unity

Unity技术开放日

UNITY OPEN DAY

高灵活度、低美术成本的水体渲染系统

王骁建
Graphics Programmer @ Unity TA Team

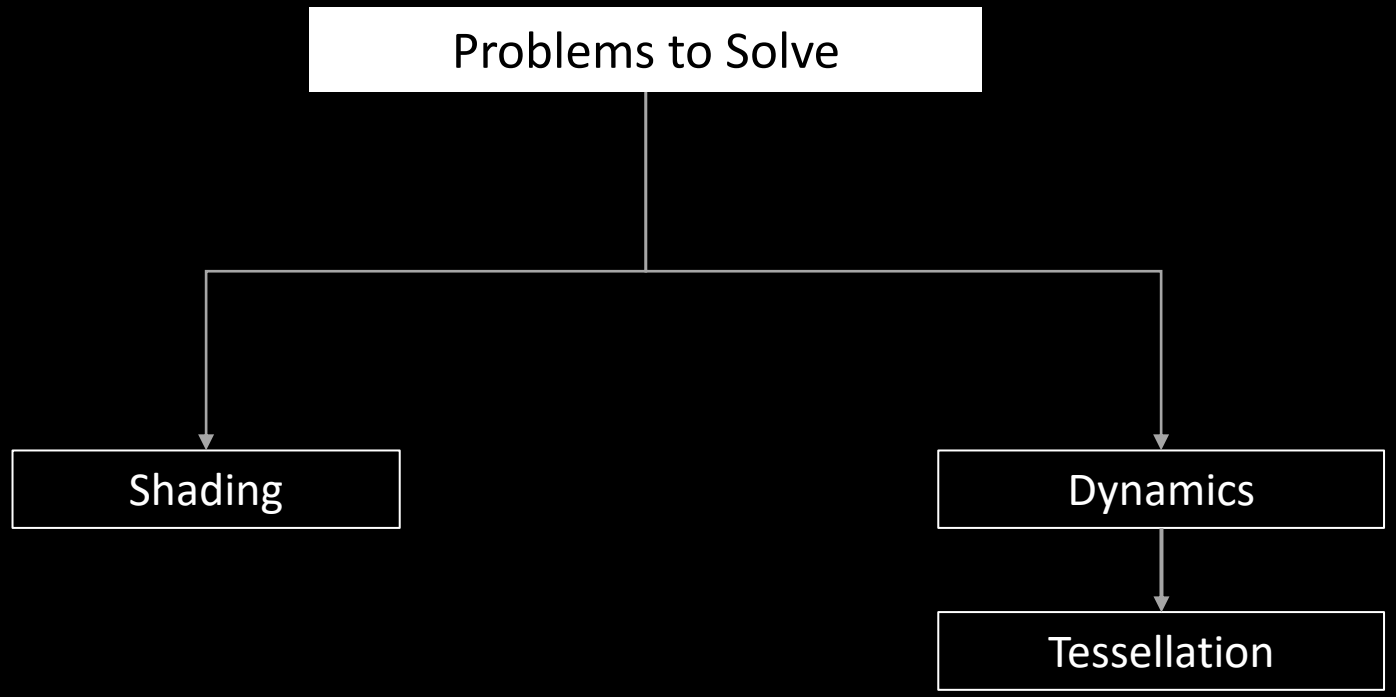
Diverse Appearance

Drastic Motion

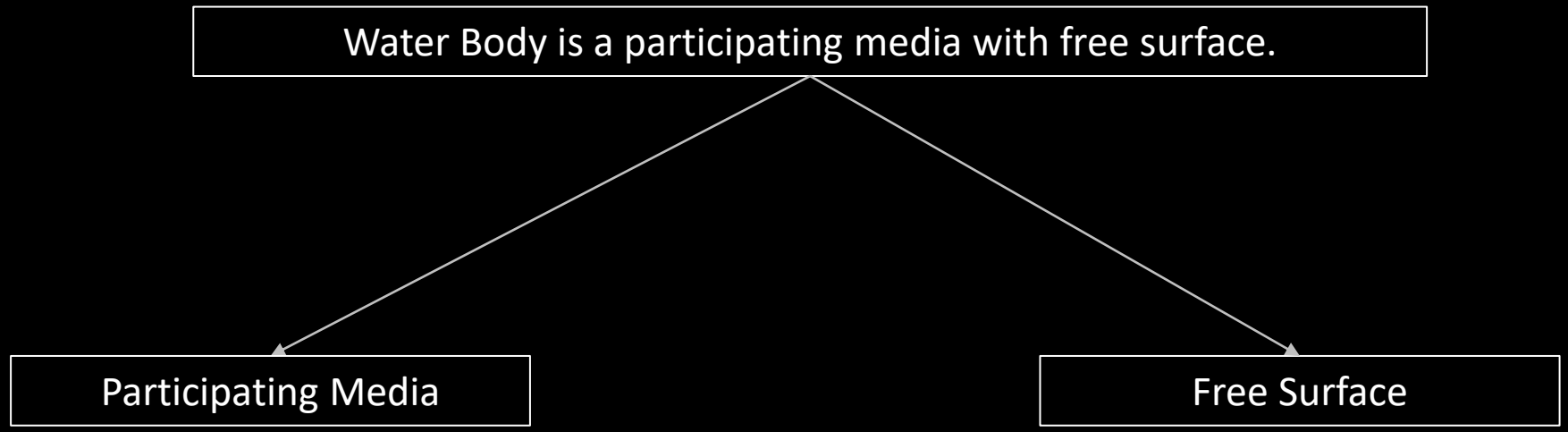
Complex Optical Behavior

Highly Detailed

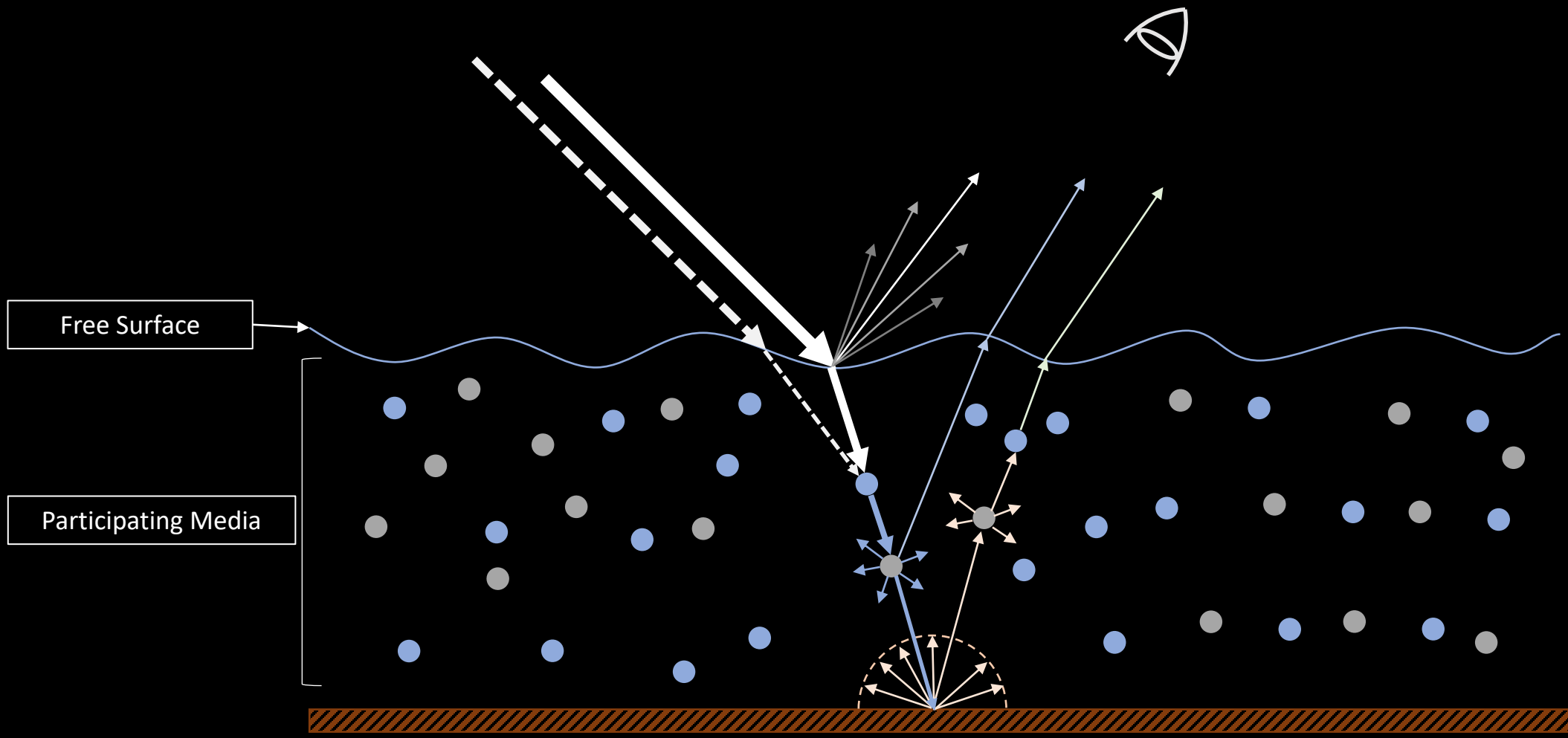
Water Rendering



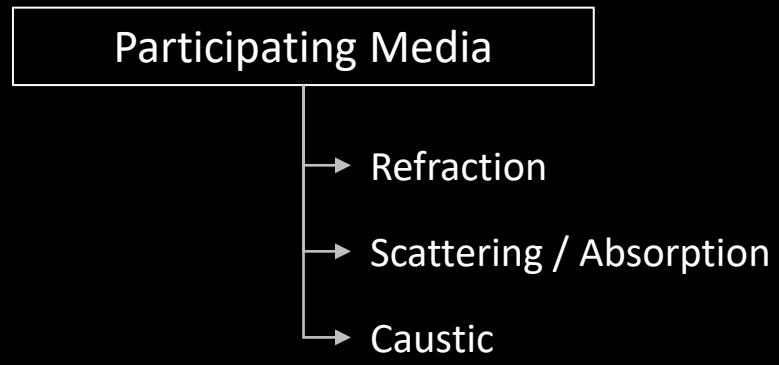
Shading

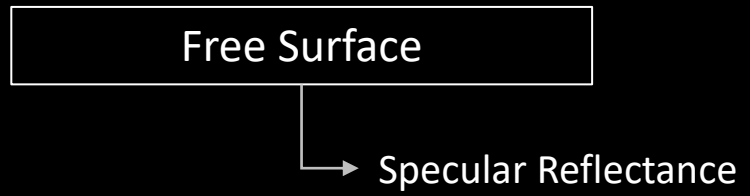

Water Body is a participating media with free surface.
Participating Media
Free Surface

Water Lighting Behavior



Shading


Participating Media
Refraction
Scattering / Absorption
Caustic


Free Surface
Specular Reflectance


Original Scene
y
z
x
Persp


Refraction
y
z
x
Persp


Absorption
y
z
x
Persp


Scattering
y
z
x
Persp


Screen Space Reflection
y
z
x
Persp


Reflection Probe
y
z
x
Persp


Specular Lighting
y
z
x
Persp


Caustics
y
z
x
Persp

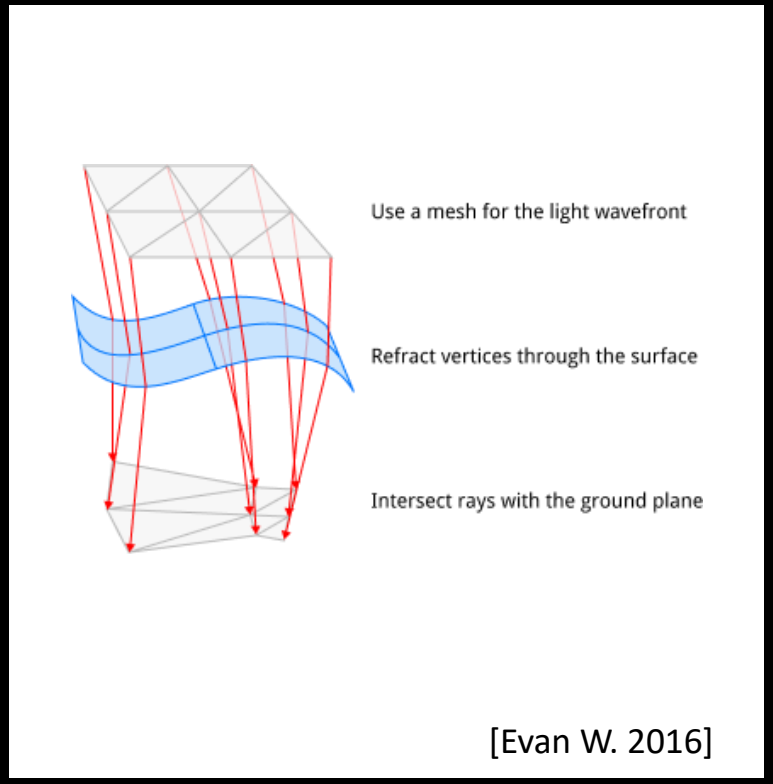
Shading

Debris

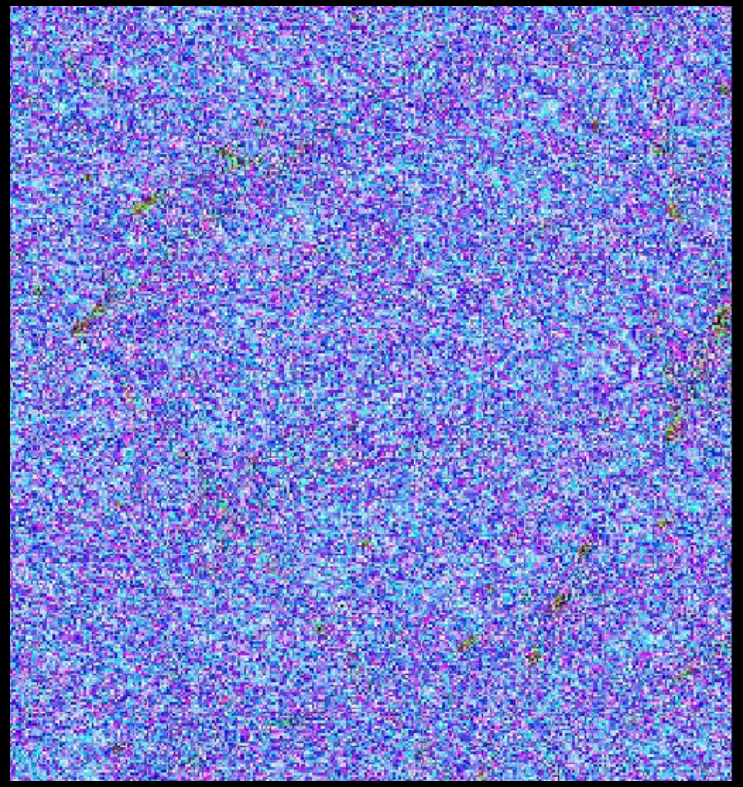
Foam

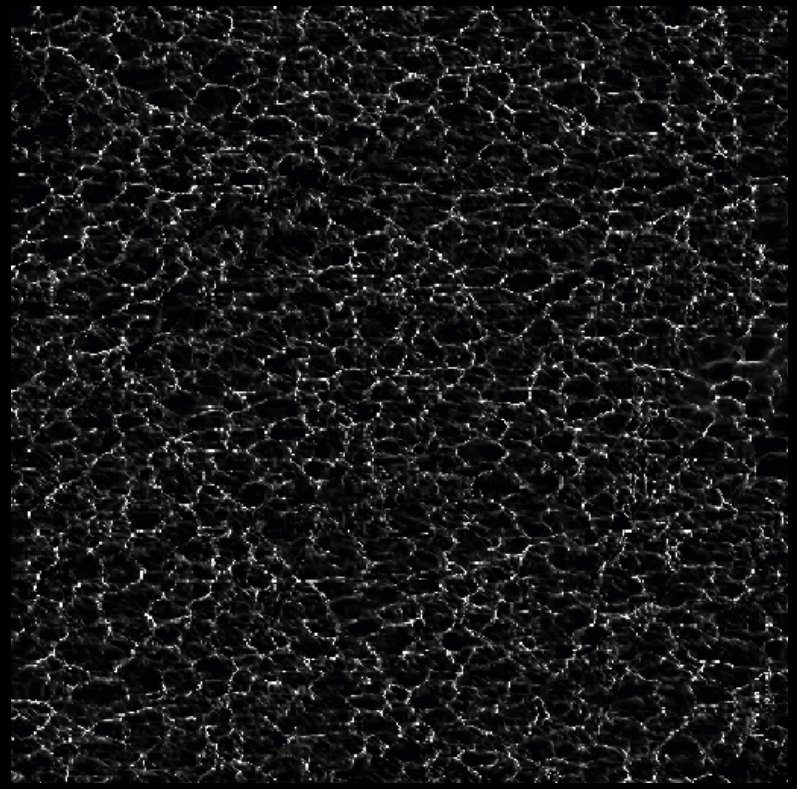
Decal

Realtime Caustics

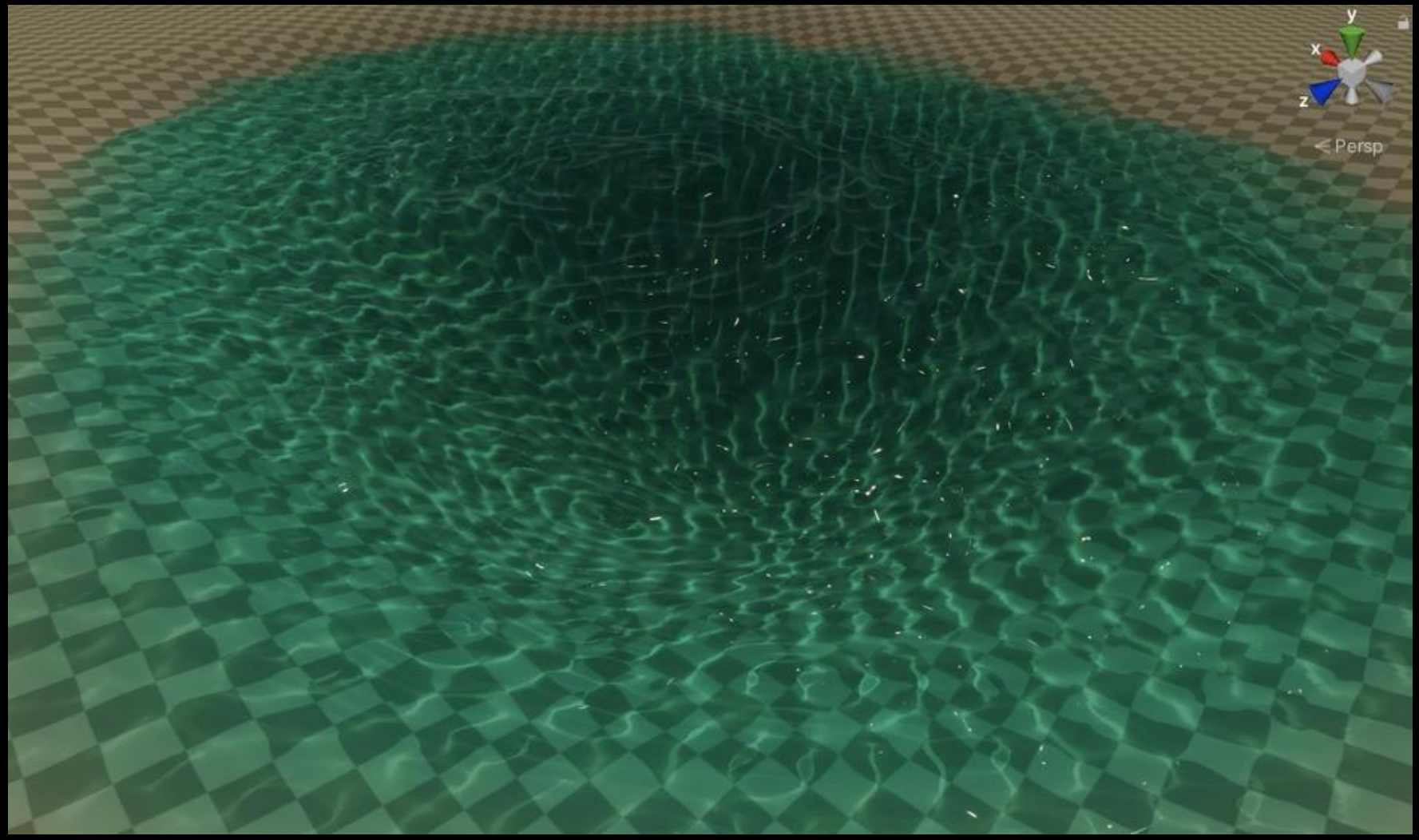


[Evan W. 2016]

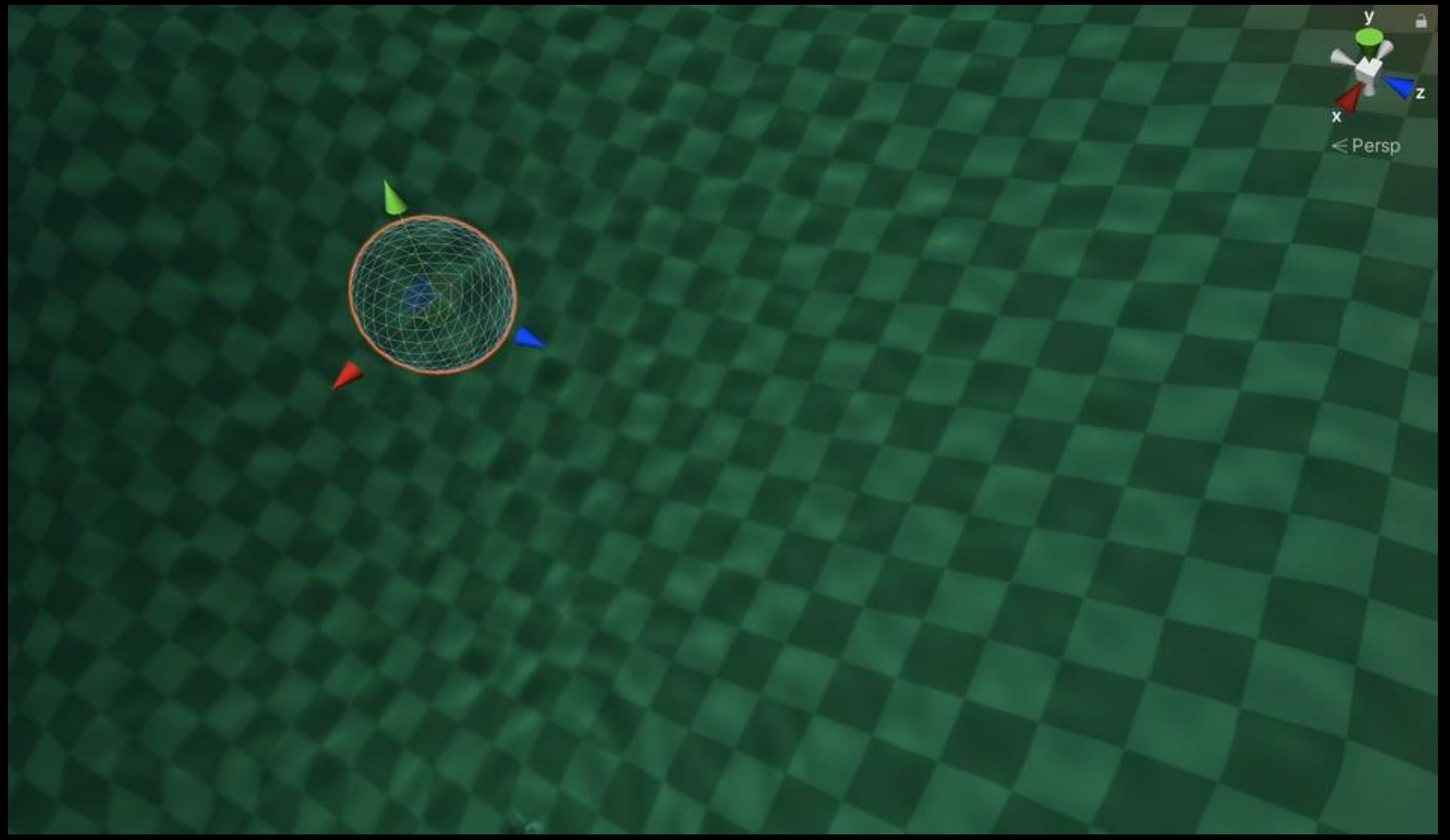




Realtime Caustics

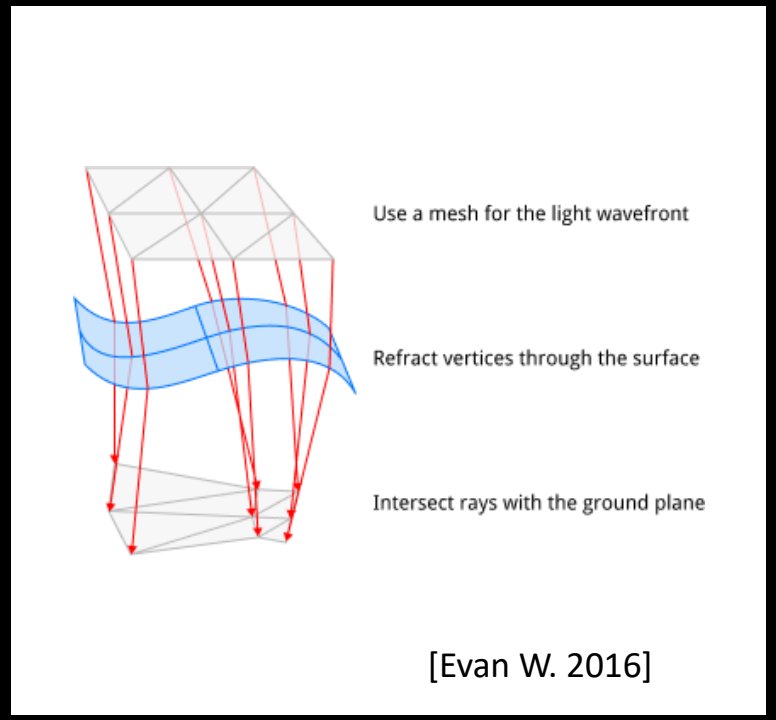


Realtime Caustics



Realtime Caustics

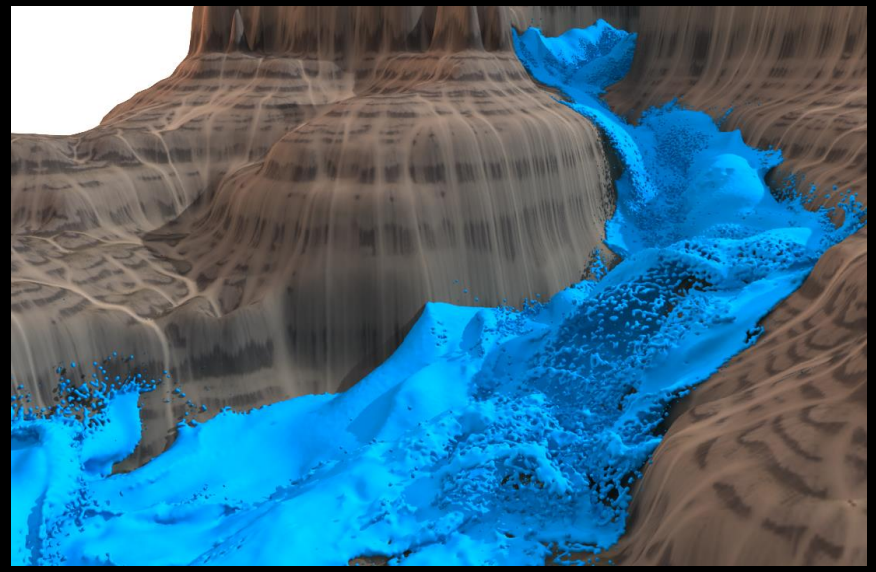
```
85      float3 posWS = mul(float4(posNDC, 1), shadowToWorld);
86      float3 posWSUp = mul(float4(posNDCUp, 1), shadowToWorld);
87      float3 posWSRight = mul(float4(posNDCRight, 1), shadowToWorld);
88
89      float3 posWSRefracted = RefractVertex(posWS, normalWS, L, refractionDepth);
90      float3 posWSUpRefracted = RefractVertex(posWSUp, normalWSUp, L, refractionDepth);
91      float3 posWSRightRefracted = RefractVertex(posWSRight, normalWSRight, L, refractionDepth);
92
93      float area = TriangleArea(posWS, posWSUp, posWSRight);
94      float areaRT = TriangleArea(posWSRefracted, posWSUpRefracted, posWSRightRefracted);
95      output.caustic = float3(area, areaRT, 0);
```



[Evan W. 2016]

Dynamics

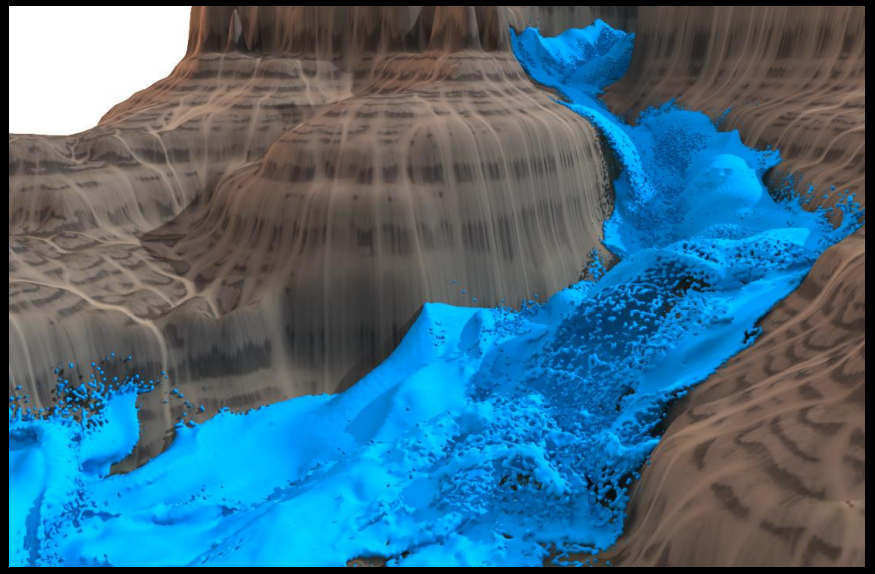
Partial Differential Equation Based



Spectrum Based

Dynamics

Partial Differential Equation Based



Solve the Navier-Strokes Equation
Eularian/Lagrangian

Pros

- Physically Correct
- Rich Appearance
- Fully Dynamic
- ……

Cons

- Too expensive to be real-time

Dynamics

Spectrum Based

Select a group of frequency
Propagate them by some rules

Pros

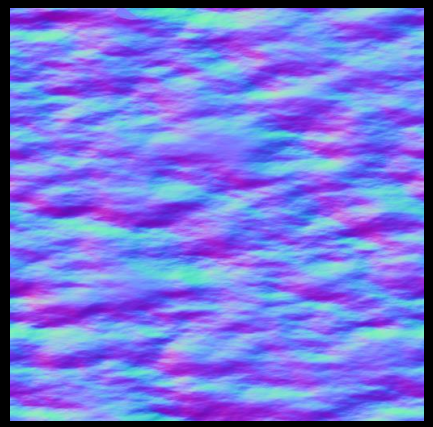
- Faaaaaaaaaaaaaaaaaaast!
- Better artistic control
- Rich appearance also can be achieved

Critical

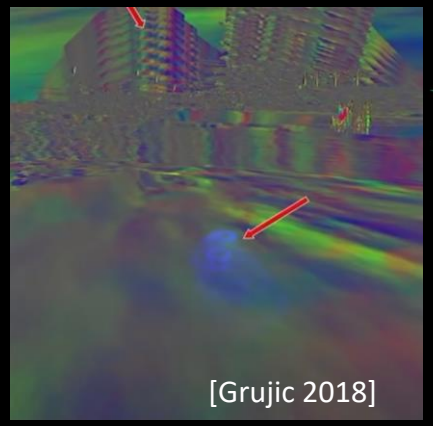
Cons

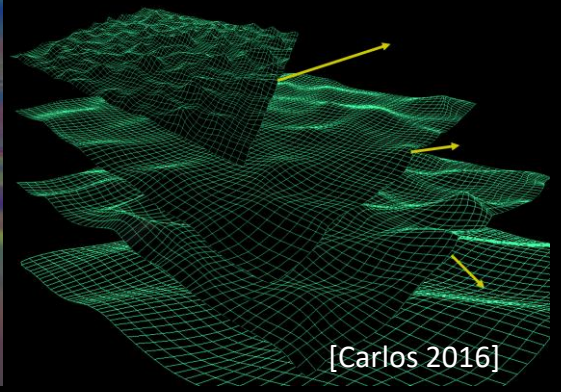
- Static
- Need a lot of artiest’s work

Spectrum Based Water Dynamics

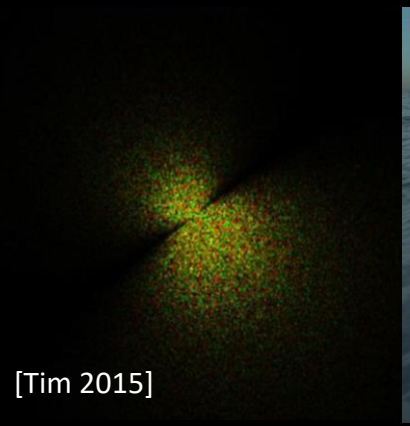


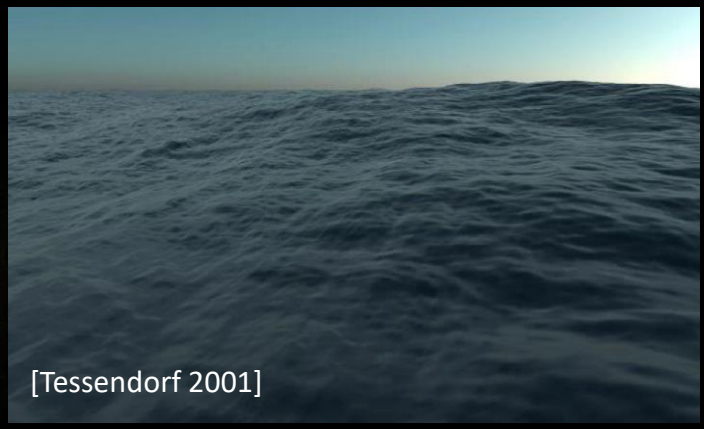
Classic Normal Map



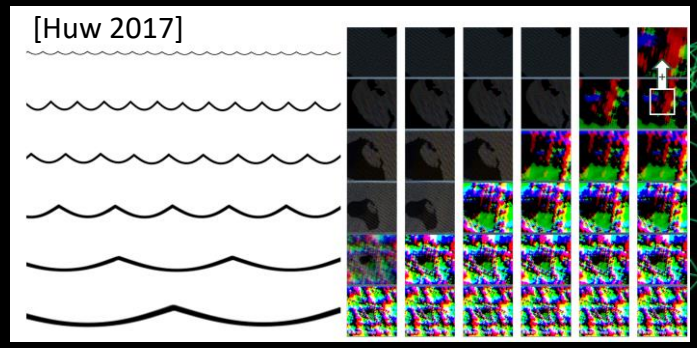


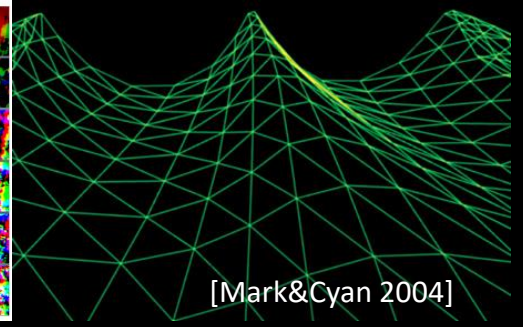
Pre-made Multiple Frequency Displacement





Oceanography Spectrum + iFFT





Trochoidal/Gerstner Wave Composition

Spectrum Based Water Dynamics

Multiple Frequency Gerstner wave for finite water body

FFT for Infinite Ocean

FFT for Infinite Ocean

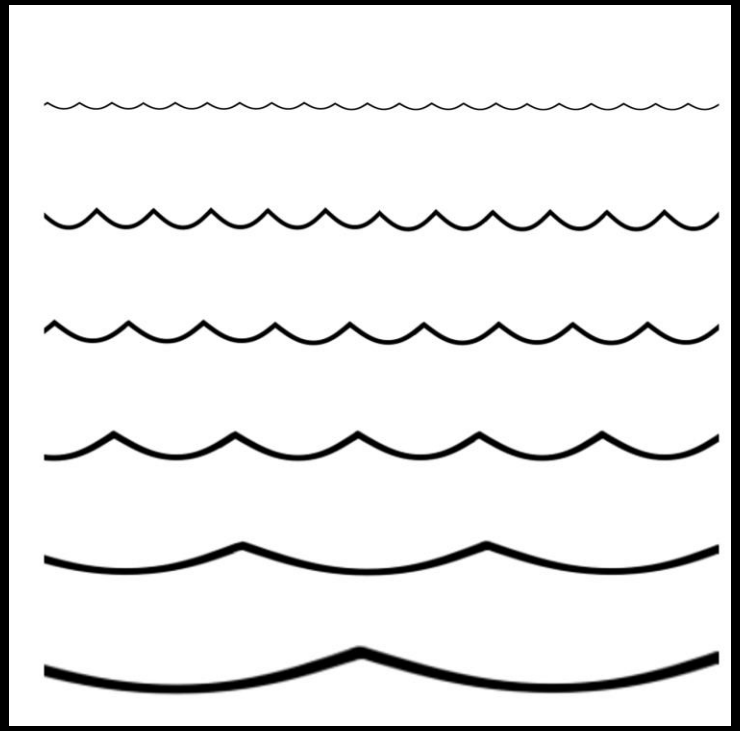
- Wind force driven spectrum
- Cascade approach to avoid tiling artifact
- Highly Detailed

Gerstner Wave Composition

for finite water body

Why?

- Intuitive
- Easy to get same result on both CPU and GPU
- More artistic control
- No texture asset needed for wave displacement



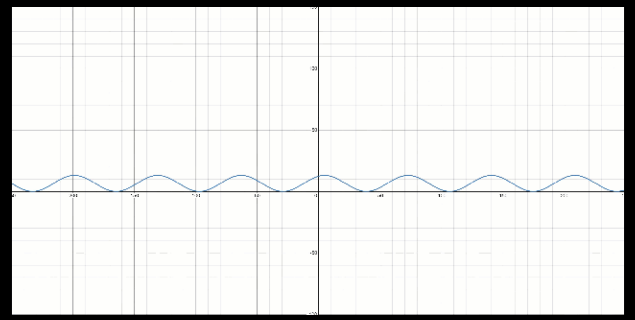
[Huw 2017]

Gerstner Wave

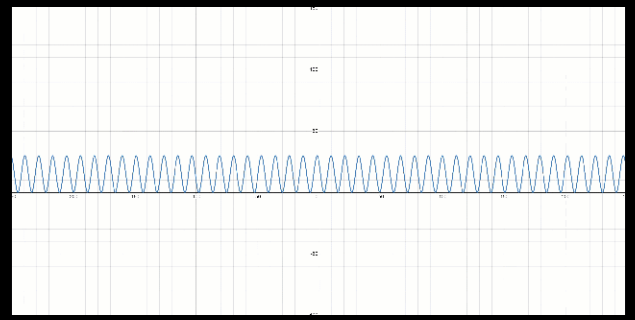
$$y = A\cos\left(\frac{2\pi}{L} \cdot \mathbf{x}_0 - \omega t\right)$$

$$\mathbf{x} = \mathbf{x}_0 + QA\sin\left(\frac{2\pi}{L} \cdot \mathbf{x}_0 - \omega t\right)$$

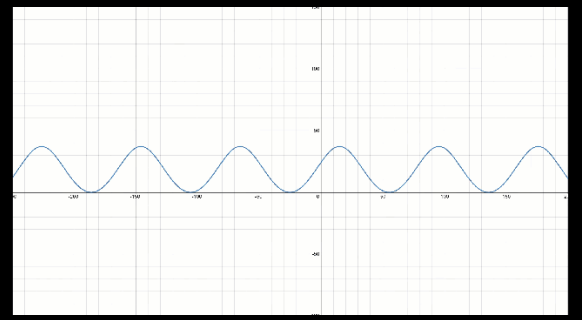
Amplitude



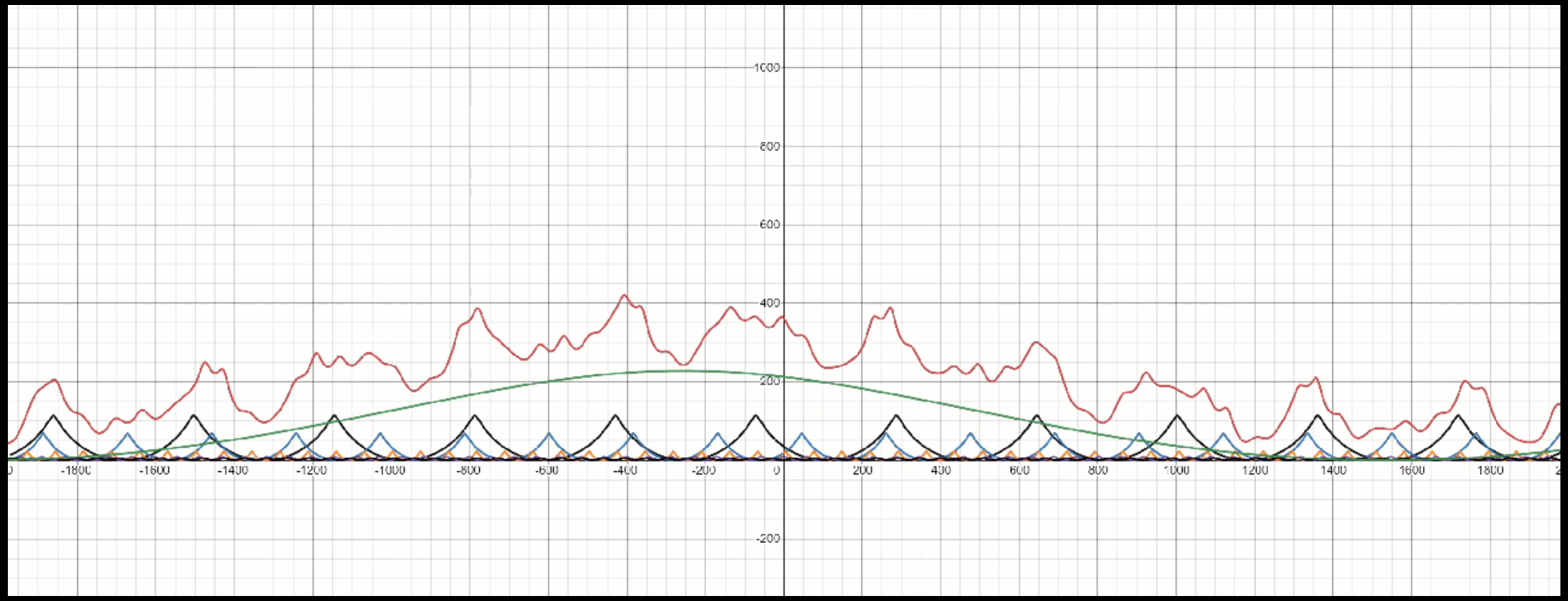
Wavelength



Steepness



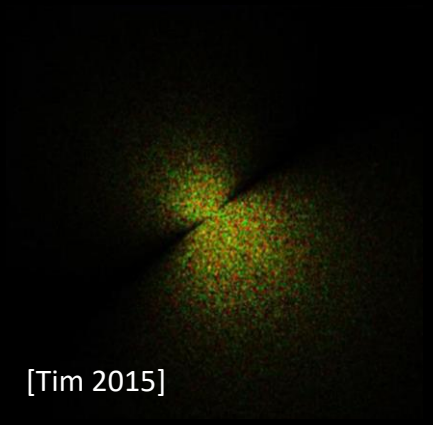
Multi-Frequency Gerstner Wave Composition



6 Gerstner Wave Summation

Multi-Frequency Gerstner Wave Composition

How to extend it to 2D plane ?



[Tim 2015]

[Epic 2020]

Multi-Frequency Gerstner Wave Composition

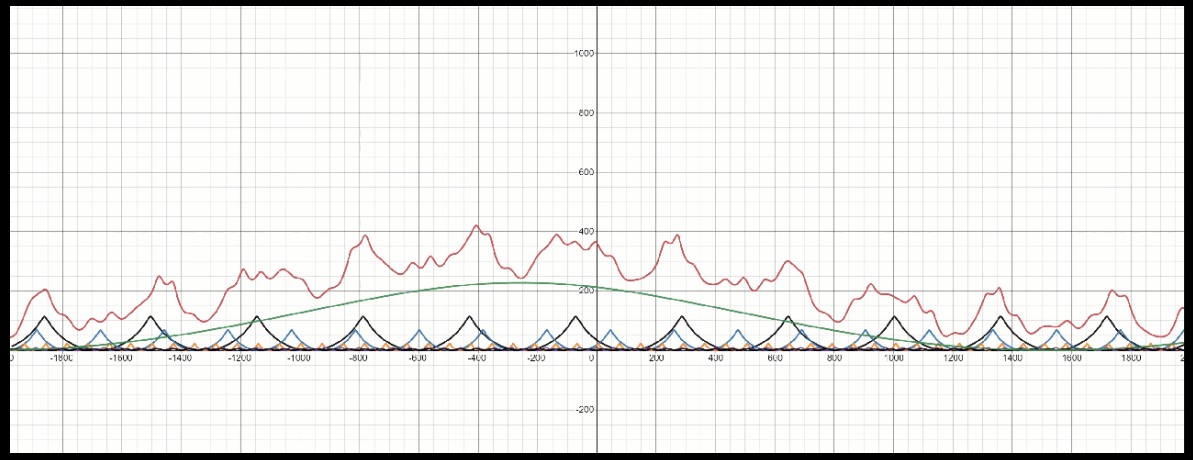
What about rivers ?

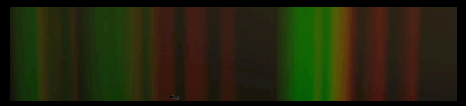
River needs **FLOW MAPS**

Multi-Frequency Gerstner Wave Composition

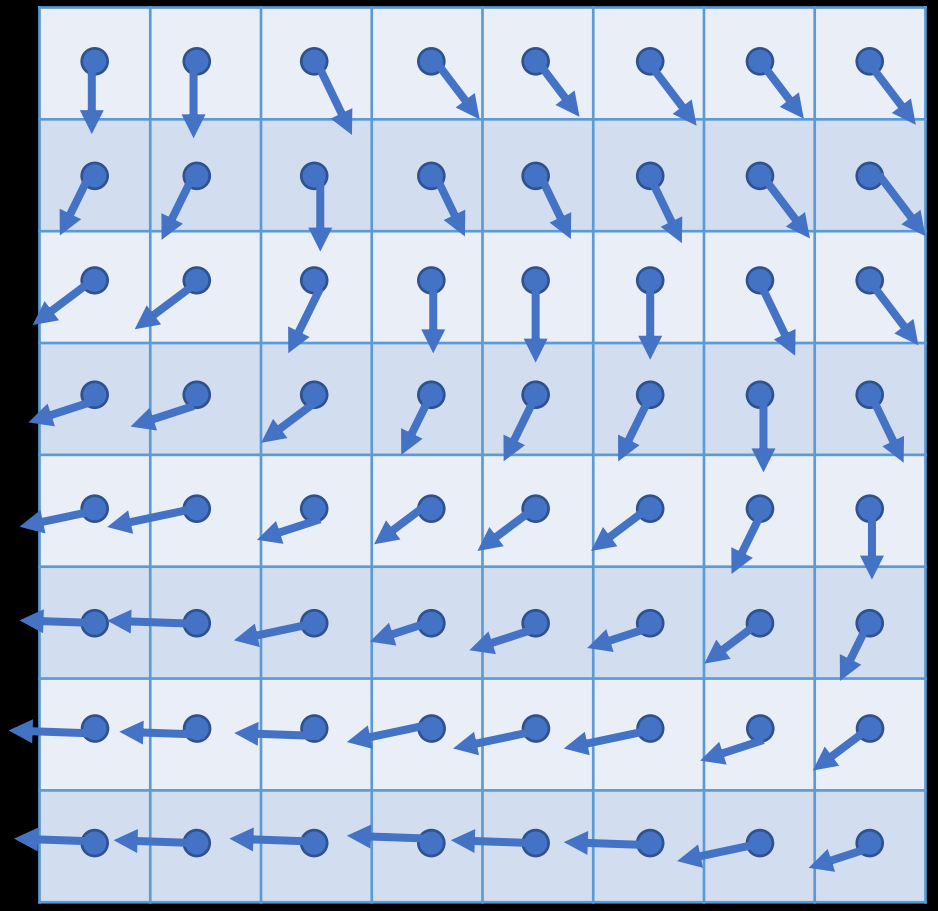
1D Wave Profile + Direction Spreading Function

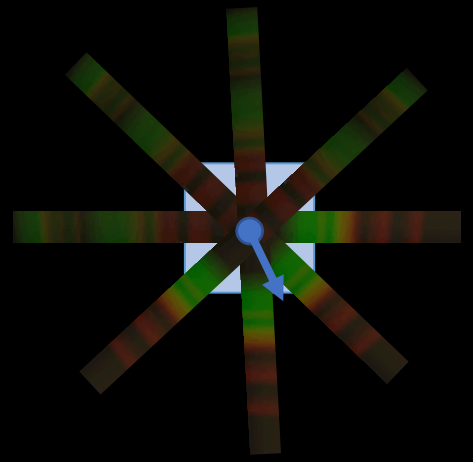
[Stefan J. 2018]





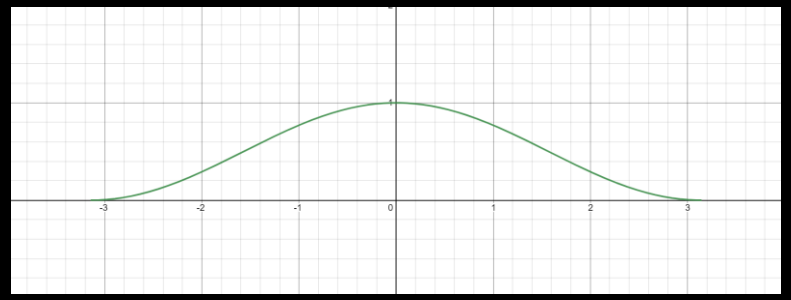
Multi-Frequency Gerstner Wave Composition

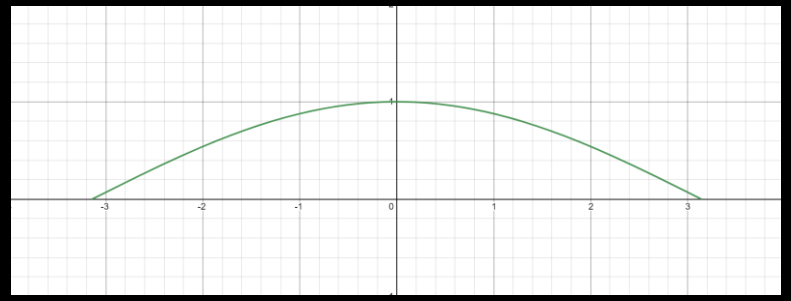


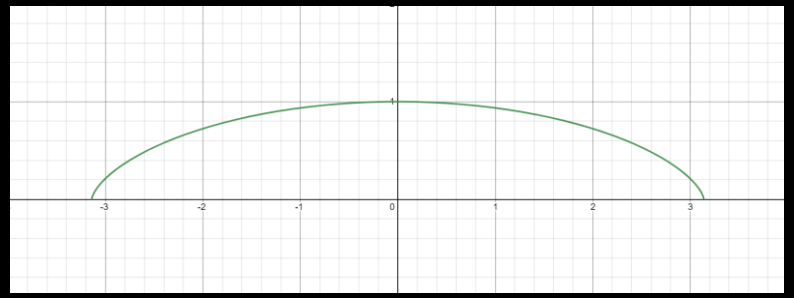


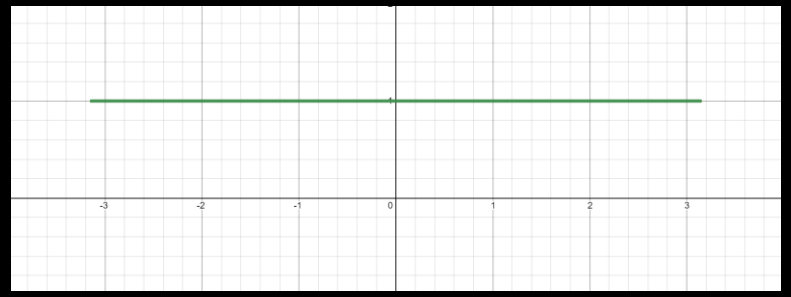
Multi-Frequency Gerstner Wave Composition

Direction Spreading Function



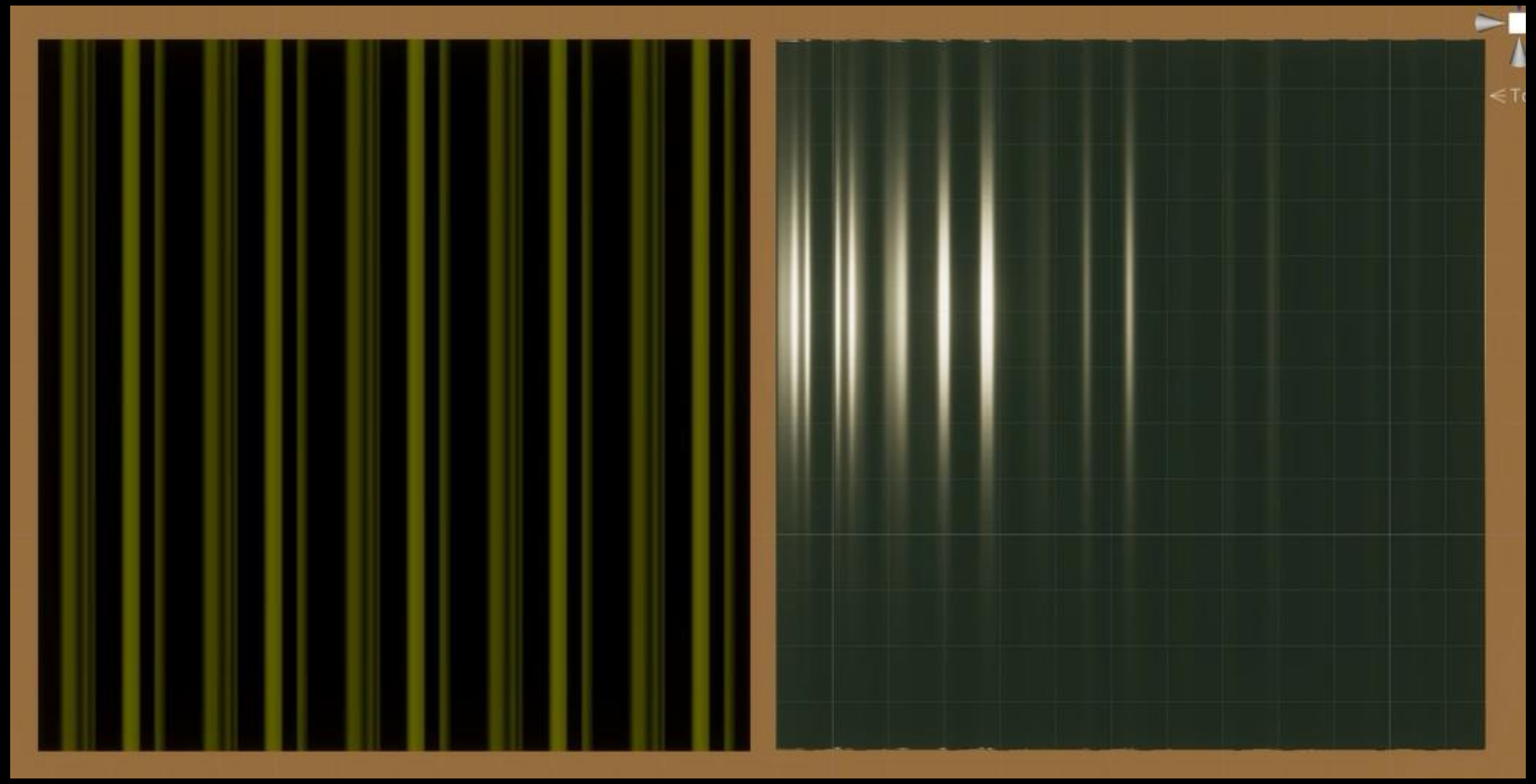






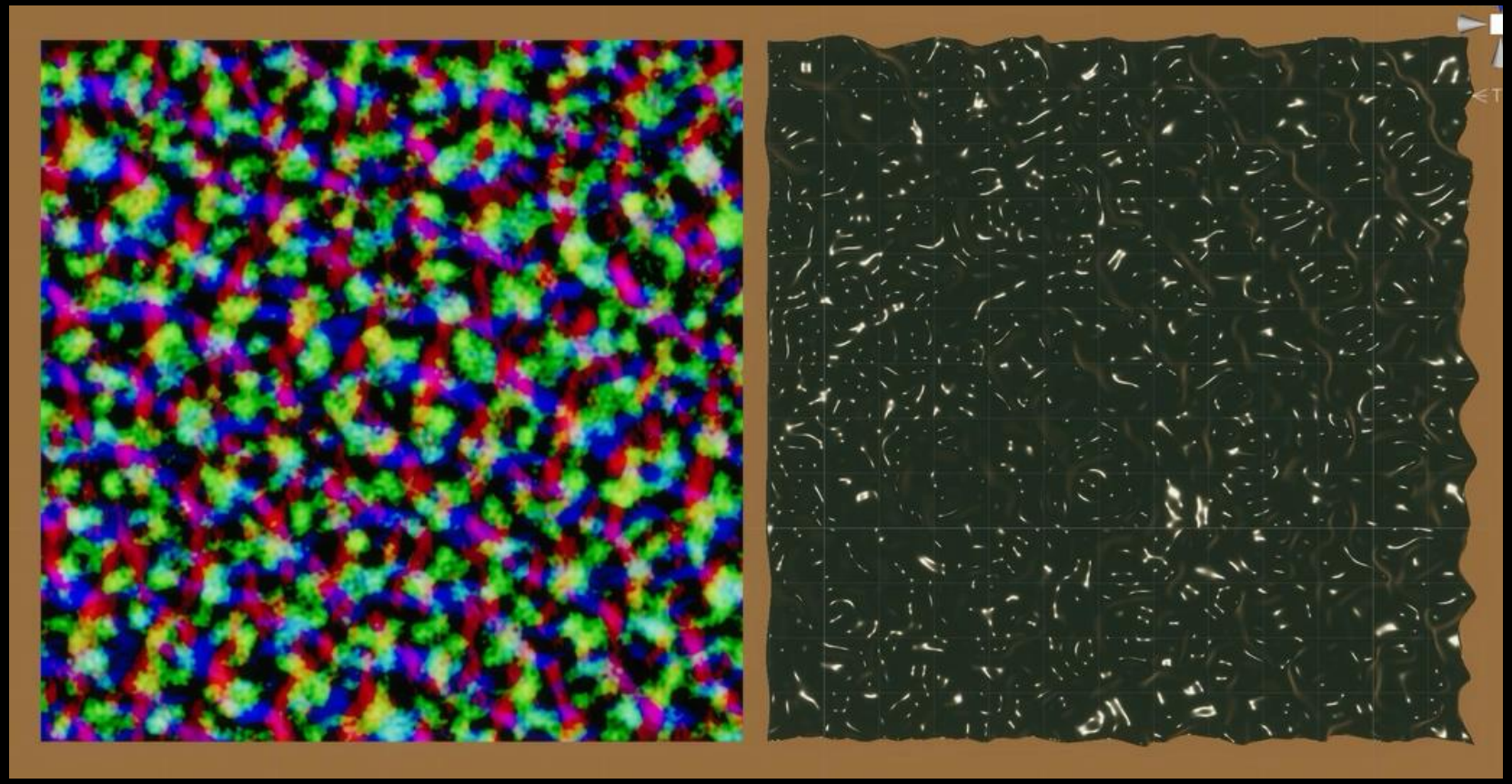
Multi-Frequency Gerstner Wave Composition

1D Wave Profile



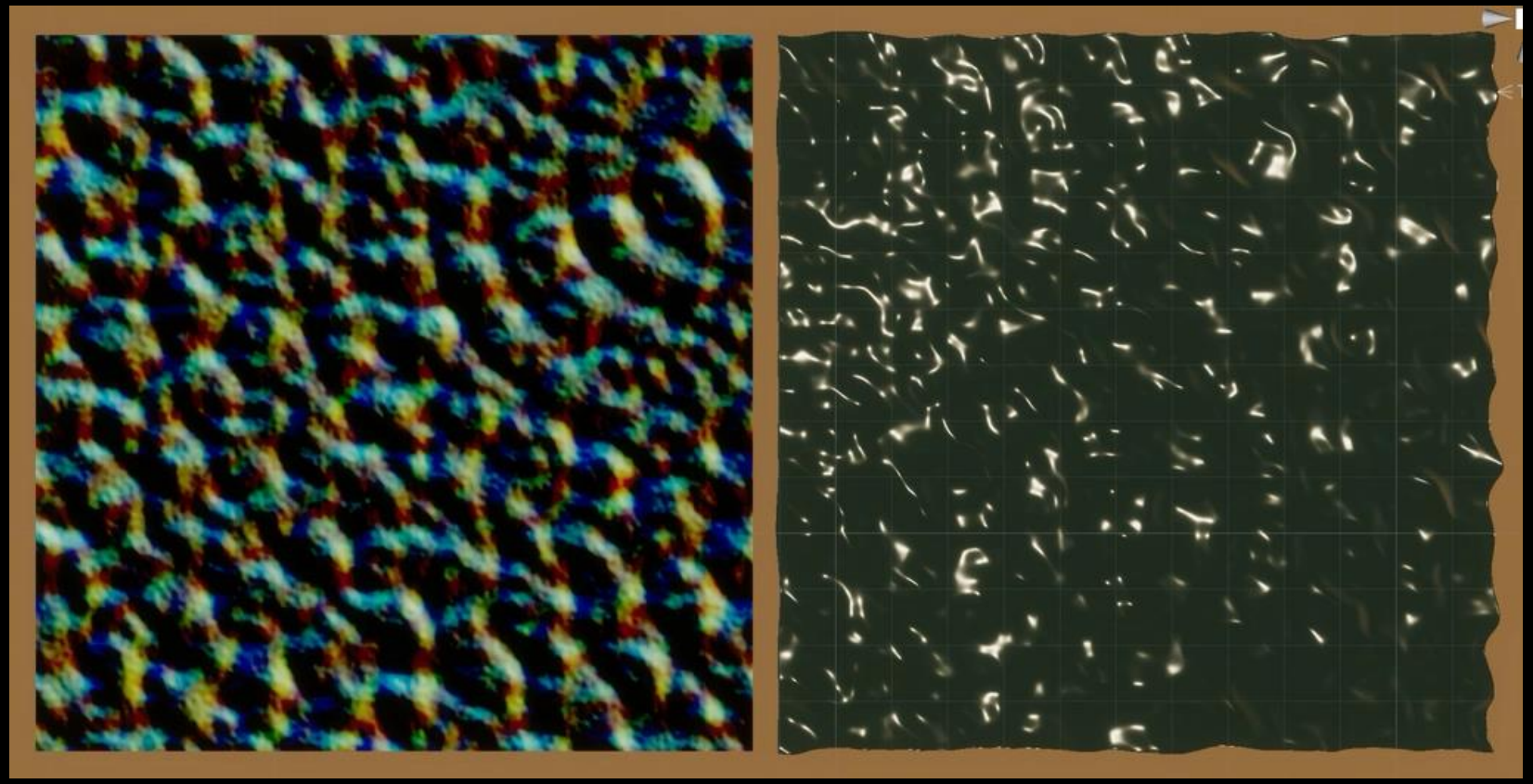
Multi-Frequency Gerstner Wave Composition

Integration

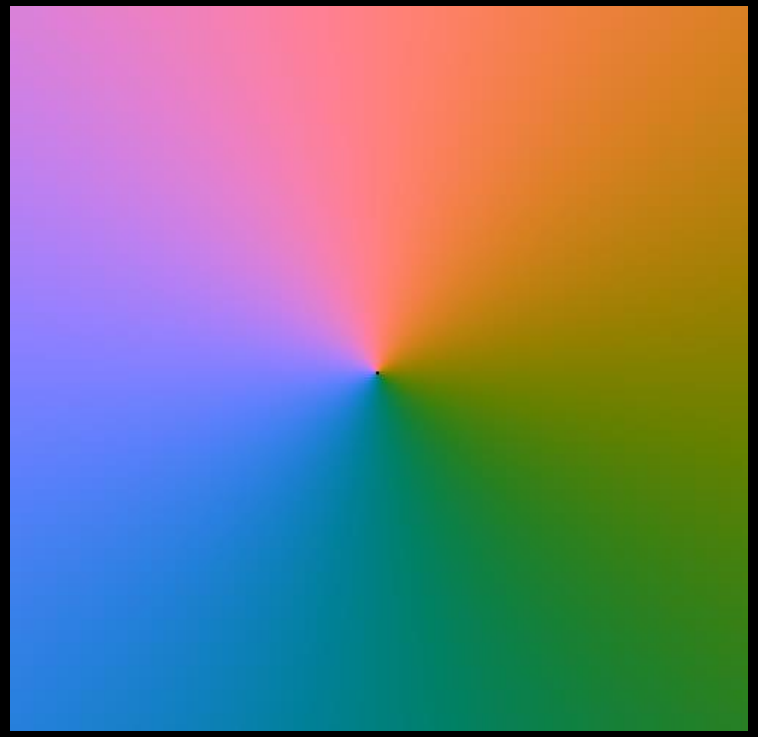


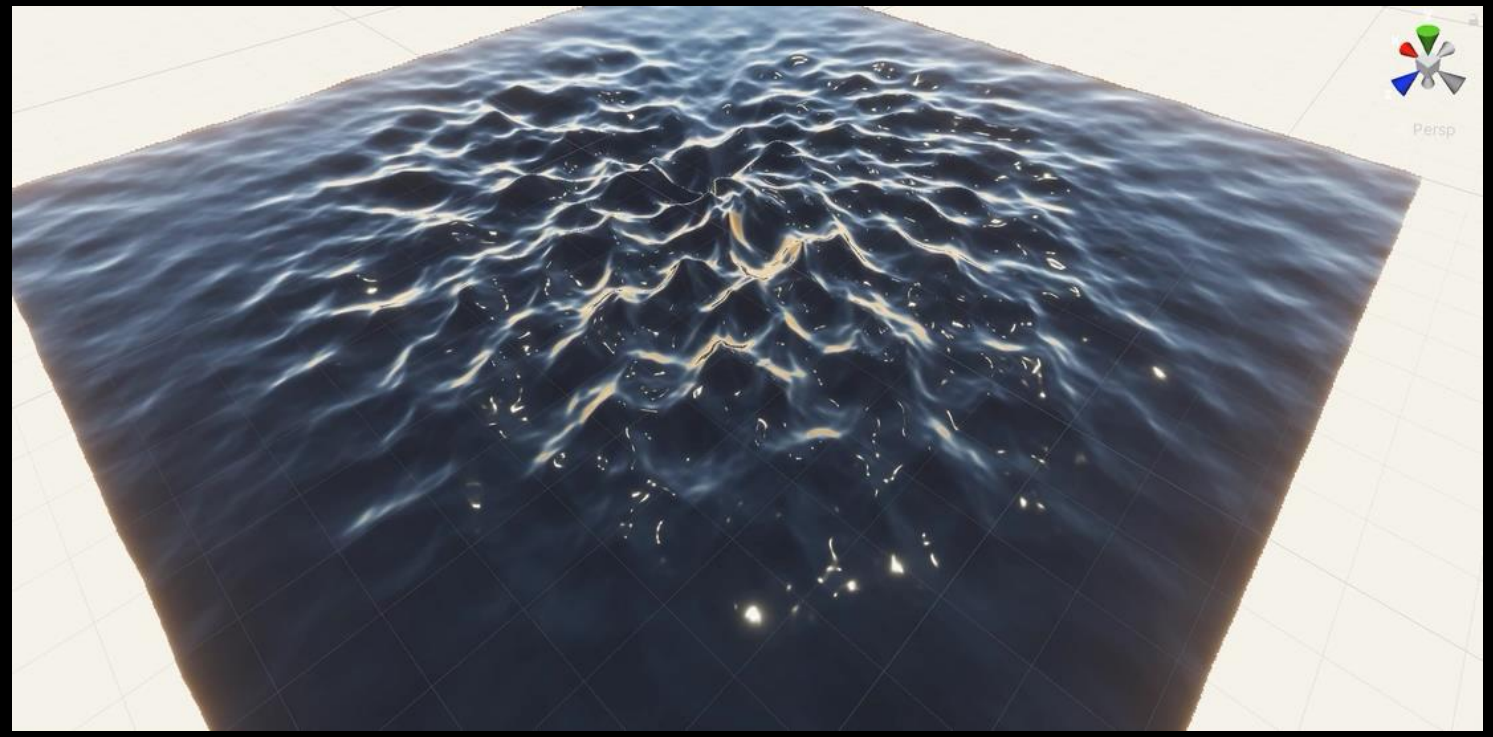
Multi-Frequency Gerstner Wave Composition

Direction Spreading

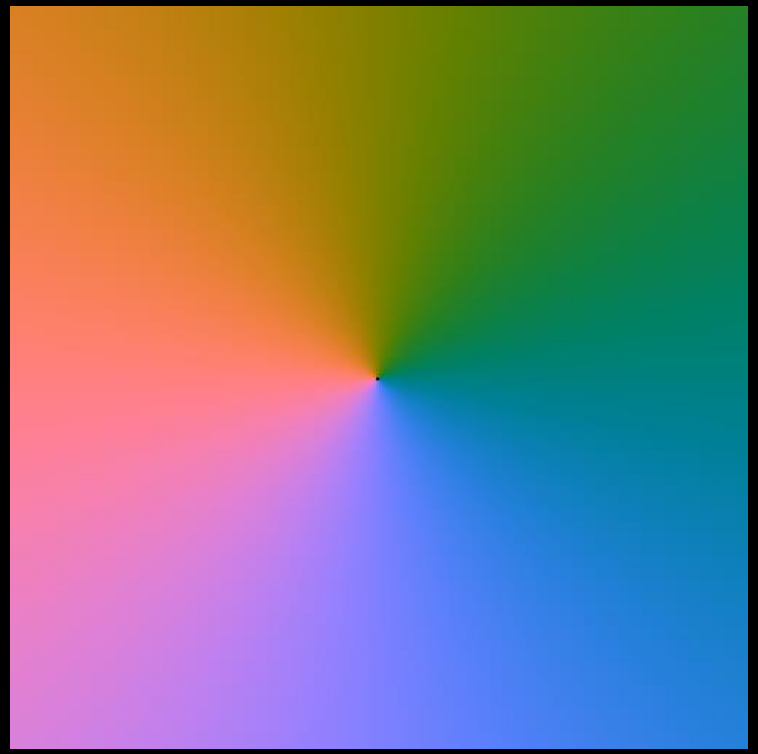


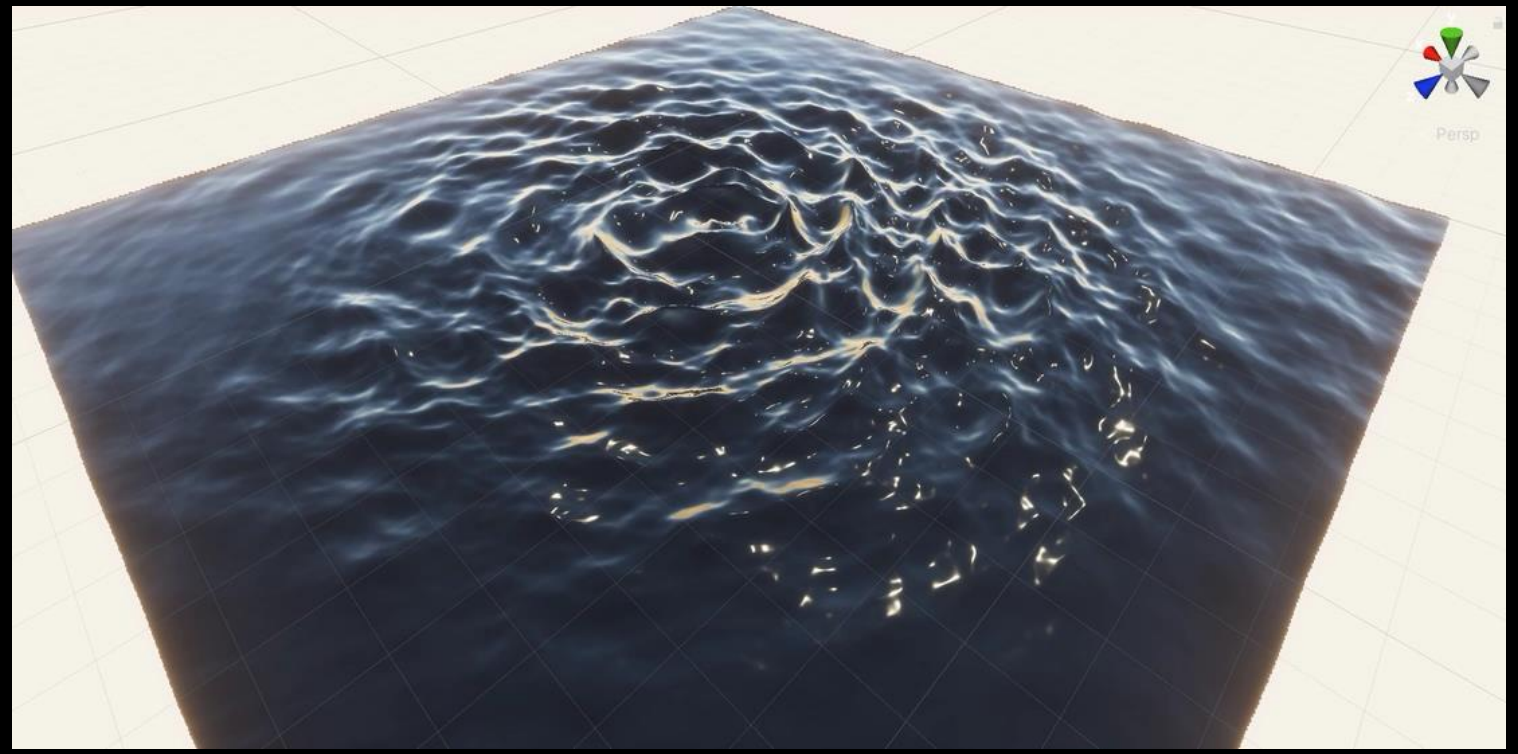
Multi-Frequency Gerstner Wave Composition





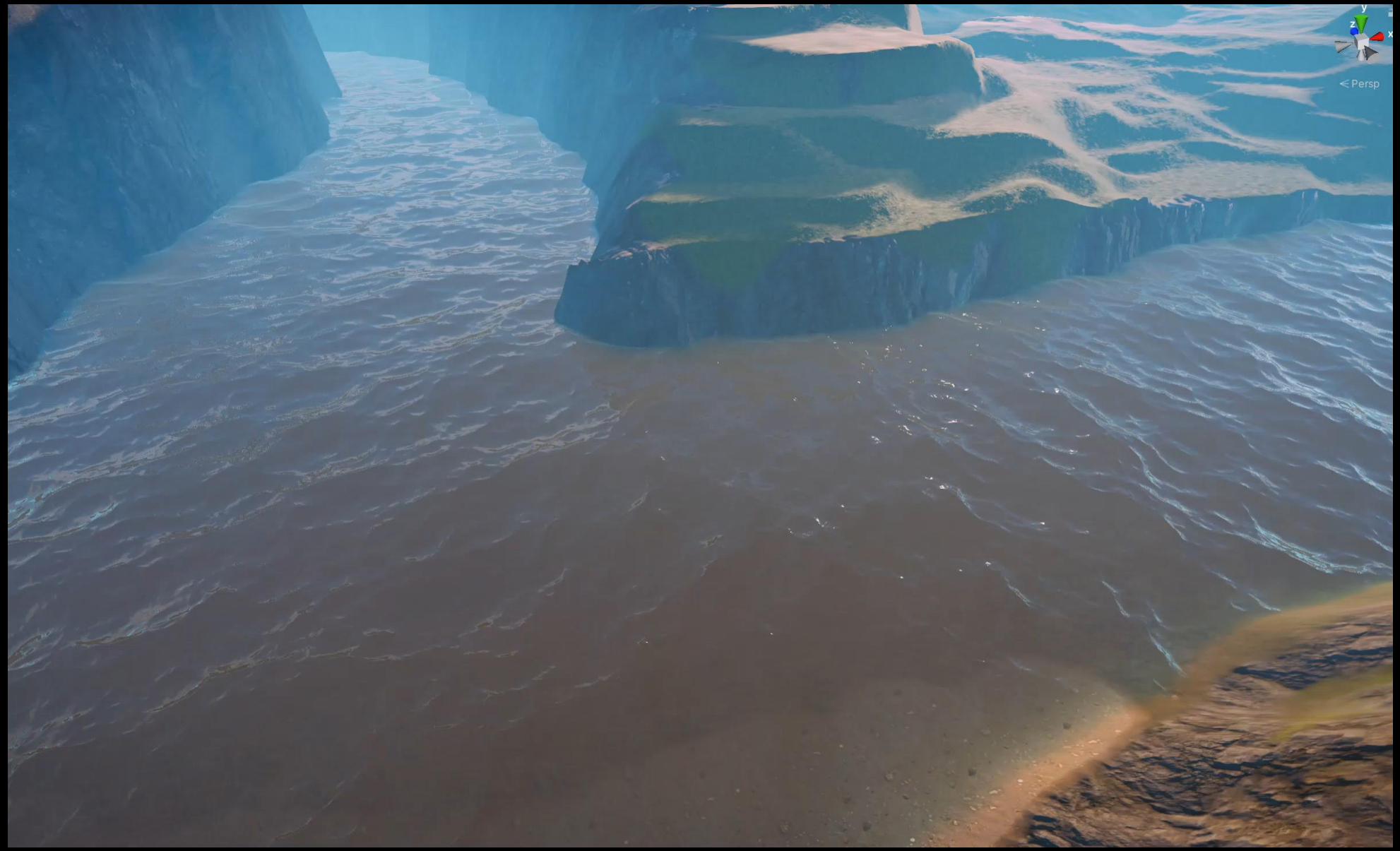
Multi-Frequency Gerstner Wave Composition





Flow Map Generation

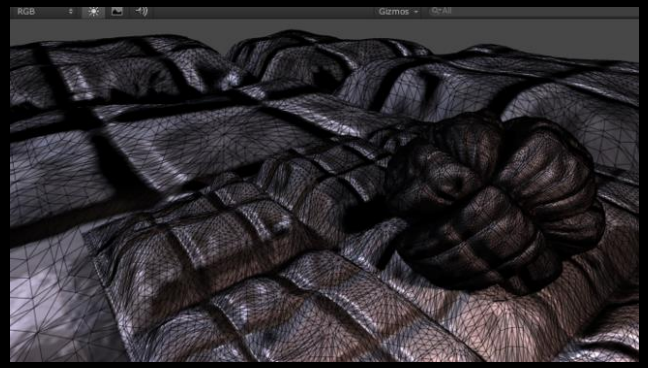
In-engine Simulation	DCC Tools
LBM Based	Houdini Photoshop Etc.


y
z
x
Persp

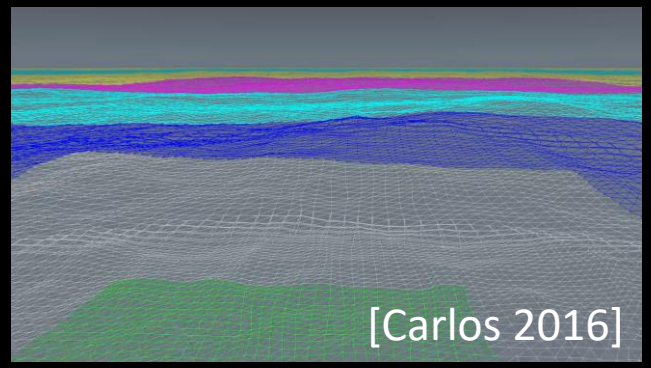

y
z
Persp

Tessellation

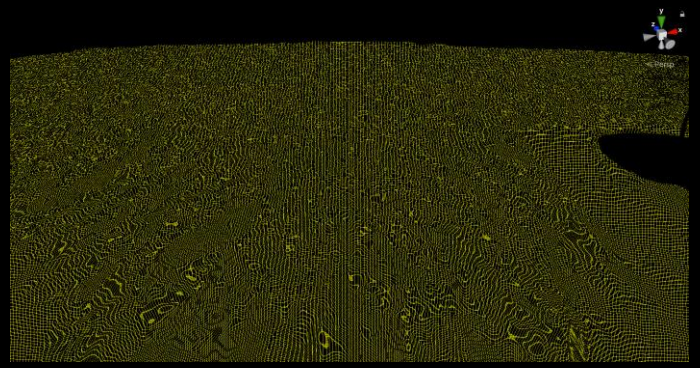
Hardware Tessellation



Cascade Grid

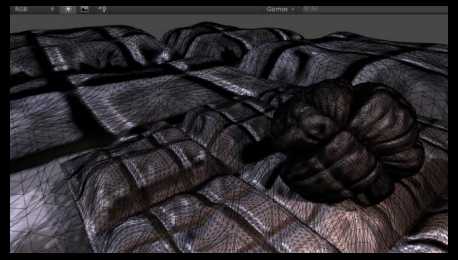


Screen Space Tessellation



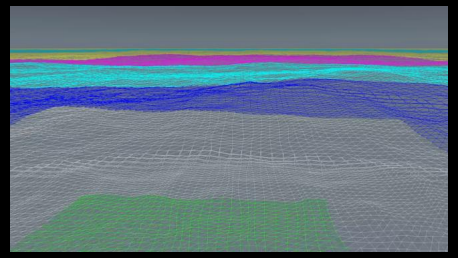
Tessellation

Hardware Tessellation



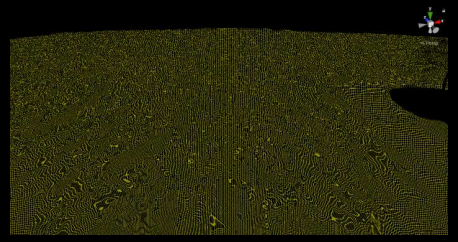
- GPU
- Mannually control density
- Traditional Art Pipeline

Cascade Grid



- CPU & GPU
- Mannually control density
- Extra Art Pipeline

Screen Space Tessellation

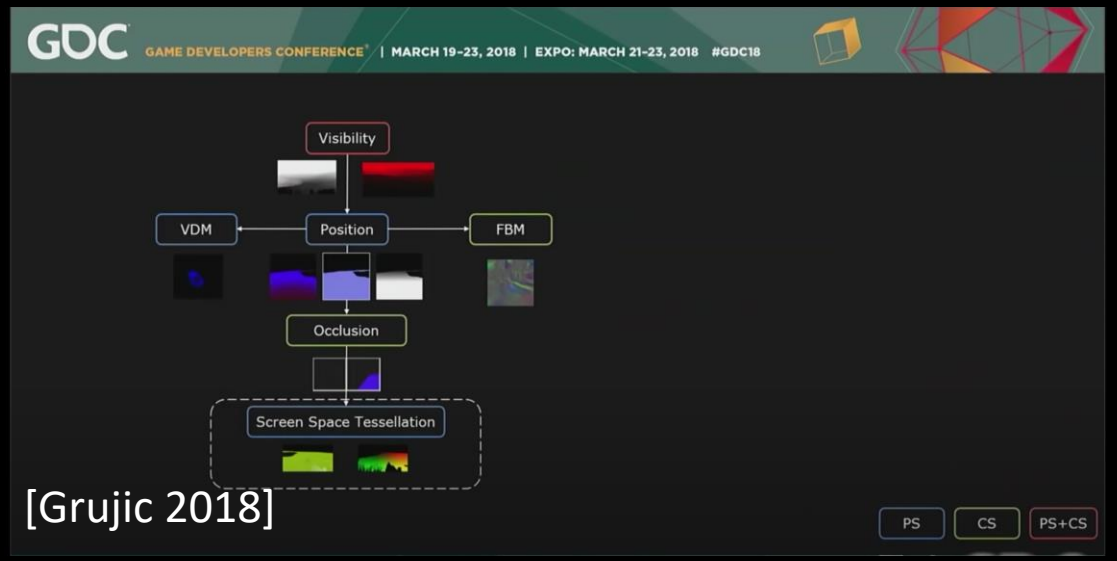


- GPU
- Costant density in screen space
- Traditional Art Pipeline

Screen Space Tessellation Based Water Render Pipeline

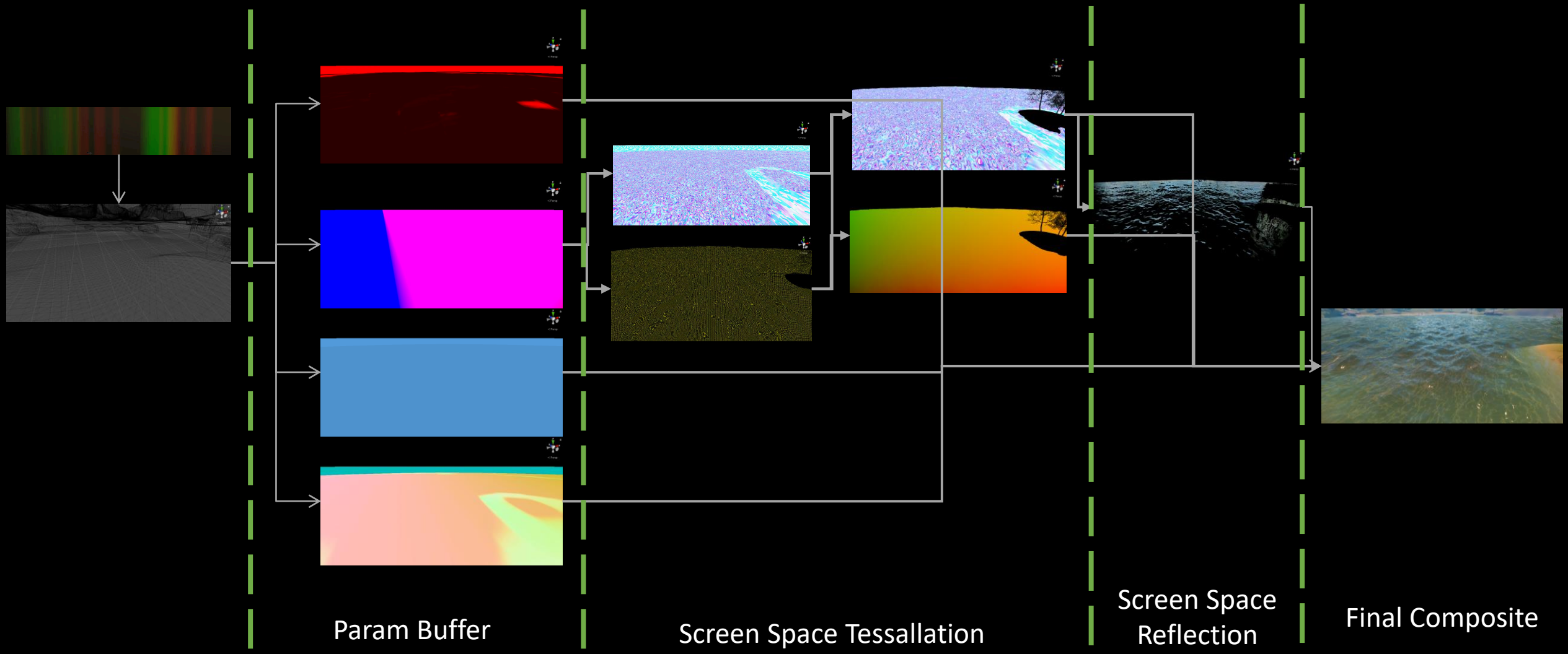
Pretty Similar to [Grujic 2018]

Deferred Shading Pipeline

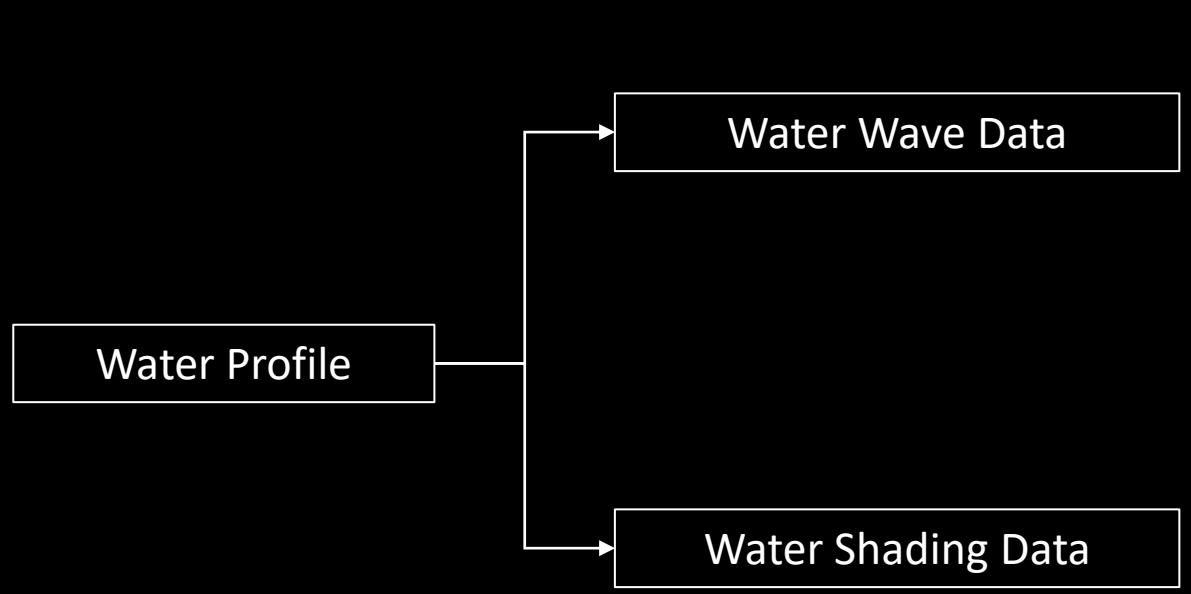


[Grujic 2018]

Screen Space Tessellation Based Water Render Pipeline



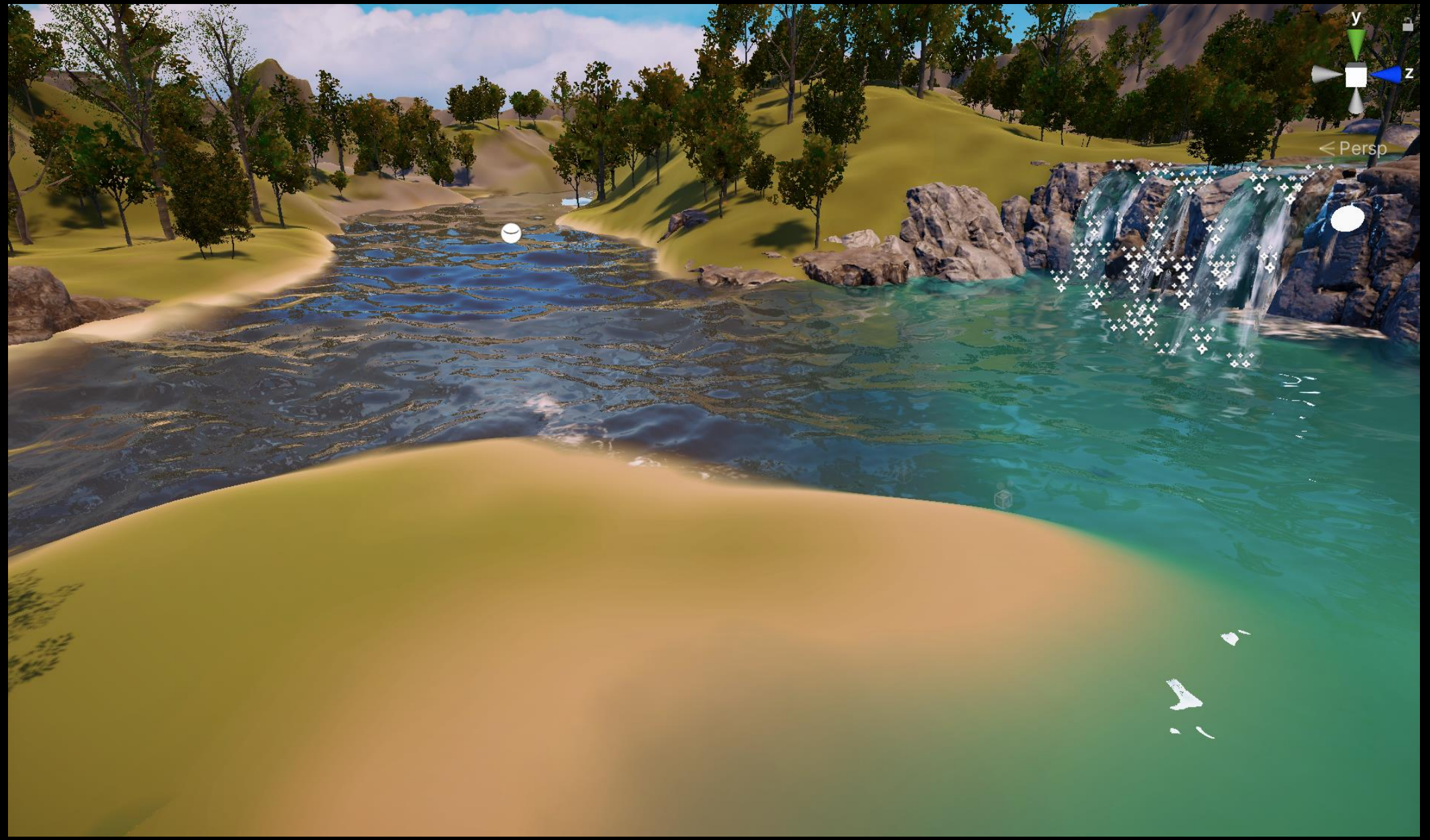
Water Data



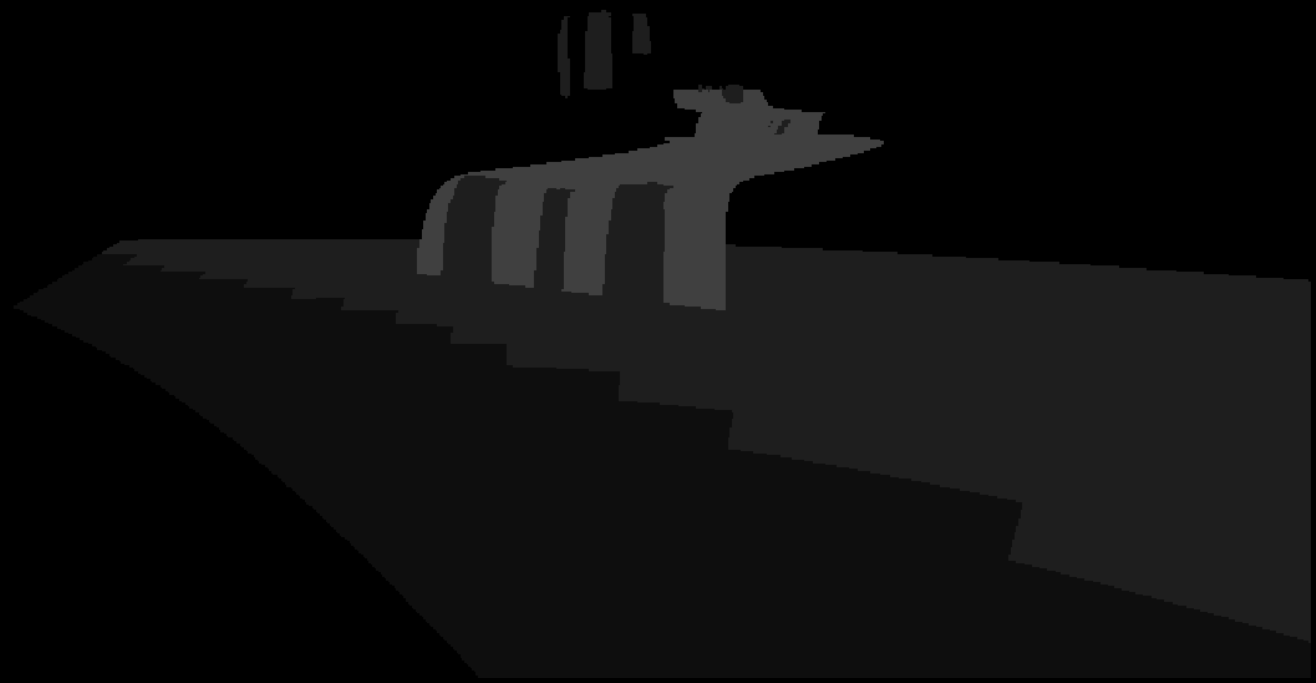

```
struct WaveIndex
{
    int waveDataOffset;
    int waveDataLength;
};
```

```
struct WaterMaterialData
{
    float4 flowParams;
    float4 foamParams;
    float4 floatObjParams;
    half4 scatteringColor;
    half4 absortionColor;
    int arrIndex;
};
```

```
struct WaveData
{
    float waveLength;
    float waveAmplitude;
    float steepness;
};
```


y
z
Persp

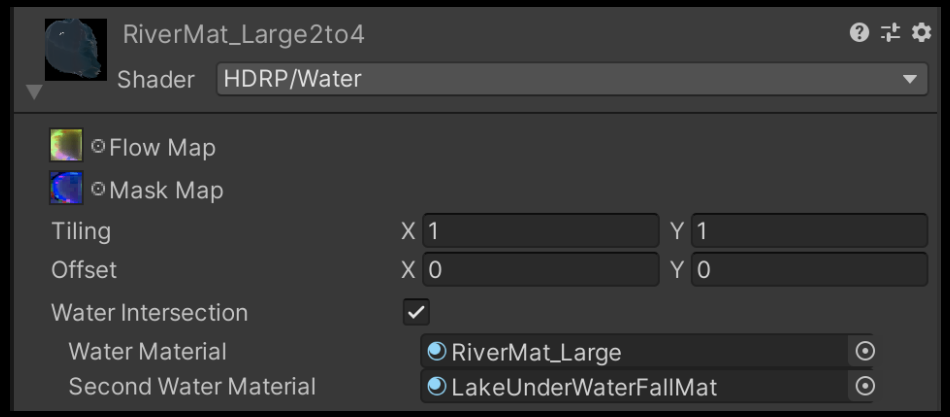
Water Data

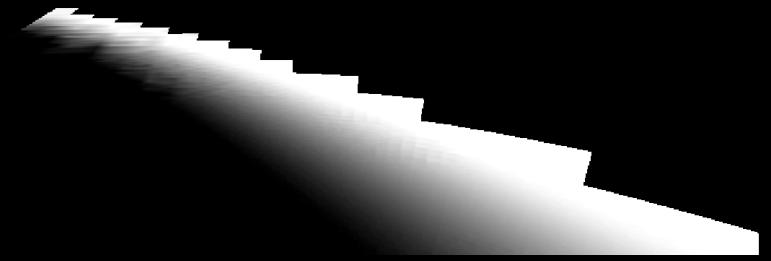


Water Profile ID

Water Intersection

Water Intersection Mask





Support both water - water and water – ocean intersection

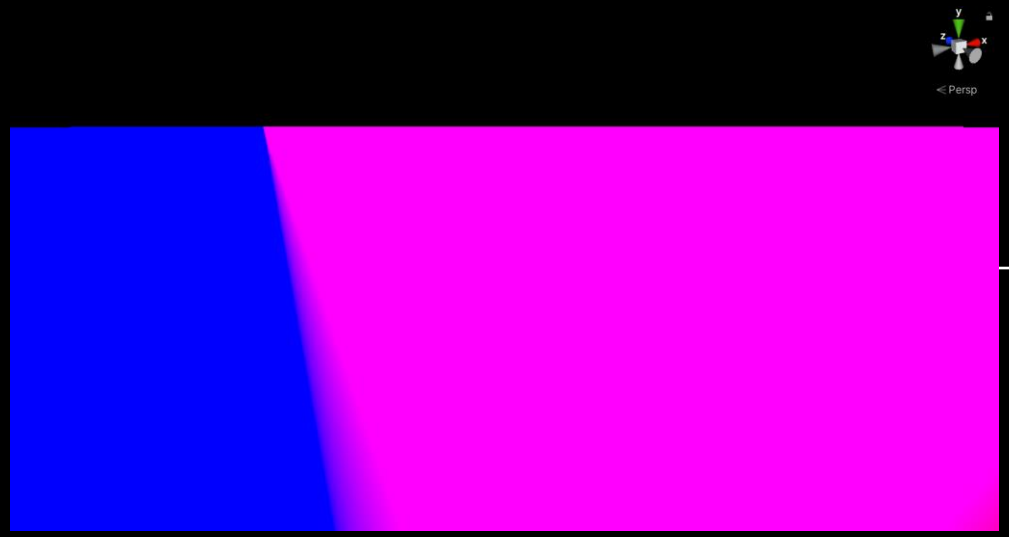
Pack Water Profile ID to Param Buffer

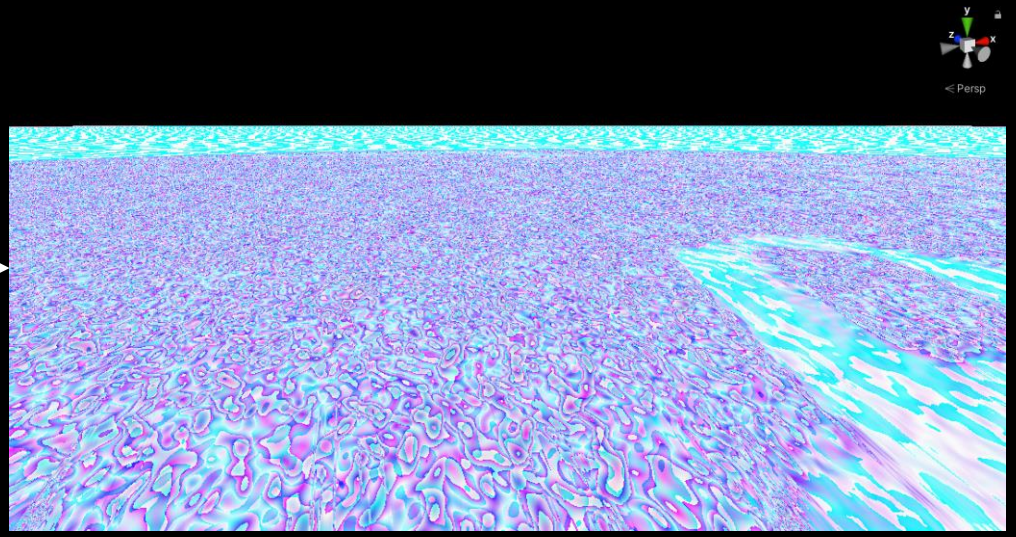
```
void EncodeIntoWaterBuffer(WaterSurfaceData data, out float4 outWaterBuffer0, out float4 outWaterBuffer1, out float3 outWaterBuffer2, out float4 outWaterBuffer3)
{
    // This encode normalWS and PerceptualSmoothness into GBuffer1
    outWaterBuffer0 = float4(data.masks.rgb, PackIntInt8Bit( i0: data.profileId,  i1: data.profileId1,  maxi: 16));
    outWaterBuffer1 = float4(data.positionWS, 1);
    outWaterBuffer2 = data.bakedGI;
    outWaterBuffer3 = data.flowDir;
}
```

Support both water - water and water – ocean intersection

Water Intersection

Water Normal Generation





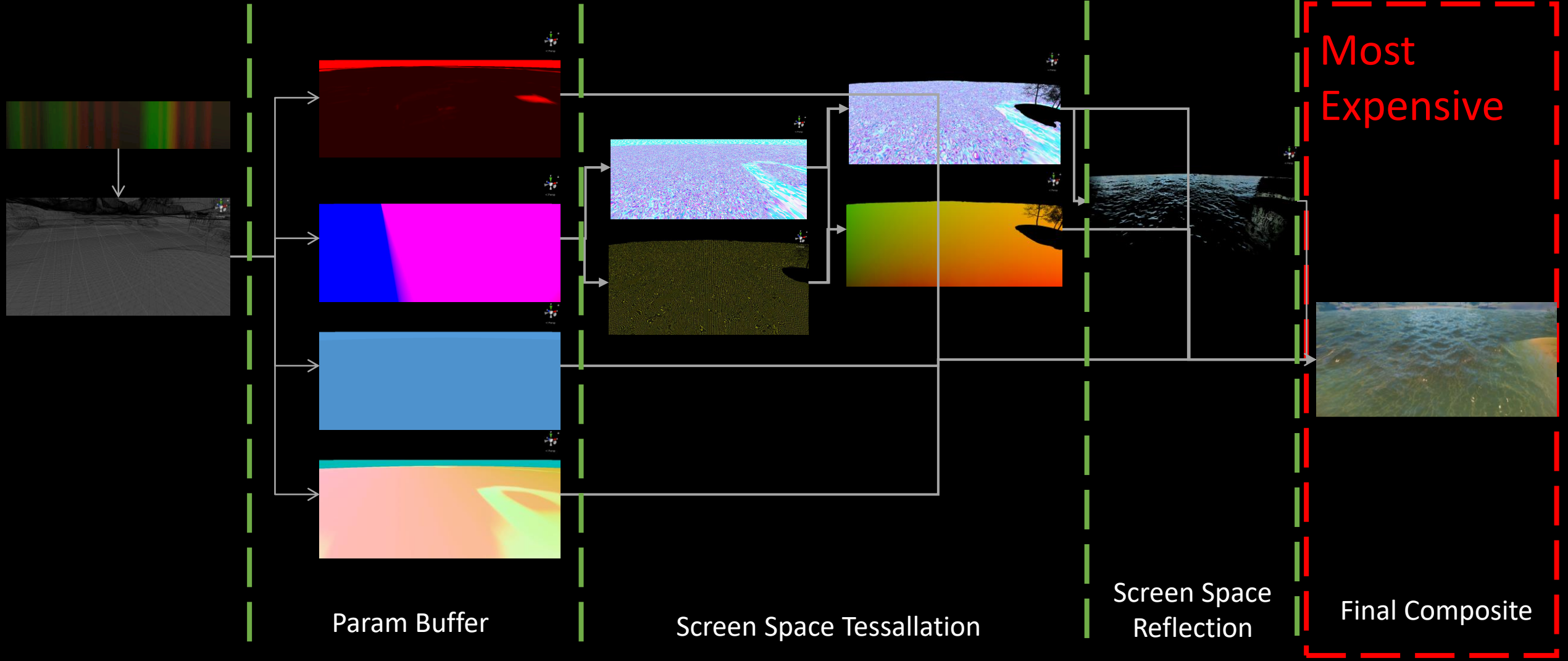
Why?

Water Normal Generation

Gerstner Wave Profile + Flow Map ⟵ Difficult to solve analytically

Other effect like shore wave

Optimization



Optimization

Final Composite

- Evaluate scattering/absorption
- Evaluate foam with flow
- Evaluate caustic
- Evaluate full lighting
- Composite with scene

Optimization

Very poor occupancy on PS4 – 10%

Lots of compute buffer read

Light data buffer

Reflection probe data buffer

Water material data buffer

Etc.

Complicated light calculation

PBR lighting

Scattering lighting

Area lights

Optimization

Full screen pixel pass to indirect compute pass

GPU scalarization

- Light data scalarization
- Water material data scalarization

1D wave profile


Persp


y
x
z
< Persp

References

[Grujic 2018] Water Rendering in “Far Cry 5”

[Carlos 2016] Rendering Rapids in Uncharted 4

[Tim 2015] Ocean simulation and rendering In War Thunder

[Tessendorf 2001] Simulating Ocean Water

[Mark&Cyan 2004] Effective Water Simulation from Physical Models

[Huw 2017] Crest: Novel Ocean Rendering Techniques in an Open Source Framework

[Jean-Philippe 2018] River Editor: Water Simulation in Real-Time

[Jean, Axel, Yixin 2020] Rendering the Hellscape of Doom Eternal

[Epic 2020] Building Worlds in ‘Fortnite’ With Unreal Engine

[Stefan J. 2018] Water Surface Wavelets

[Evan W. 2016] Rendering Realtime Caustics in WebGL


THANKS