FFX CACAO

JAY FRASER

COMBINED ADAPTIVE COMPUTE AMBIENT OCCLUSION

- CACAO is a highly optimised adaptation of the Intel ASSAO screen space ambient occlusion implementation.
- CACAO provides 5 quality levels for SSAO generation (LOWEST, LOW, MEDIUM, HIGH, HIGHEST), the last of which uses an adaptive approach.

PIPELINE OVERVIEW

Non-Adaptive

Adaptive

PIPELINE OVERVIEW (DOWNSAMPLED)

Non-Adaptive

Prepare → Generate SSAO → Edge Aware Blur → Bilateral Upsample

Adaptive

Prepare → Generate SSAO → Create Importance Map → Generate Adaptive SSAO → Edge Aware Blur → Bilateral Upsample

PREPARE

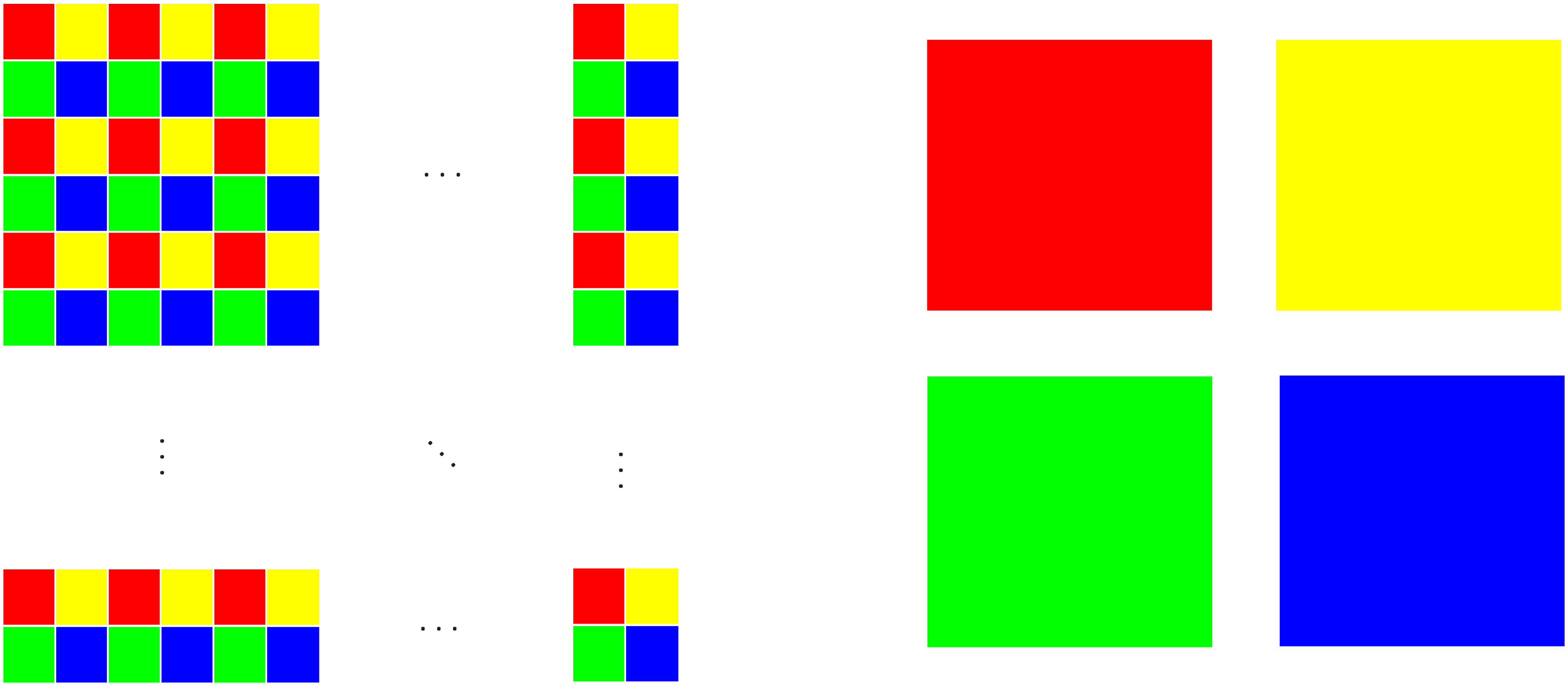
Inputs:

- Depth buffer
- Optional normal buffer

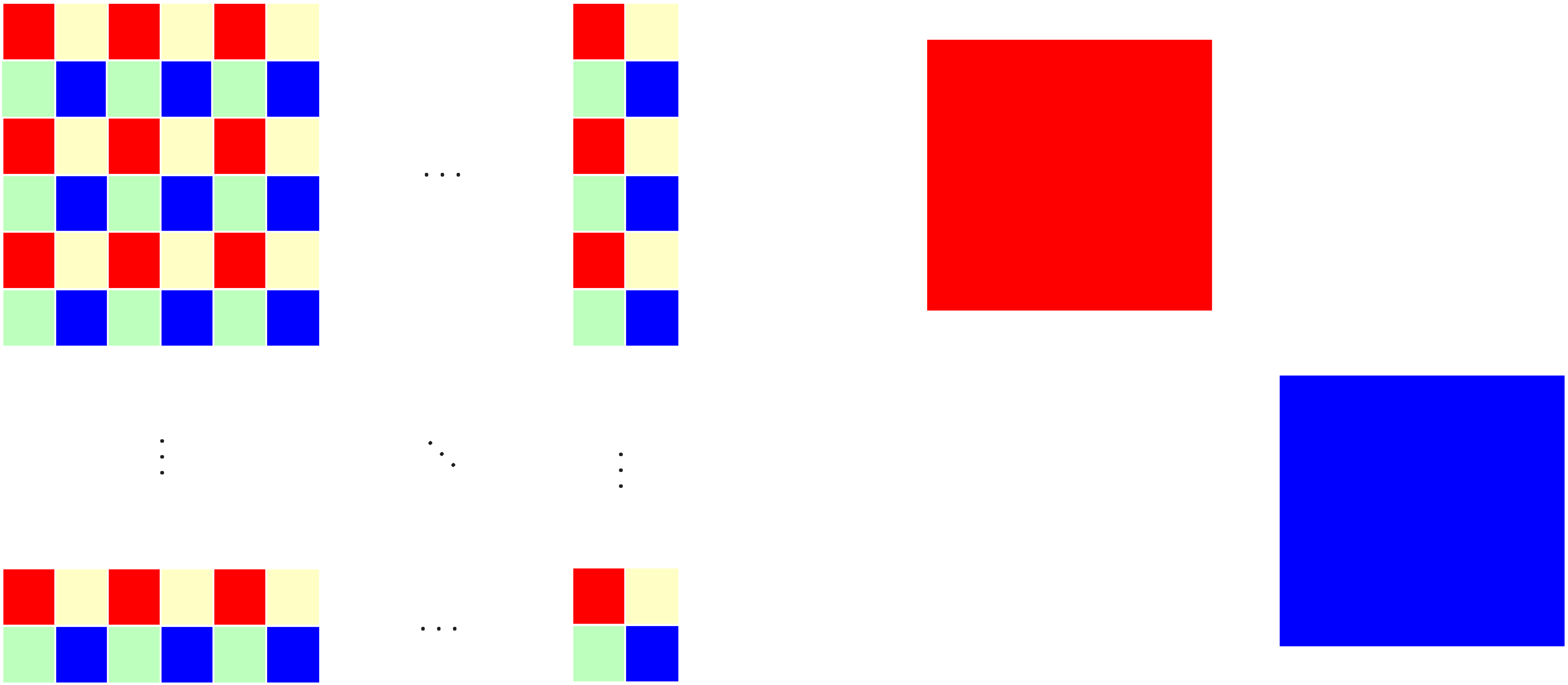
Outputs:

- Deinterleaved depth buffer
- Deinterleaved depth mips (quality level “FFX_CACAO_MEDIUM” or higher)
- Deinterleaved normal buffer
  - Deinterleaved normal buffer is generated from depth buffer when no normal buffer provided

DEINTERLEAVE



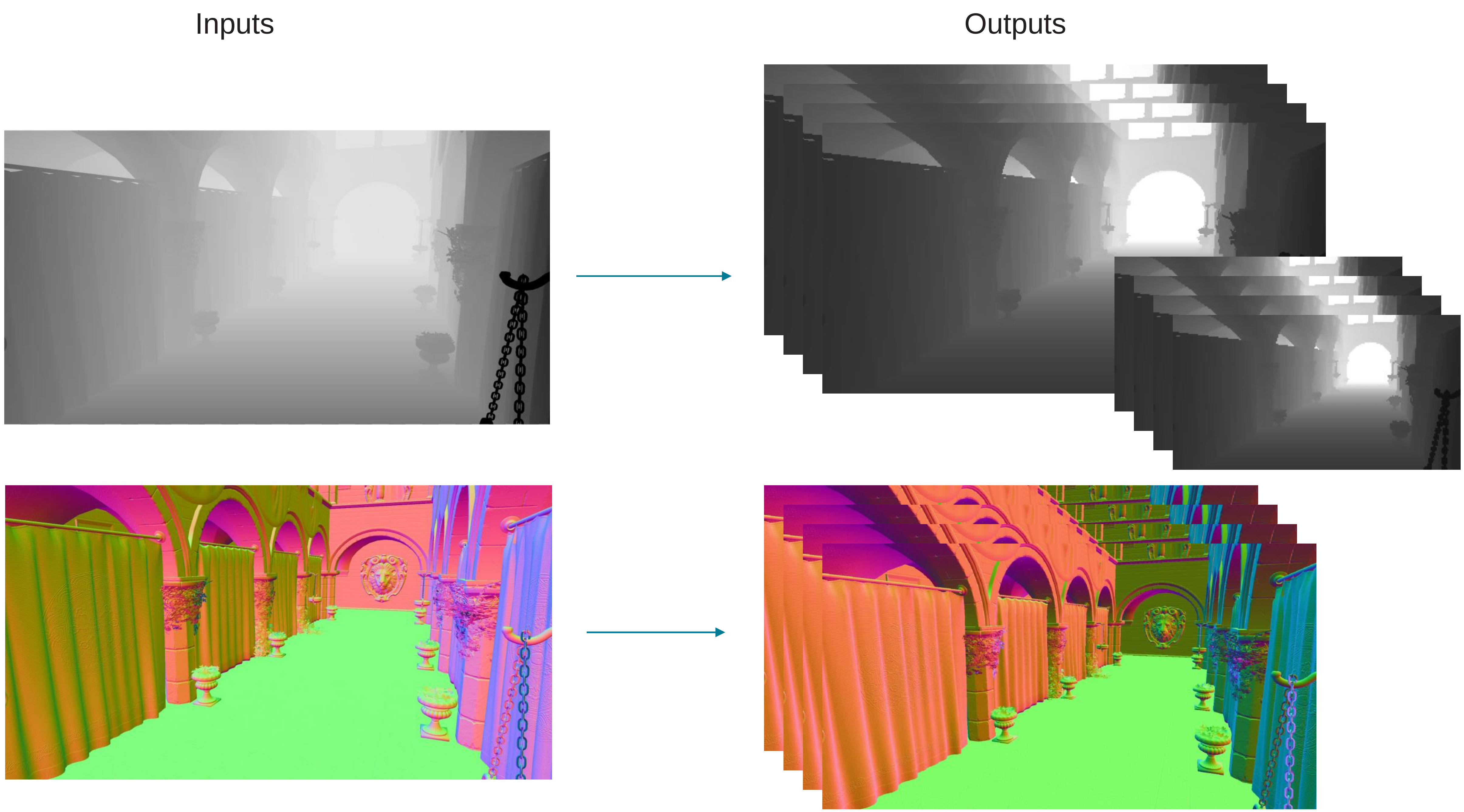
DEINTERLEAVE (LOWEST QUALITY)



PREPARE SHADERS

Each shader name begins with “CSPrepareNative” or “CSPrepareDownsampled” corresponding to generating native resolution SSAO or downsampled SSAO.

- [Prepare]DepthsAndMips – Prepares deinterleaved depth buffer with mip chains
- [Prepare]Depths – Prepares deinterleaved depth buffer without mip chains (for low quality)
- [Prepare]DepthsHalf – Prepares the deinterleaved depth buffer for lowest quality
- [Prepare]NormalsFromInputNormals – Prepares the deinterleaved normal buffer from an input normal buffer
- [Prepare]Normals – Prepares the deinterleaved normal buffer from the depth buffer


Inputs
Outputs

GENERATE SSAO

The generate SSAO stage calculates obscurance values and detects edges for use in subsequent edge sensitive blurring.

Inputs

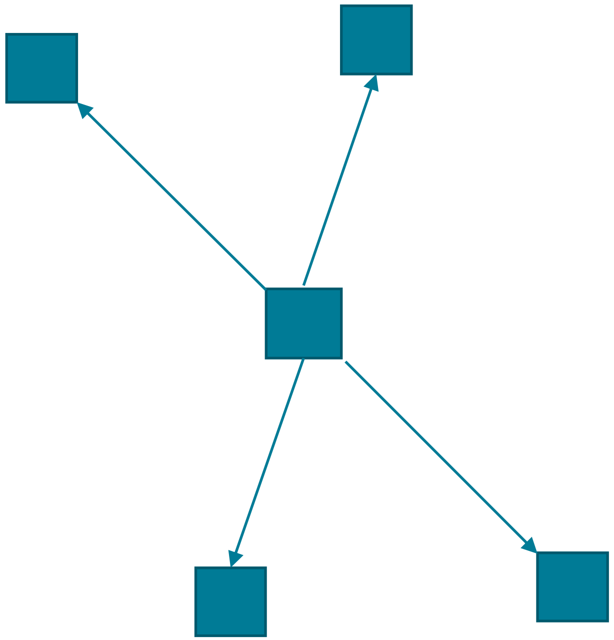
- Deinterleaved depth buffer, with mip chain if applicable to quality level
- Deinterleaved normal buffer

Outputs

- Red channel – obscurance values
- Green channel – edge values

GENERATE SSAO – OBSCURANCE VALUES

For each pixel we read its depth and normal values. We then sample depths in a rotationally symmetric pattern around the pixel. At higher quality levels, we may sample depths from multiple mip levels. The sampling pattern is scaled depending on the depth of the pixel. The sampling pattern is rotated for neighbouring pixels. For each pixel we sample, we calculate an obscurance value. The final obscurance value for each pixel is a weighted average of all obscurance values from the samples.



GENERATE SSAO – OBSCURANCE VALUES

The calculated obscurance value for a pixel with position **p** and normal **n** from a sample at position **q** is as follows.

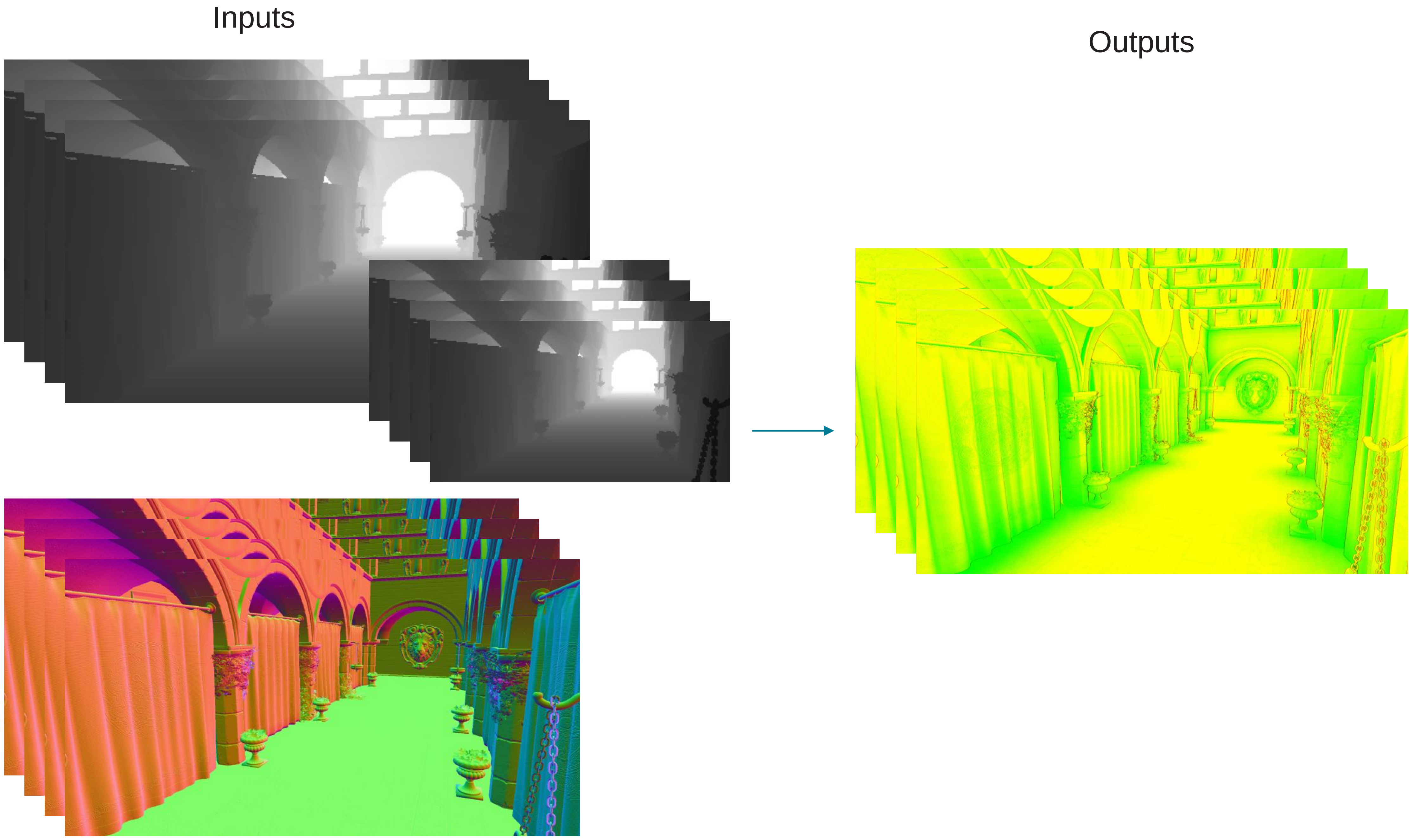
d = (**q** – **p**)

theta = **n** . **d** / |**d**|

fall_off_multiplier = max(0, 1 – fall_off_constant * |**d**|^2)

obscurance = max(0, theta – horizon_angle_threshold) * fall_off_multiplier

The obscurance terms are the cosine of the angle between the hit direction and the normal, multiplied by a fall off which increases with the square of the distance between the pixel and the sample.


Inputs
Outputs

GENERATE SSAO - ADAPTIVE

At adaptive quality levels, there are two further SSAO generating passes.

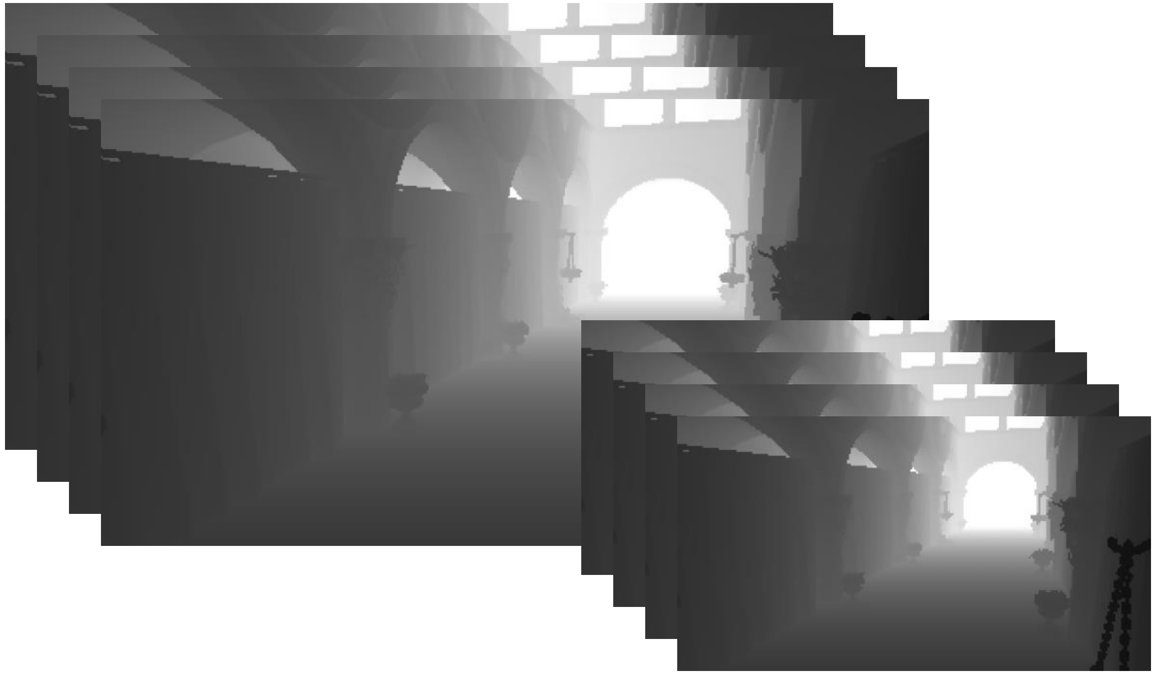
- Generate Adaptive SSAO base pass
- Generate Adaptive SSAO

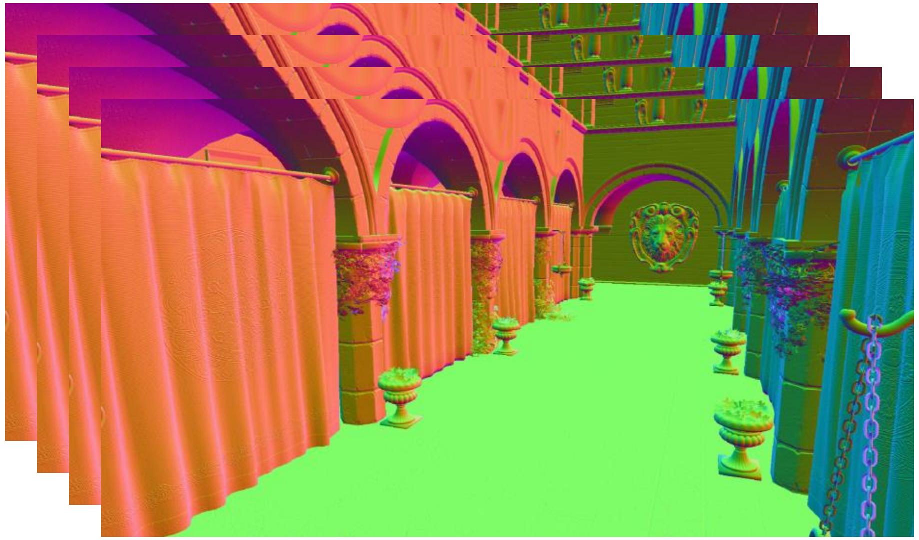
The base pass calculates SSAO in the same way as the other passes, however it exits early after writing non-transformed obscurance values and skipping edge detection.

The adaptive SSAO generation takes additional inputs (the importance map, load counter, and output from the base pass), and then performs a variable number of additional samples after the base pass based on the computed importance for the location given by the importance map.

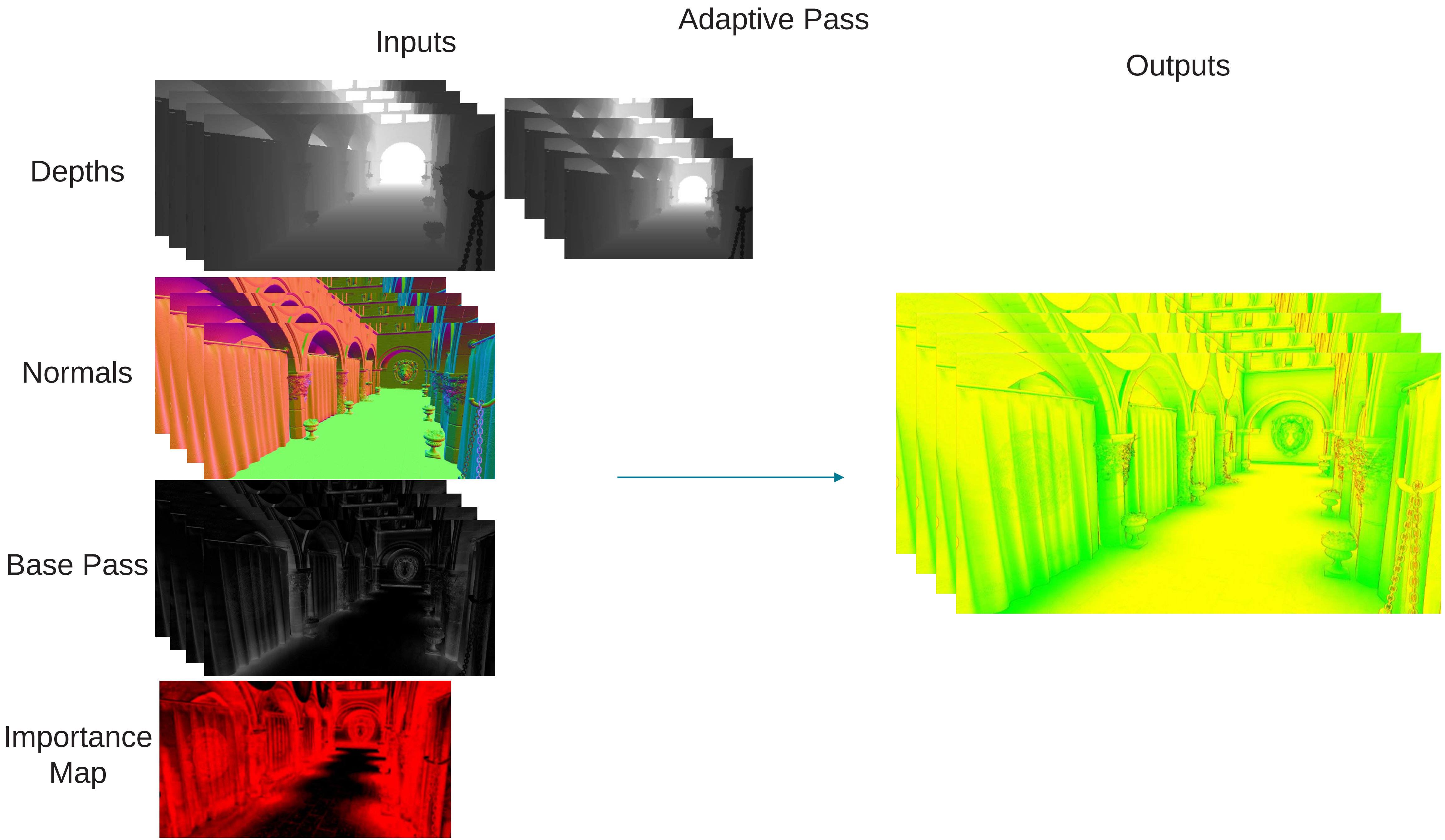
Base Pass

Inputs





Outputs



Adaptive Pass
Inputs
Outputs
Depths
Normals
Base Pass
Importance Map

GENERATE SSAO - SHADERS

- GenerateQ0 – Low quality SSAO generation (used in quality LOWEST and LOW)
- GenerateQ1 – Medium quality SSAO generation (used in quality MEDIUM)
- GenerateQ2 – High quality SSAO generation (used in quality HIGH)
- GenerateQ3Base – Base pass for adaptive SSAO generation (used in quality HIGHEST)
- GenerateQ3 – Adaptive pass for adaptive SSAO generation (used in quality HIGHEST)

CREATE IMPORTANCE MAP

In adaptive quality, after the SSAO base pass has been run an importance map is generated to determine where to use most samples in the final effect.

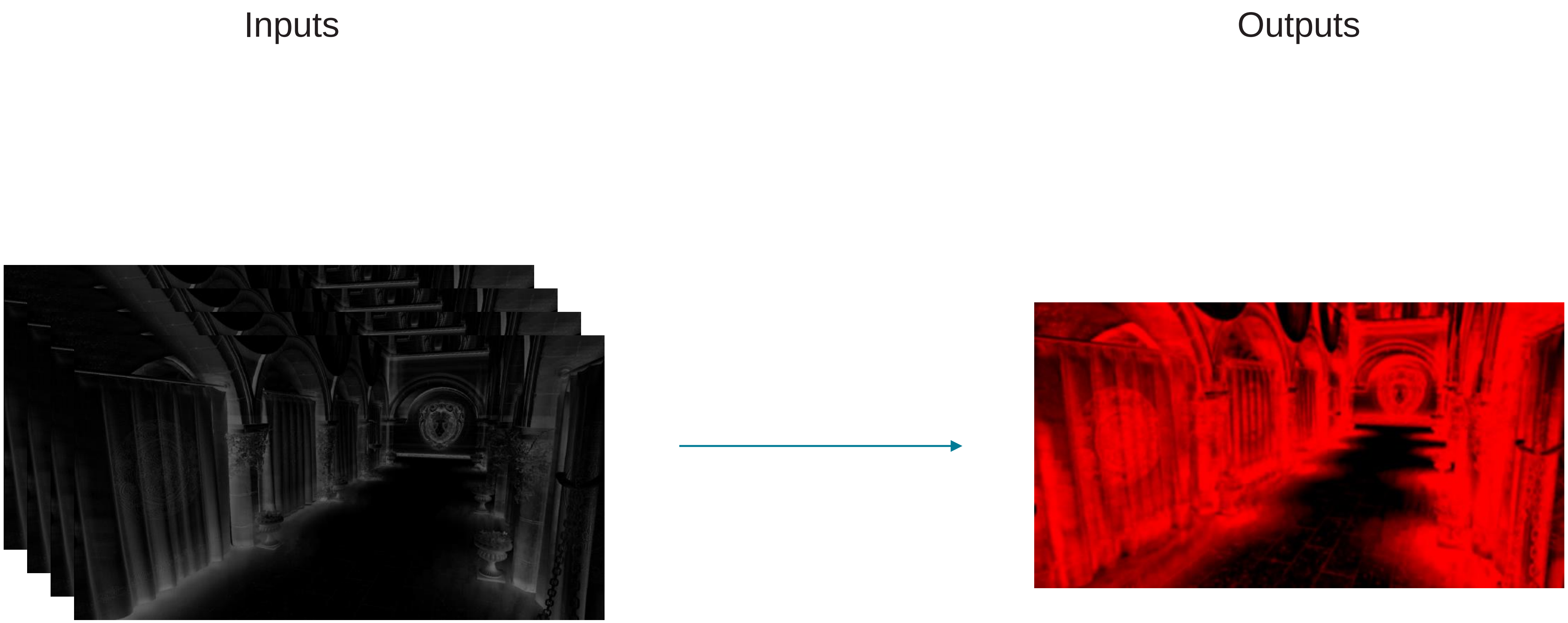
Inputs:

- SSAO generated from the base pass

Outputs:

- Importance map

Each importance value is the importance map corresponds to an 8x8 square of SSAO values, and the importance is set to the difference between the minimum and maximum values in that square. The importance map is then blurred to avoid sharp transitions from important to non-important areas.


Inputs
Outputs

CREATE IMPORTANCE MAP - SHADERS

- GenerateImportanceMap – Generates the importance map from the base pass
- PostProcessImportanceMapA – does a first blur pass on the importance map
- PostProcessImportanceMapB – does a second blur pass on the importance map

EDGE SENSITIVE BLUR

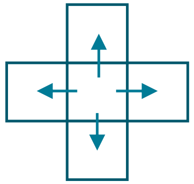
The edge sensitive blur is applied after SSAO generation to help remove noise created by the random sampling. The blur has a 3x3 kernel, where each pixel is weighted by its edge value. The blur may be run for between 0 and 8 passes to effectively create a wider kernel.

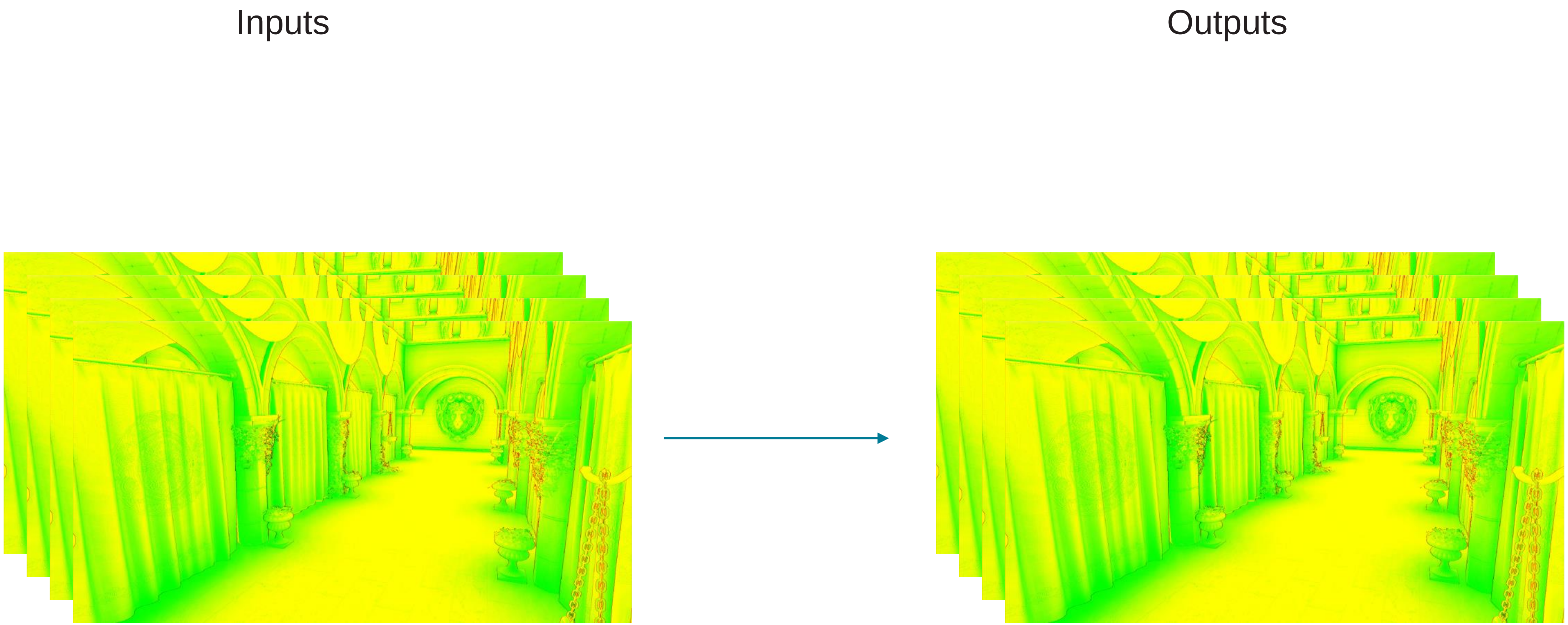
Inputs:

- Generated SSAO texture with edges

Outputs:

- Blurred SSAO textures with edges (still deinterleaved)




Inputs
Outputs

EDGE SENSITIVE BLUR - SHADERS

- EdgeSensitiveBlur1 – single pass edge sensitive blur
- EdgeSensitiveBlur2 – 2 pass edge sensitive blur
- EdgeSensitiveBlur3 – 3 pass edge sensitive blur
- EdgeSensitiveBlur4 – 4 pass edge sensitive blur
- EdgeSensitiveBlur5 – 5 pass edge sensitive blur
- EdgeSensitiveBlur6 – 6 pass edge sensitive blur
- EdgeSensitiveBlur7 – 7 pass edge sensitive blur
- EdgeSensitiveBlur8 – 8 pass edge sensitive blur

BILATERAL UPSAMPLE

A bilateral upsampler is used to create the final output. The upsampler uses a 5x5 kernel of input SSAO values and their corresponding depths and creates a blended output value.

Inputs:

- Deinterleaved SSAO textures
- Deinterleaved depths
- Input depth buffer

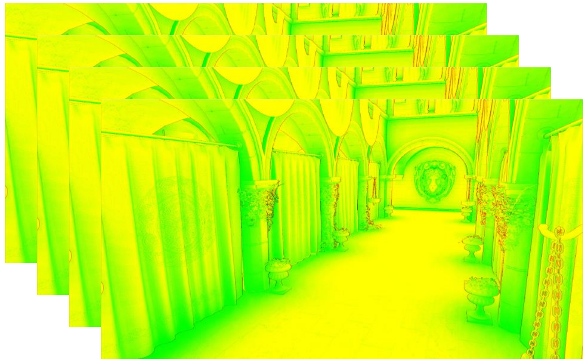
Outputs:

- Final output texture

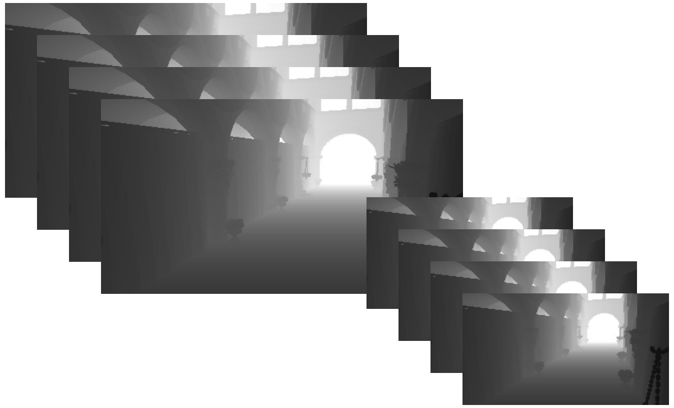
Inputs

Outputs

Blurred
SSAO



Screen
Space
Depths
Mips



Input
Depths

BILATERAL UPSAMPLE - SHADERS

- Bilateral5x5 – used for downsampled in low quality and above
- Bilateral5x5Half – used for downsampled at lowest quality