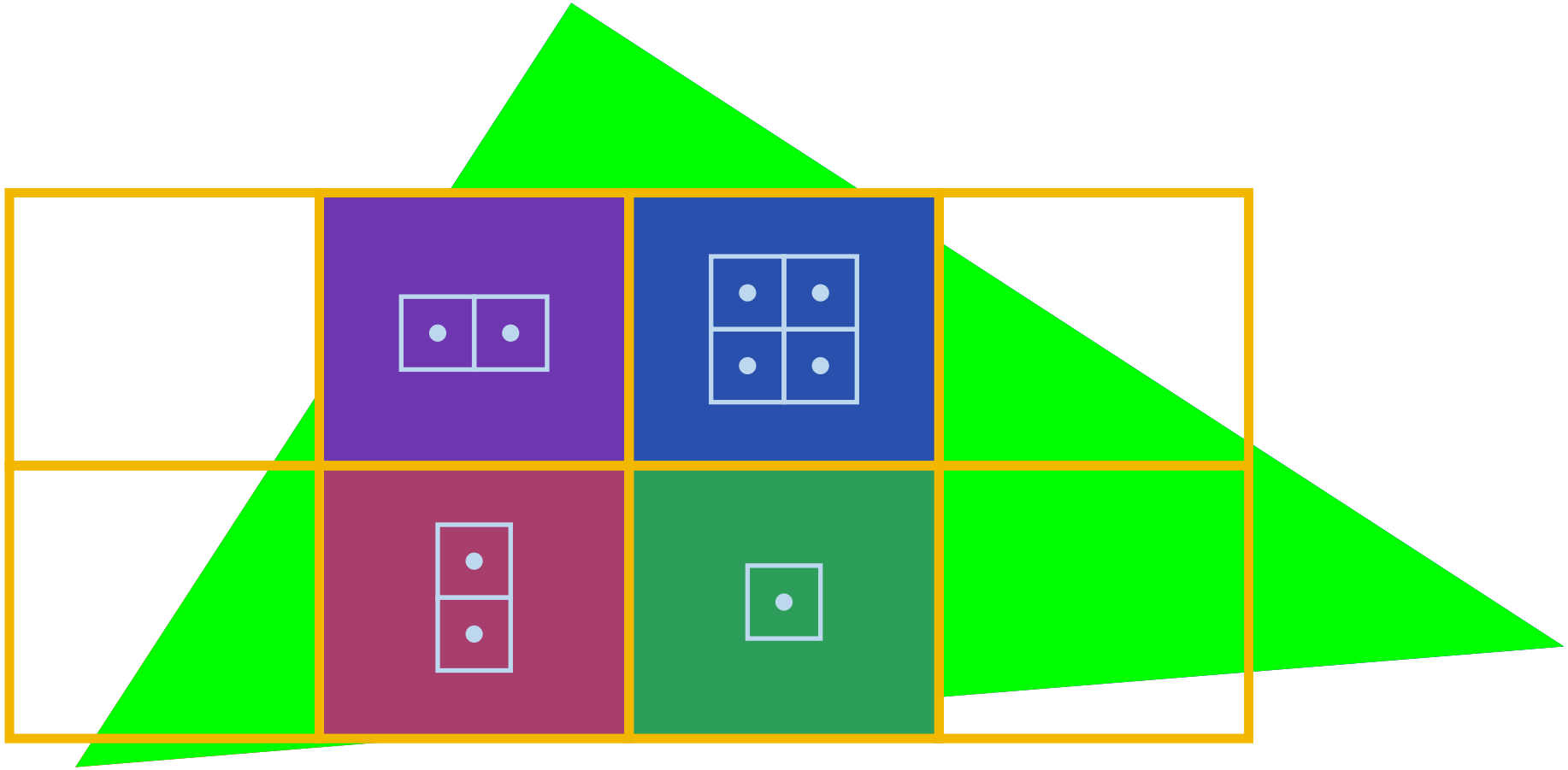
1. Constant shading rate selected via render state
2. Interpolated from vertex shader outputs
3. Radial function of screen coordinates

Programming - Example

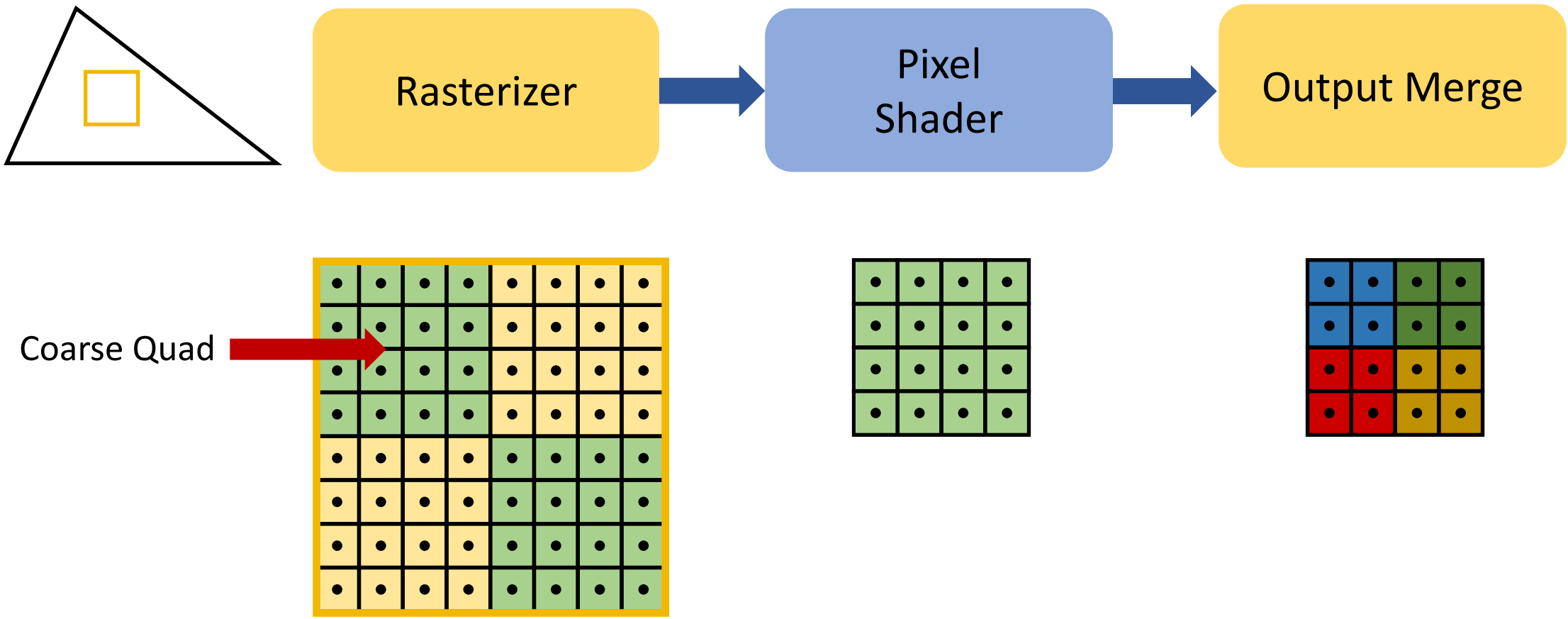
```
VS_OUT VertexShader(VS_IN In)
{
    float2 cpsize = In.pos.z * ...

    VS_OUT Out;
    Out.cpsize = cpsize;

    return out;

}
```

```
struct VS_OUT
{
  float2 cpsize : SV_CoarsePixelSize;
  ...
};
```

Pipeline Changes

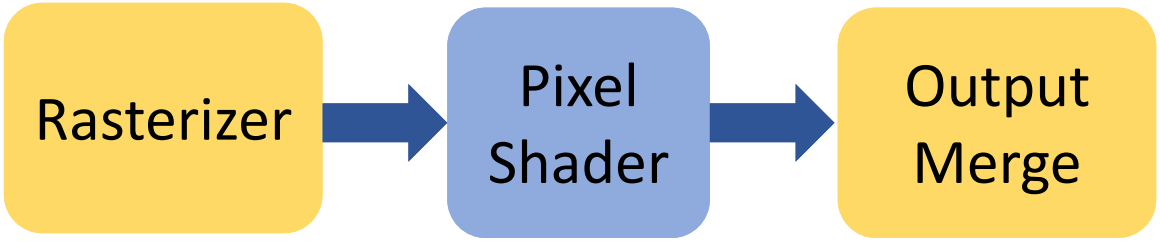


Pipeline Operation

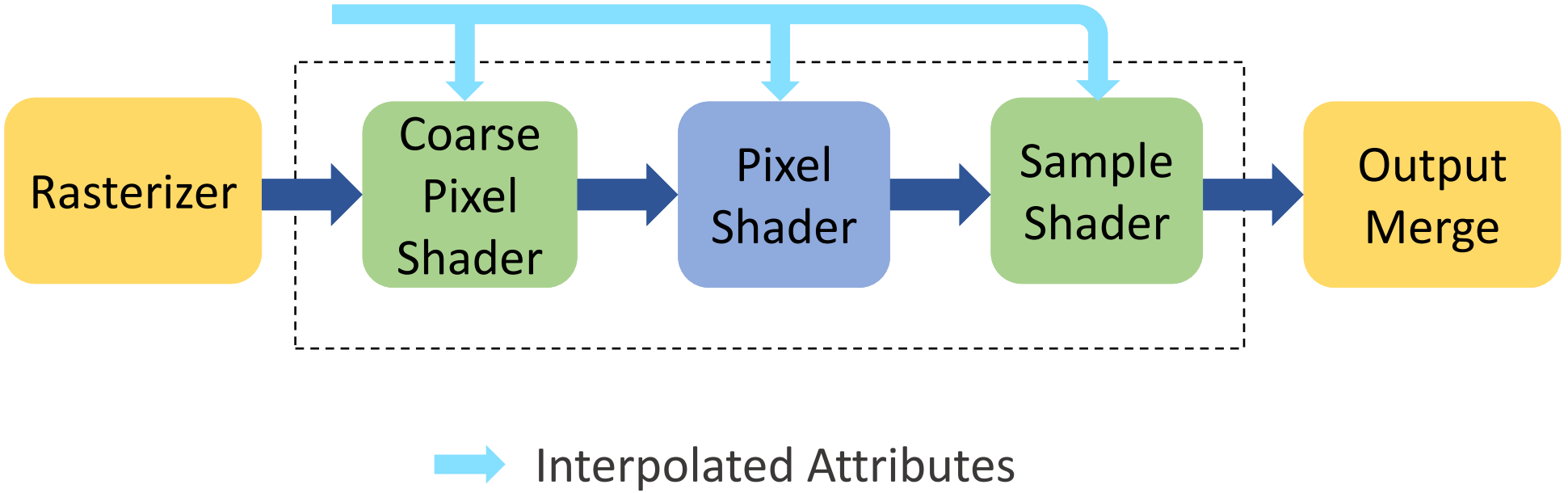


Multi Rate Shading

Multi-Rate Shading



Multi-Rate Shading



Example Usage

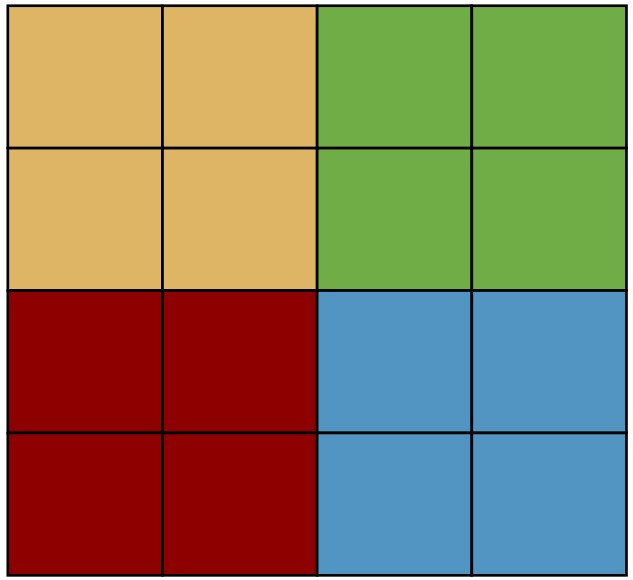
```
[shadingphase("coarse-pixel")]
[nextshaderfunc("PixelShader")]

float4 CoarseShader( VS_OUT In )
{
    float4 res = ComputeDiffuse(In)

    return res;

}
```

```
[shadingphase("pixel")]

float4 PixelShader( VS_OUT In , float4 CPIn )
{
    float4 specular = ComputeSpecular(In)

    return (specular*CPIn);

}
```

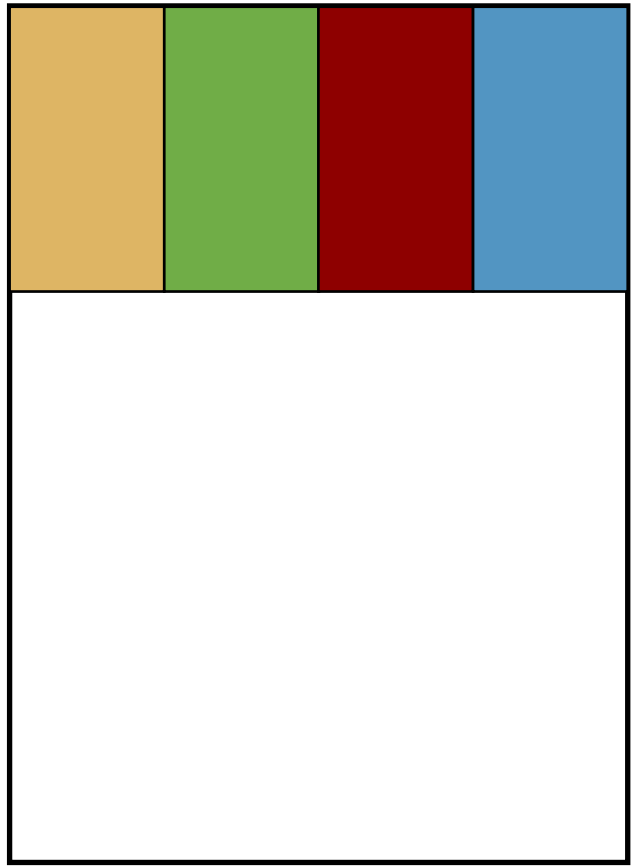
Scheduling

- All phases scheduled on the same thread
- Execute coarse phase then loop over pixels/samples
- Inter-phase data in registers

Scheduling



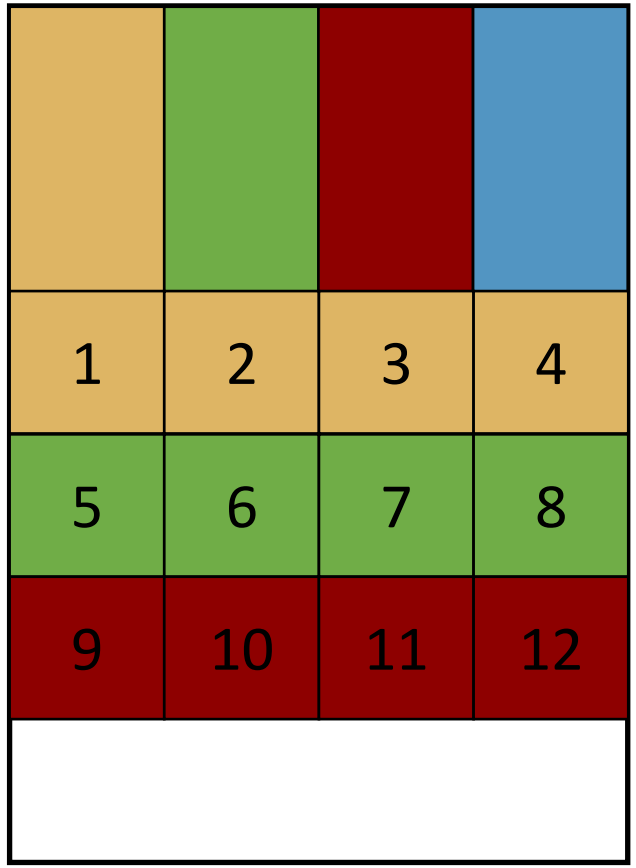
SIMD Lanes

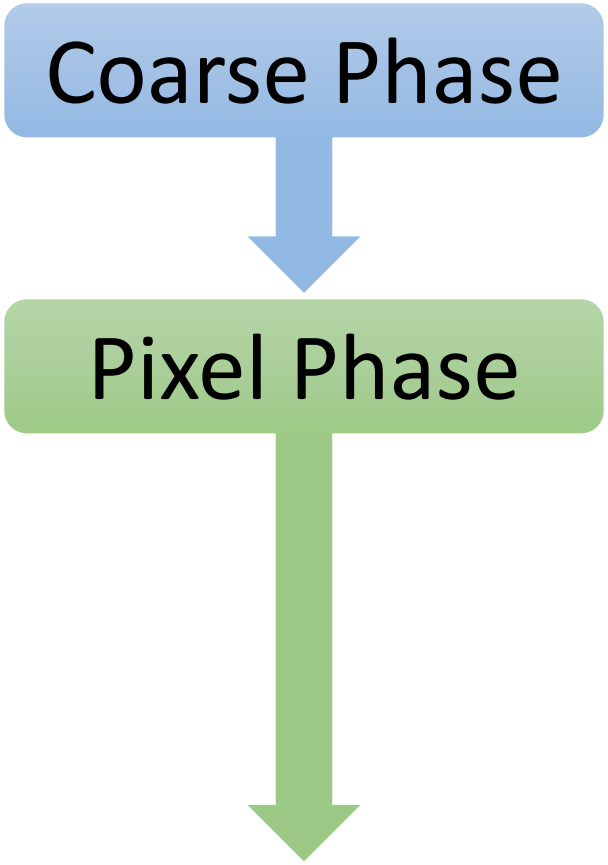


Coarse Phase

Scheduling

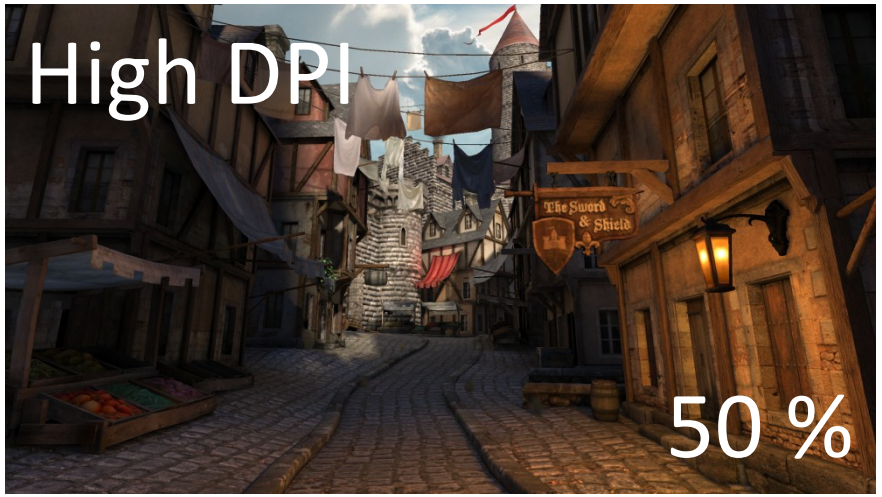
SIMD Lanes

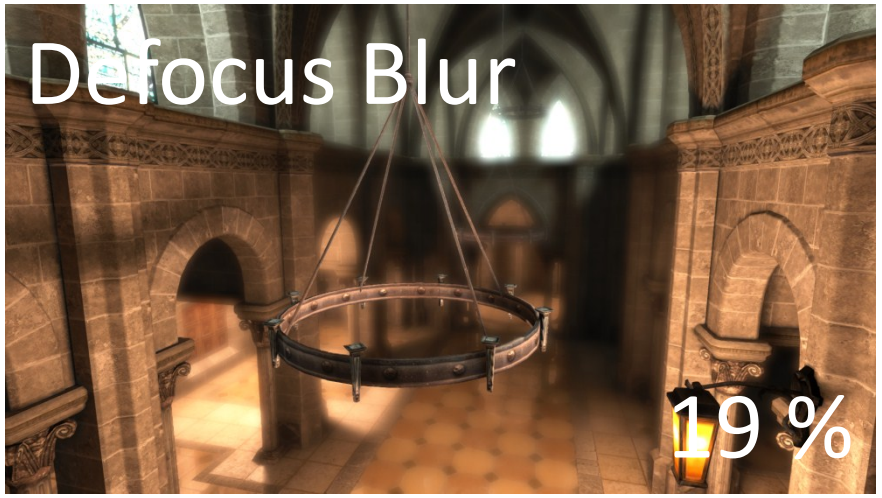


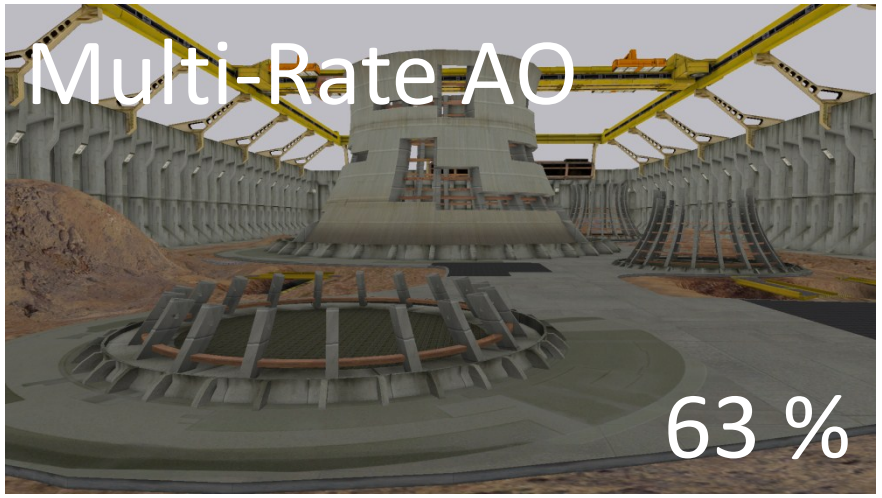


Results

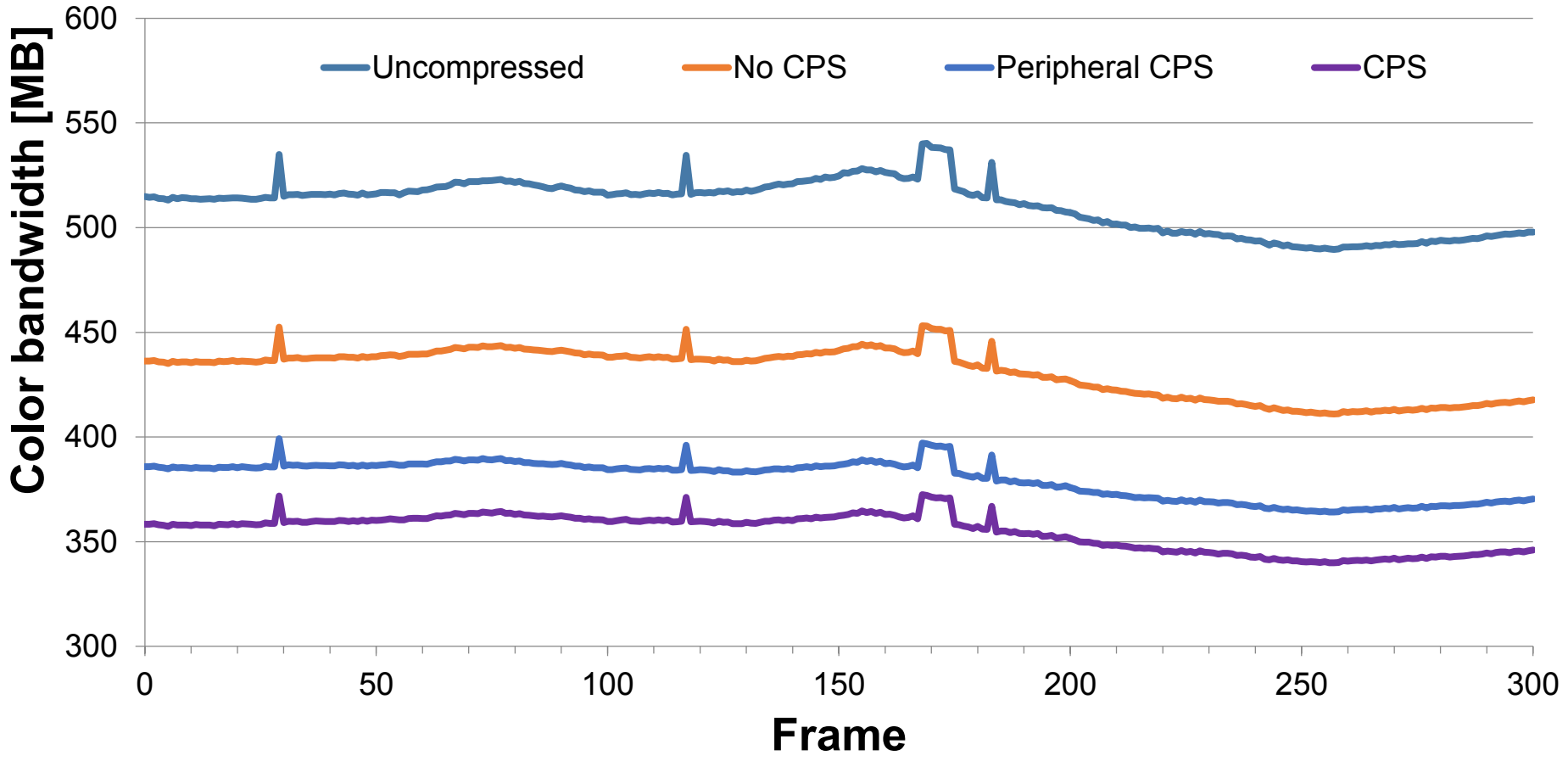
Applications







Color Compression



Acknowledgements

- Tom Piazza
- David Blythe
- Charles Lingle
- Prasoonkumar Surti
- Uzi Sarel
- Bar On Tomer

We’re hiring in San Francisco!

http://bit.ly/sfposition