

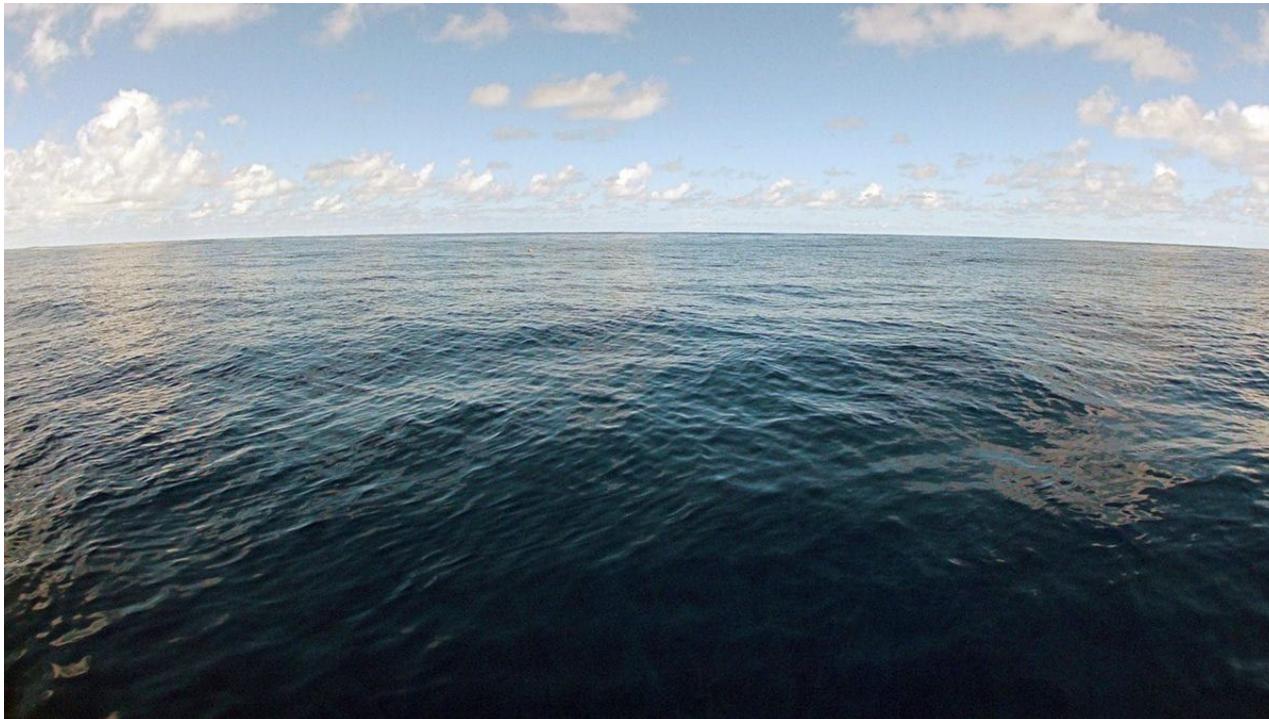
Improving the Accuracy of FFT-based GPGPU Ocean Surface Simulations

A master's thesis in computer science and engineering

Thesis and presentation by:
Jacob Eriksson and Joakim Wingård

Oceans

They are everywhere!



Oceans in...

Paintings



神奈川沖浪裏
(1831)

Movies



Battleship
(2012)

Games



Sea of Thieves
(2018)

Simulations



Delgado Maritime &
Industrial Training Center
(2019)

Oceans in...

Paintings



神奈川沖浪裏
(1831)



Manually draw

Games



Sea of Thieves
(2018)

Movies



Battleship
(2012)

Simulations



Delgado Maritime &
Industrial Training Center
(2019)

Oceans in...

Paintings



神奈川沖浪裏
(1831)



Manually draw

Fluid Simulation (+ VFX)



Movies



Battleship
(2012)

Games



Sea of Thieves
(2018)

Simulations



Delgado Maritime &
Industrial Training Center
(2019)

Oceans in...

Paintings



神奈川沖浪裏
(1831)



Manually draw

Fluid Simulation (+ VFX)



Movies



Battleship
(2012)

Games



Sea of Thieves
(2018)



How?



Simulations



Delgado Maritime &
Industrial Training Center
(2019)

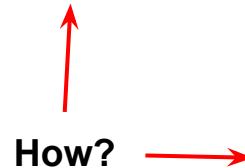
Problem: Interactive Oceans

- Unpredictable frames
- Real-time requirements (~60 FPS)
- Huge simulation volume

Games



Sea of Thieves
(2018)



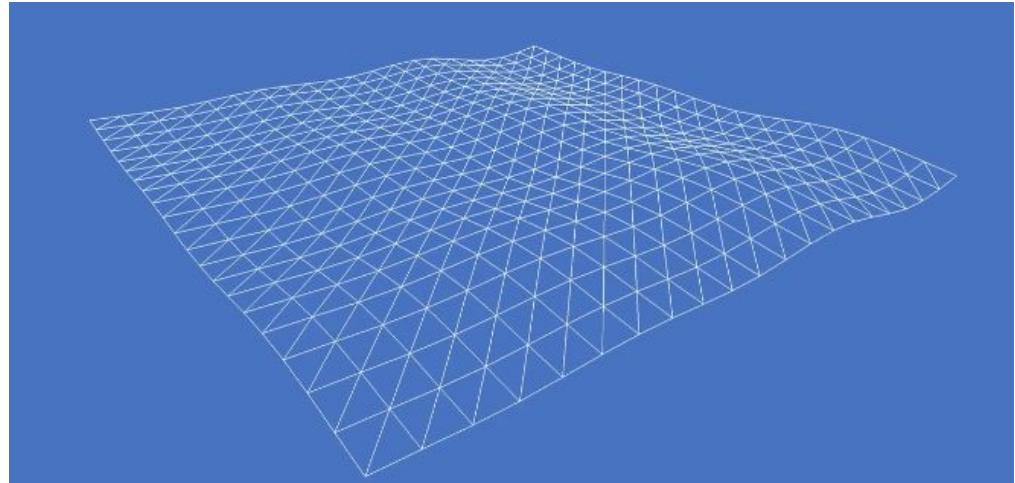
Simulations



Delgado Maritime &
Industrial Training Center
(2019)

Simplifying the Problem

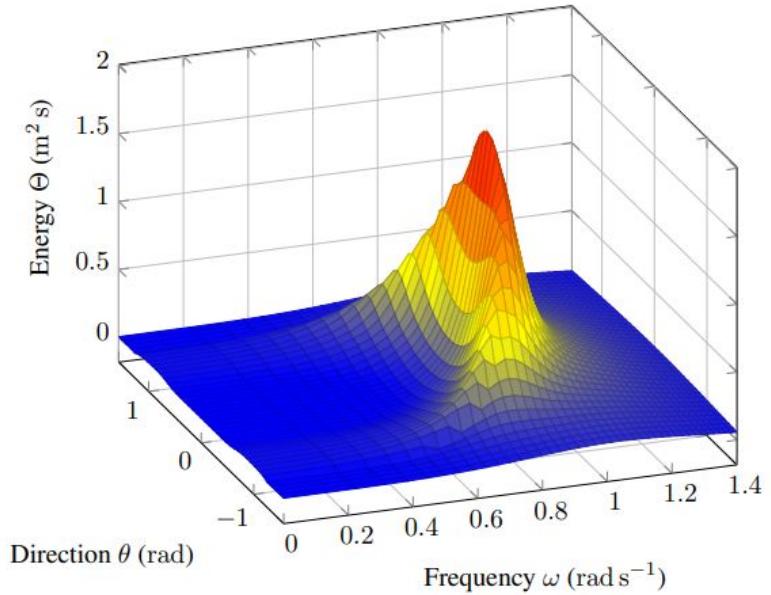
- Simplify: no depth
- Animate as heightfield
- Needs realistic animation



Triangle mesh animated with 2 sinusoids

State of the Art

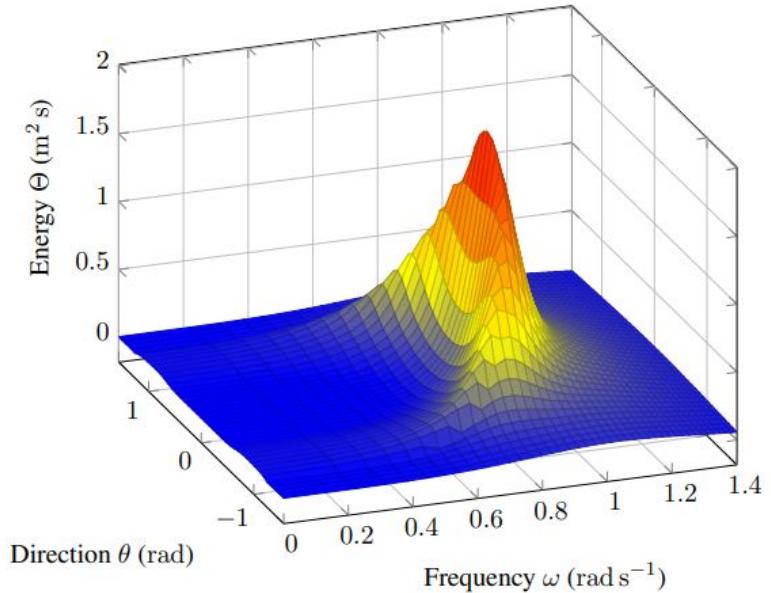
- Ocean Wave Spectrum (OWS)
- Derived from a real-life ocean
- Used to mimic that real-life ocean



Example of an OWS

State of the Art

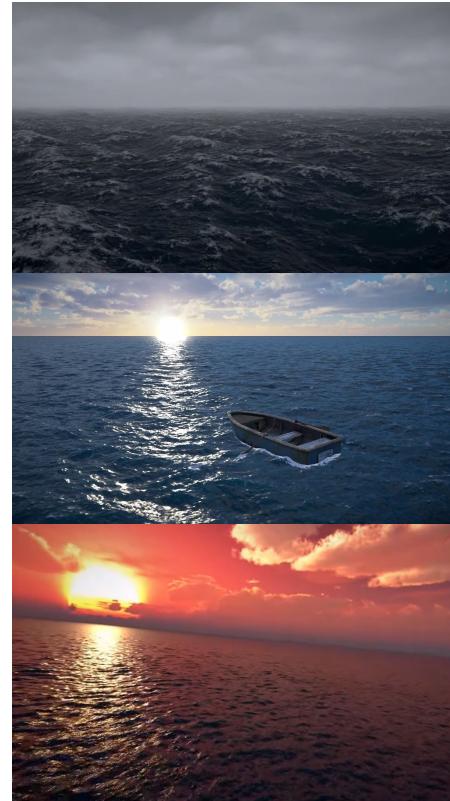
- Ocean Wave Spectrum (OWS)
 - Derived from a real-life ocean
 - Used to mimic that real-life ocean
-
- Efficiently computed using FFT
 - Fast Fourier Transform (FFT)
 - FFT efficiently computed on GPU



Example of an OWS

State of the Art

- Ocean Wave Spectrum (OWS)
 - Derived from a real-life ocean
 - Used to mimic that real-life ocean
-
- Efficiently computed using FFT
 - Fast Fourier Transform (FFT)
 - FFT efficiently computed on GPU
-
- “FFT-based GPGPU Ocean”



Examples of
FFT-based oceans

The Problem

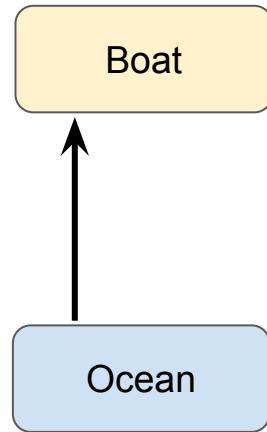


Examples of
FFT-based oceans

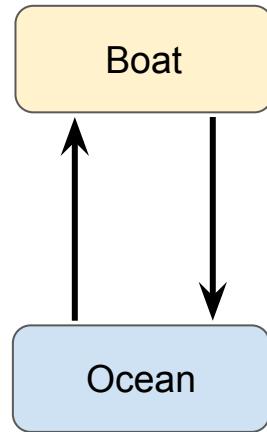
The Problem



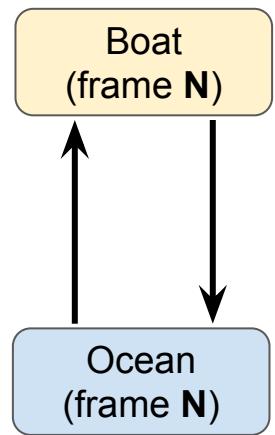
The Problem



The Problem

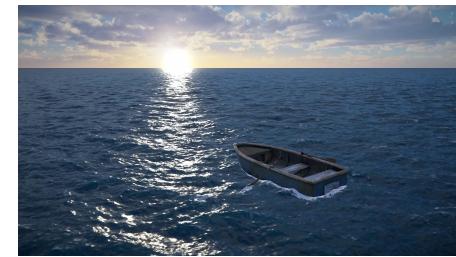
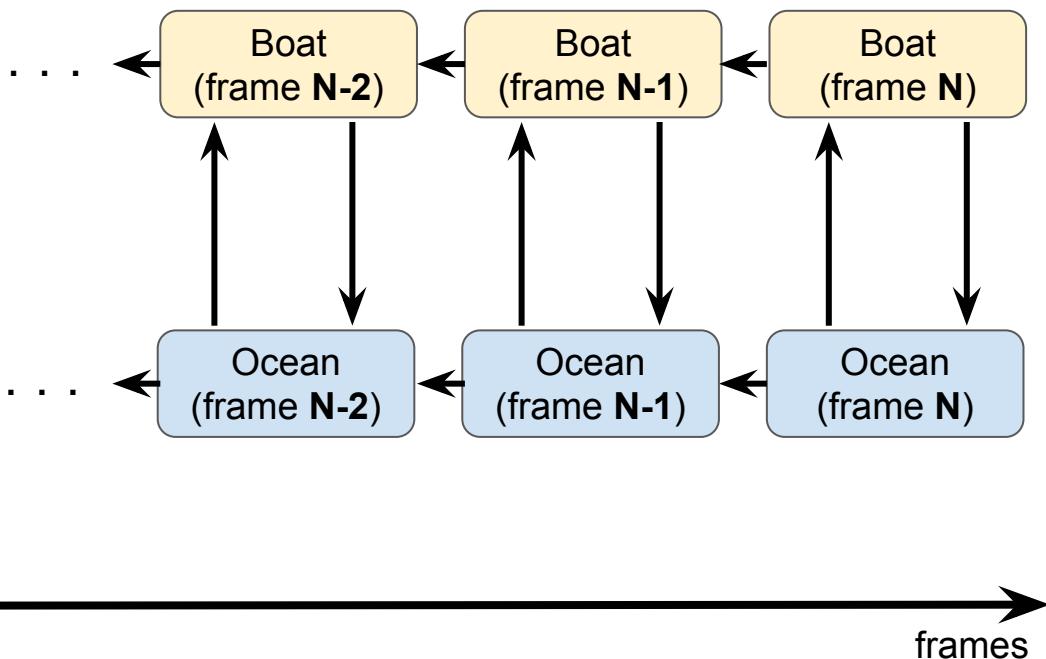


The Problem

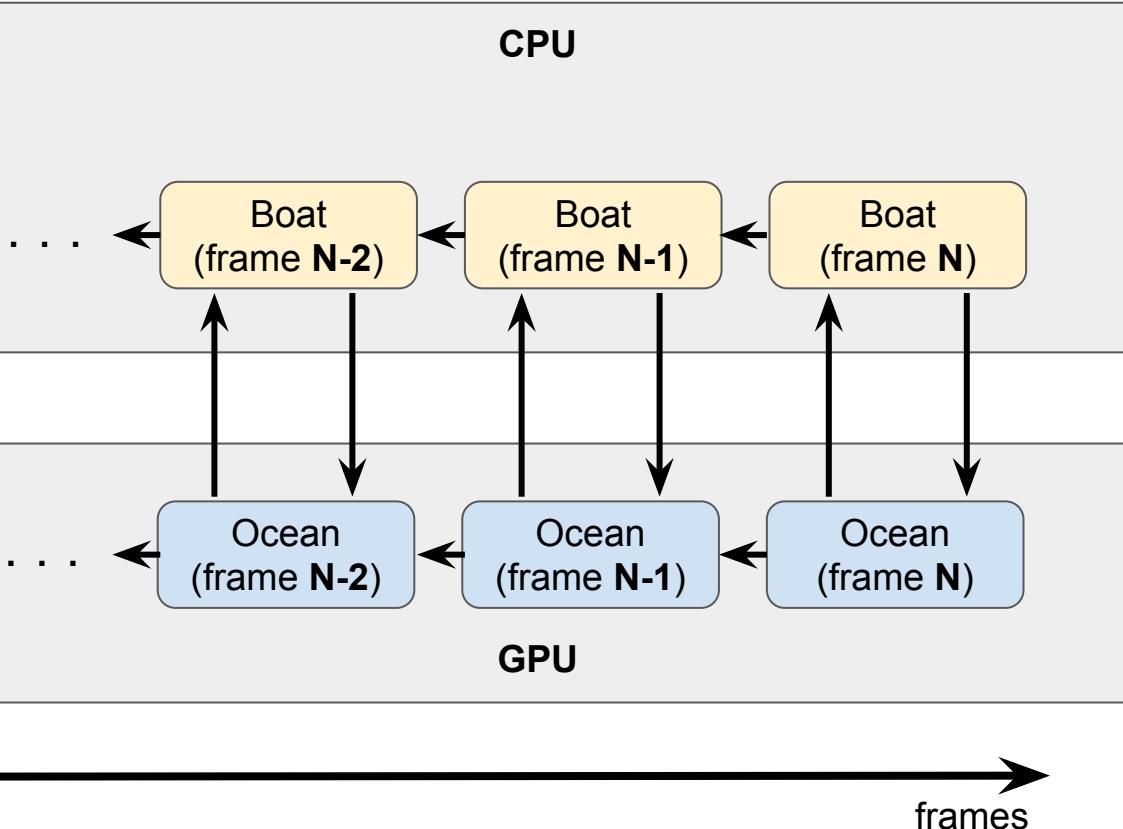


frames

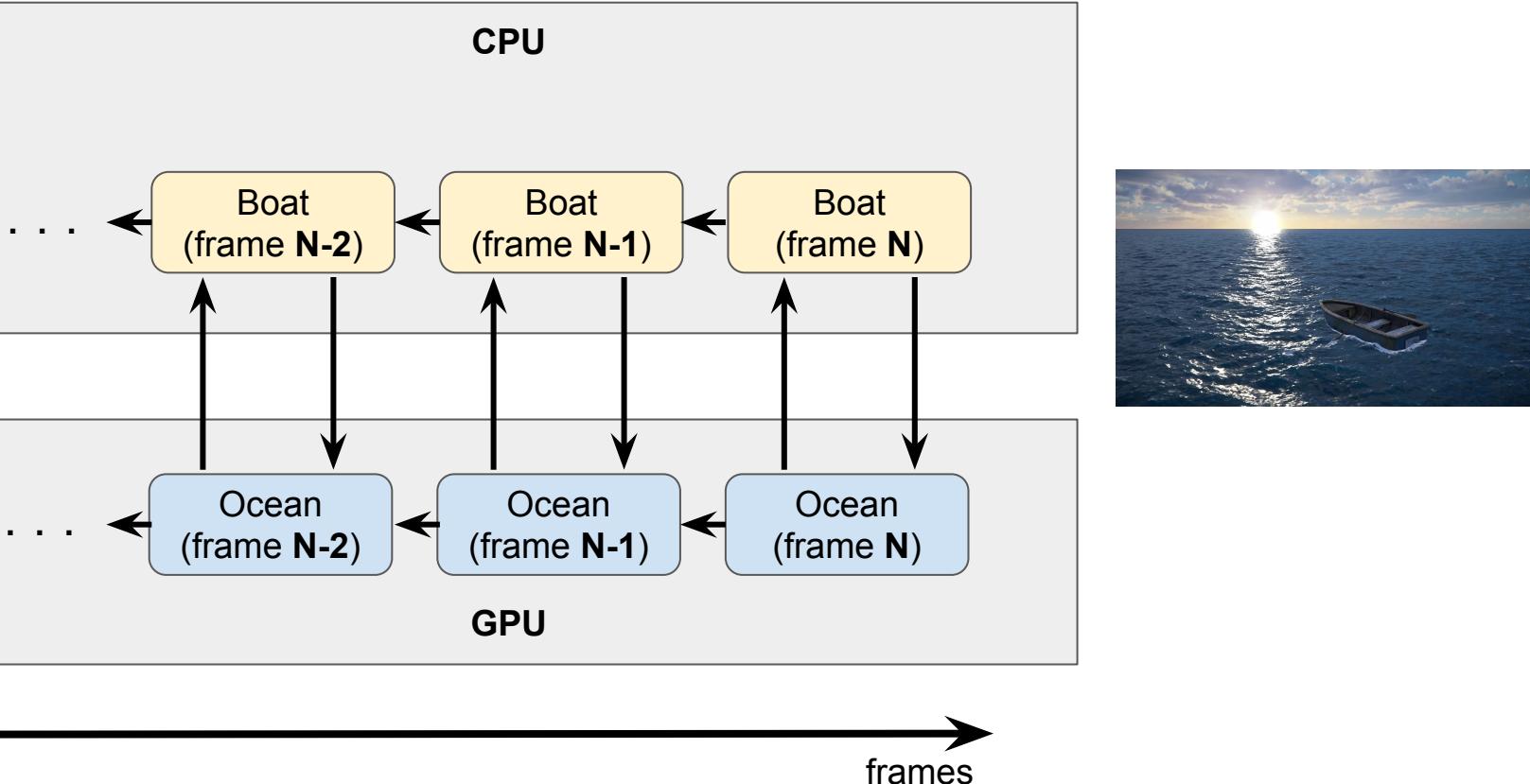
The Problem



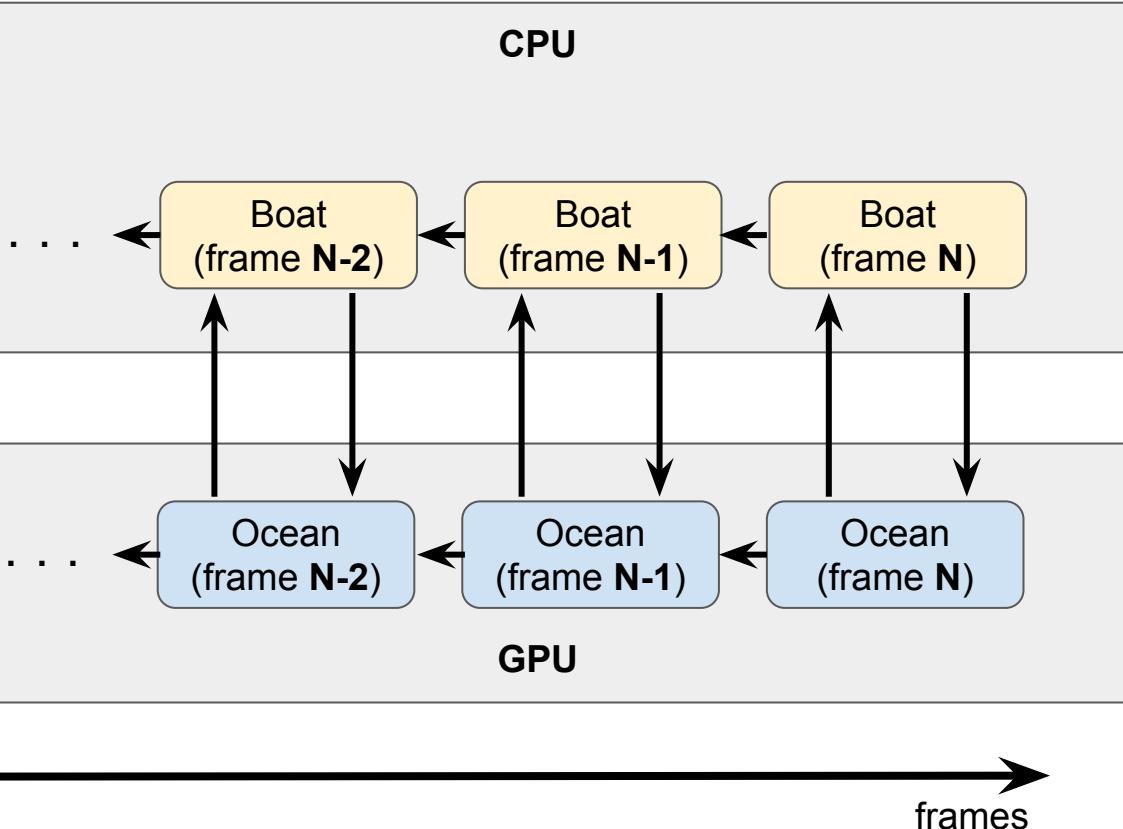
The Problem



The Problem: Simulation Across CPU-GPU



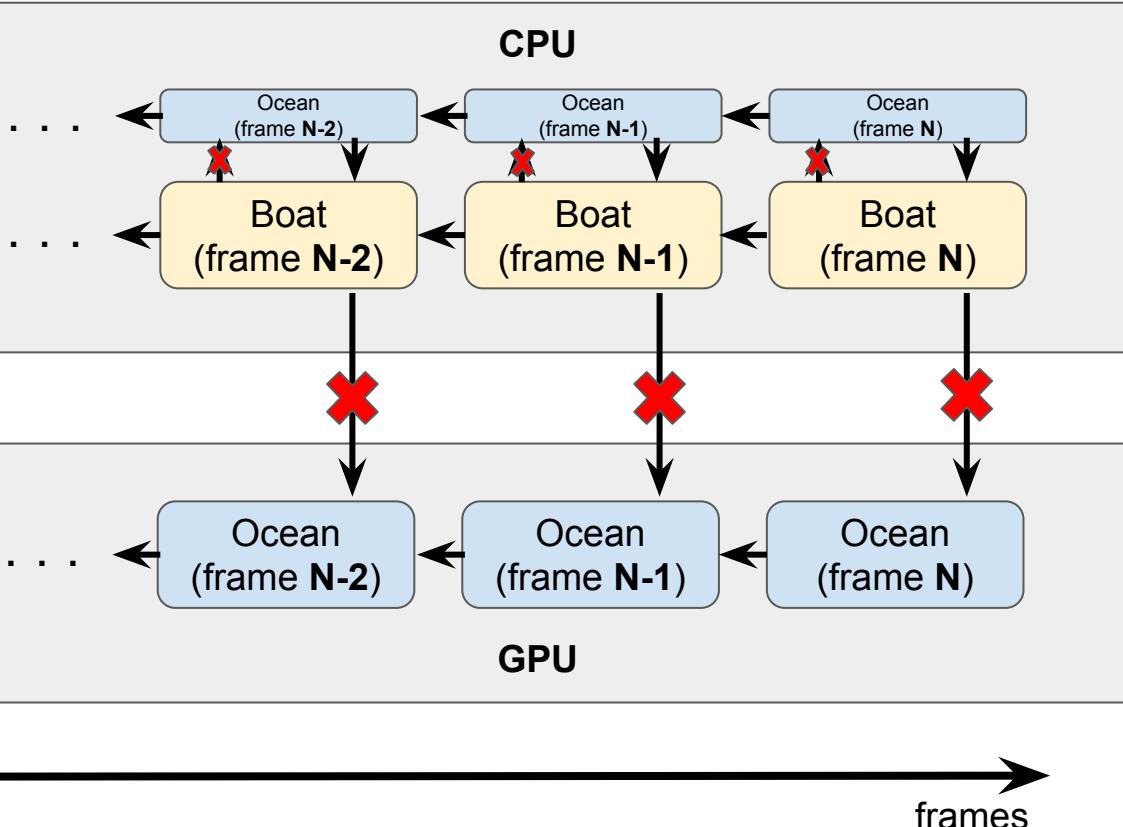
The Problem: Simulation Across CPU-GPU



Prior work:

- Replication
 - No 2-way interactions
 - Too slow or imprecise
- Synchronous readbacks
 - Too slow
- Asynchronous readbacks
 - Latency-induced errors

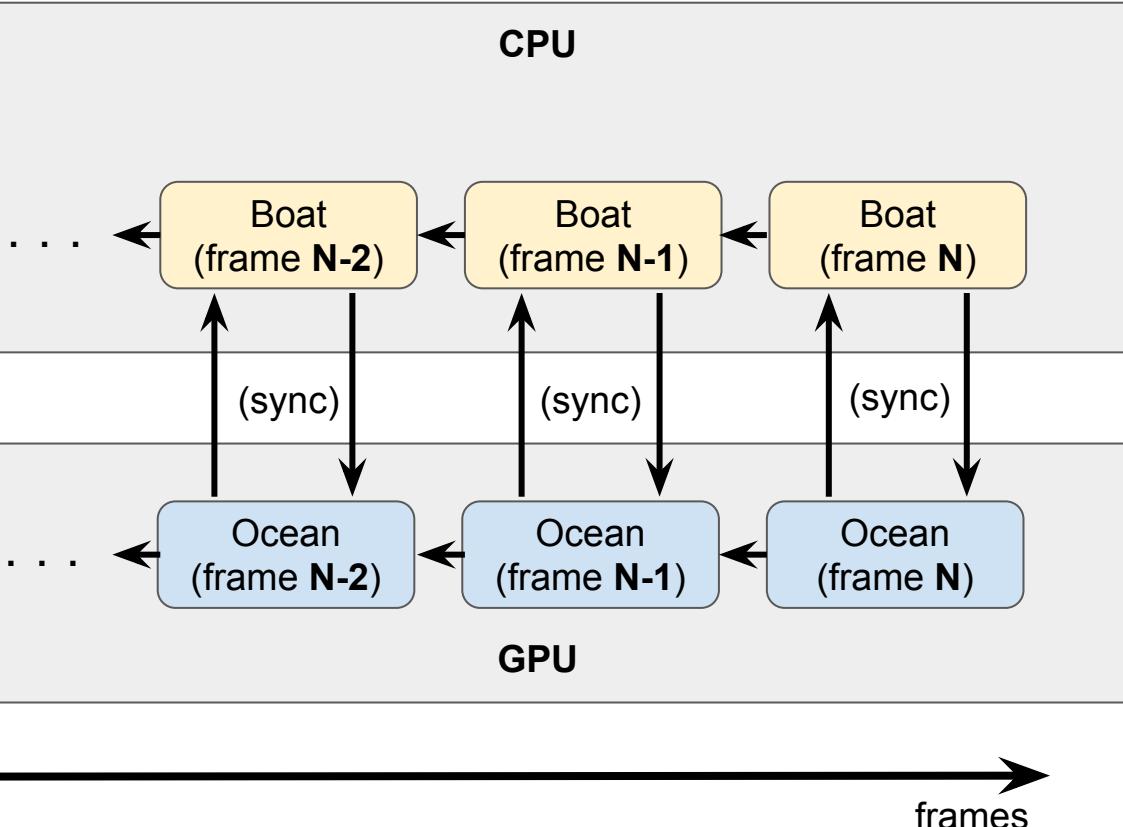
The Problem: Simulation Across CPU-GPU



Prior work:

- **Replication**
 - No 2-way interactions
 - Too slow or imprecise
- Synchronous readbacks
 - Too slow
- Asynchronous readbacks
 - Latency-induced errors

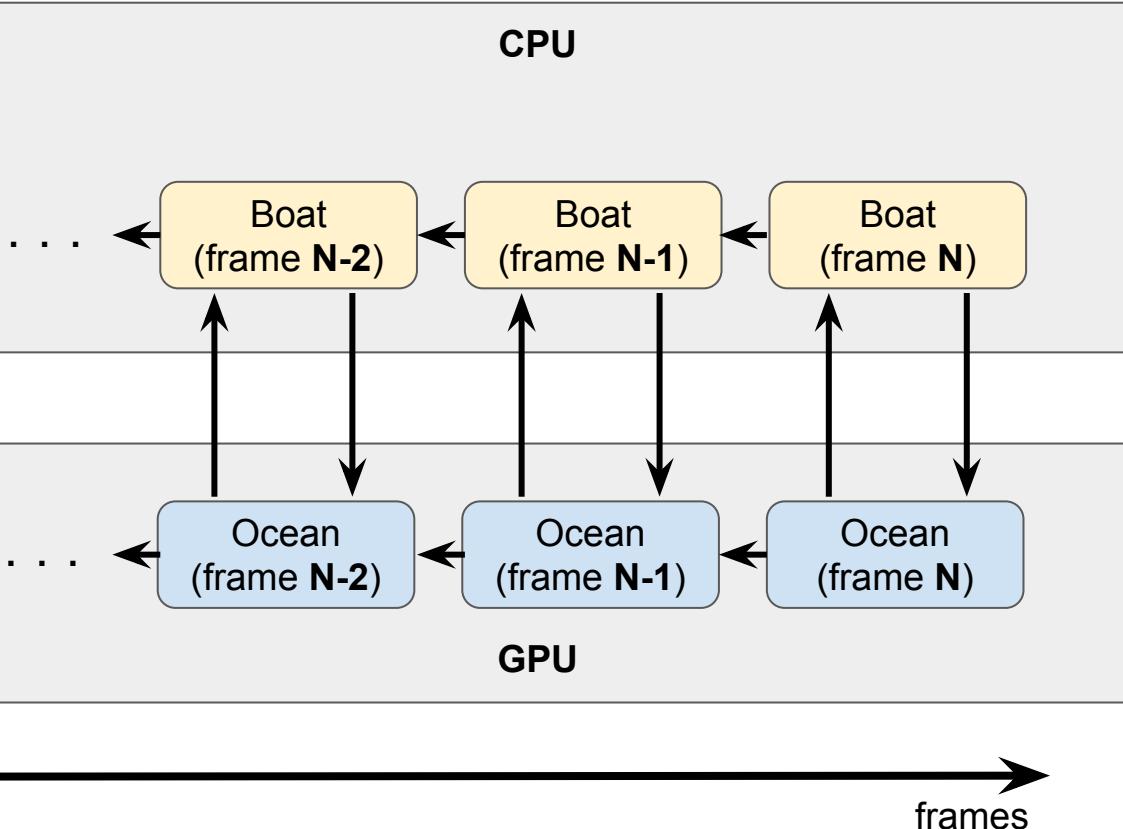
The Problem: Simulation Across CPU-GPU



Prior work:

- Replication
 - No 2-way interactions
 - Too slow or imprecise
- **Synchronous readbacks**
 - Too slow (pipeline stall)
- Asynchronous readbacks
 - Latency-induced errors

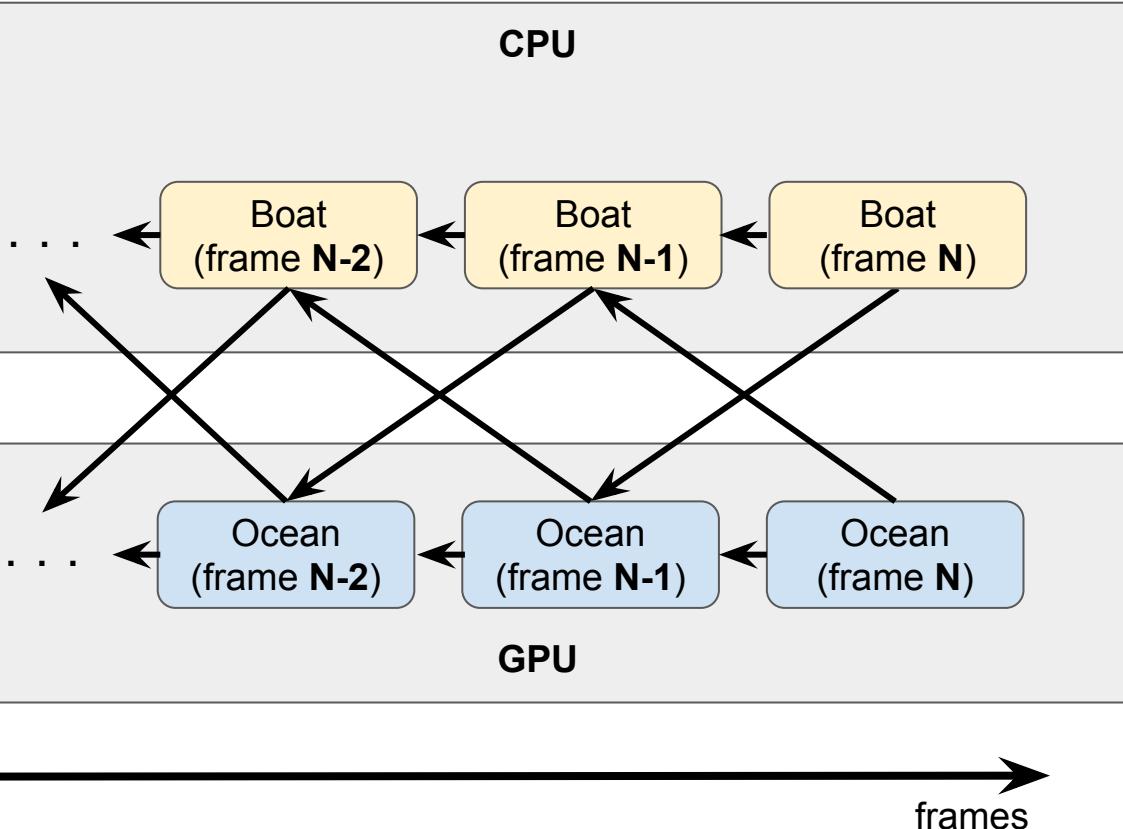
The Problem: Simulation Across CPU-GPU



Prior work:

- Replication
 - No 2-way interactions
 - Too slow or imprecise
- Synchronous readbacks
 - Too slow (pipeline stall)
- **Asynchronous readbacks**
 - Latency-induced errors

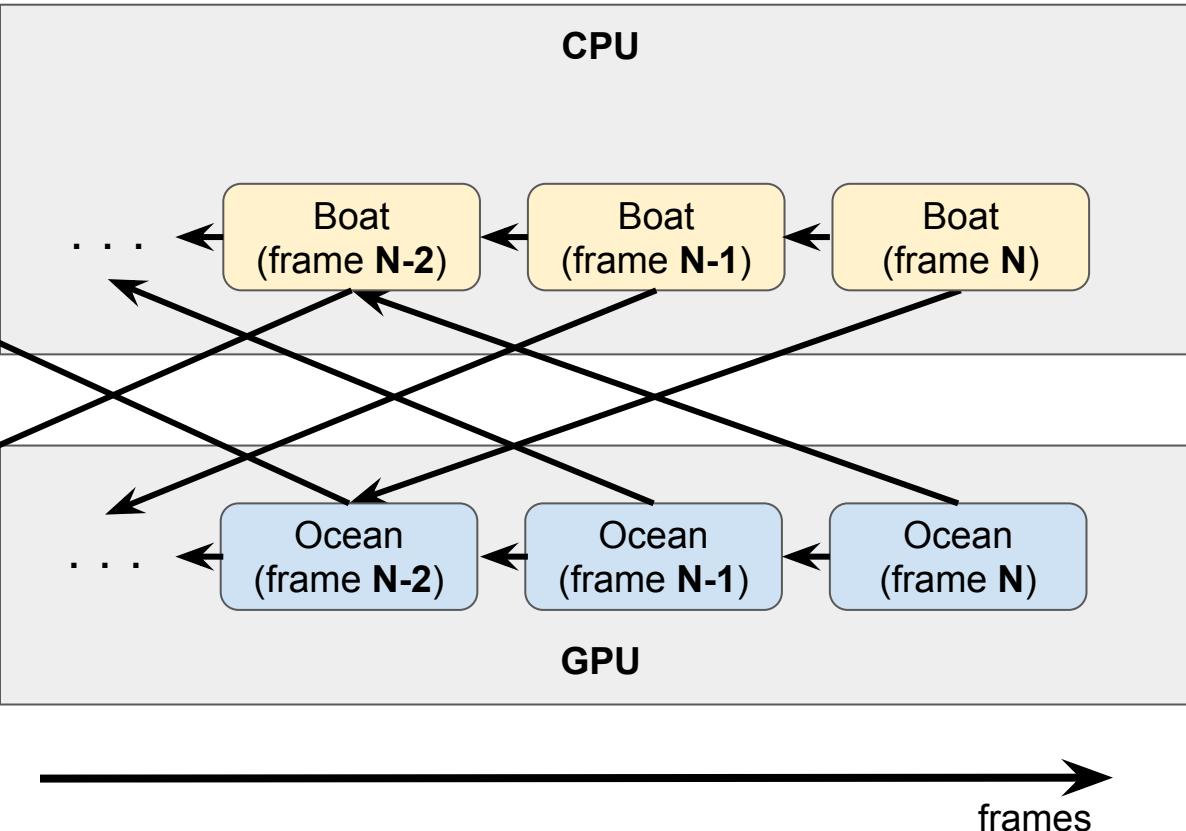
The Problem: Simulation Across CPU-GPU



Prior work:

- Replication
 - No 2-way interactions
 - Too slow or imprecise
- Synchronous readbacks
 - Too slow (pipeline stall)
- **Asynchronous readbacks**
 - Latency-induced errors

The Problem: Simulation Across CPU-GPU



Prior work:

- Replication
 - No 2-way interactions
 - Too slow or imprecise
- Synchronous readbacks
 - Too slow (pipeline stall)
- **Asynchronous readbacks**
 - Latency-induced errors

Improving the Accuracy of FFT-based GPGPU Ocean Surface Simulations

A master's thesis in computer science and engineering

Improving the Accuracy of FFT-based GPGPU Ocean Surface Simulations

A master's thesis in computer science and engineering

Research Questions:

- In practice, what is the significance of the latency-induced simulation errors?
- Could this simulation accuracy be recovered without hurting performance?

Improving the Accuracy of FFT-based GPGPU Ocean Surface Simulations

A master's thesis in computer science and engineering

Research Questions:

- In practice, what is the significance of the latency-induced simulation errors?
- Could this simulation accuracy be recovered without hurting performance?

Yes, the errors are significant. Yes, we can efficiently remove them.

Improving the Accuracy of FFT-based GPGPU Ocean Surface Simulations

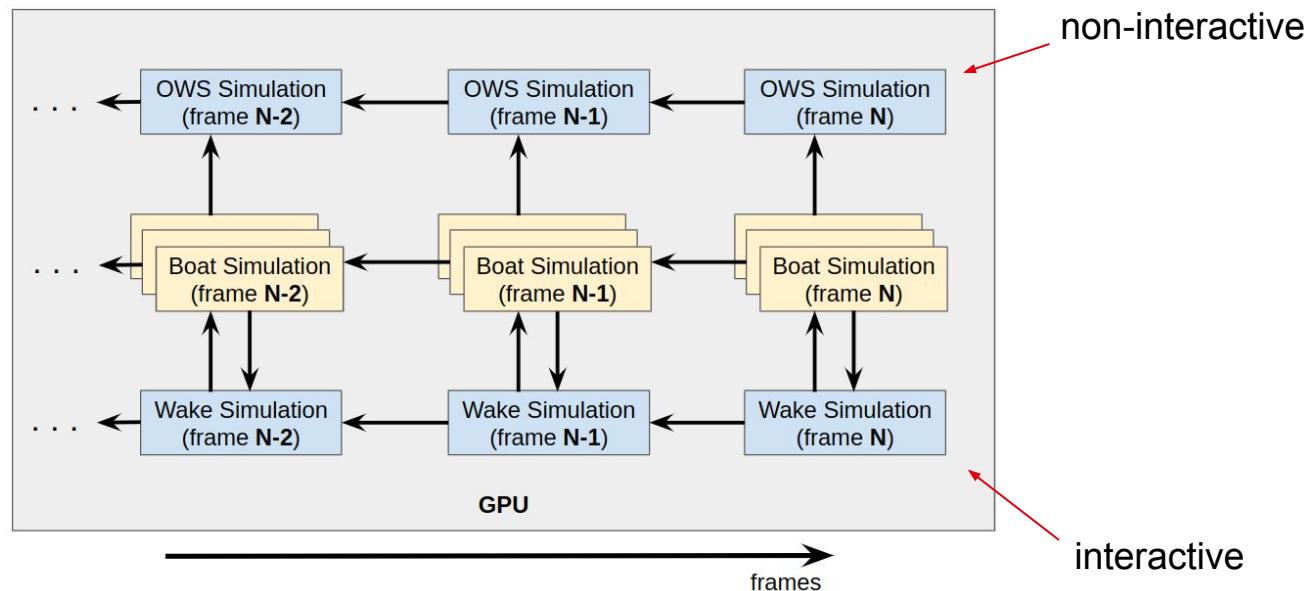
A master's thesis in computer science and engineering

Contributions:

- An investigation of the severity of the latency-induced errors
- A framework that removes these errors with virtually no performance loss

Our Framework — The Core Idea

- Bring it all to the GPU
- Circumventing the issue by avoiding all GPU readbacks
- Our framework:



Our Framework — Implementation

- Prototype implemented in Unreal Engine
 - Realistic restrictions and performance overhead
 - Does the heavy lifting for rendering
 - Let's us focus on the simulation
 - Simulation written from scratch (in C++ and shaders)

Non-interactive Ocean



Interactive Wakes

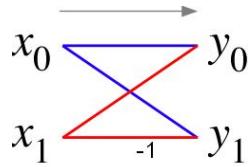


Boat Physics

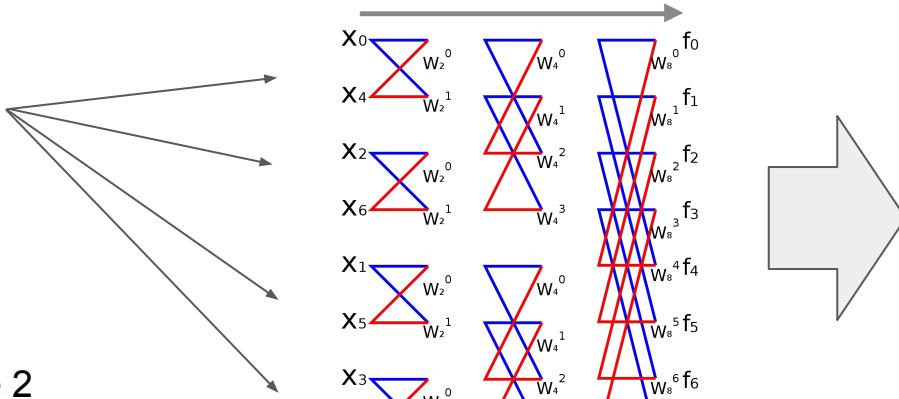


Fast Fourier Transform, Butterfly

- Discrete Fourier Transform is $O(N^2)$
- FFT is $O(N \log N)$, key for real-time
- Large DFT recursively broken down



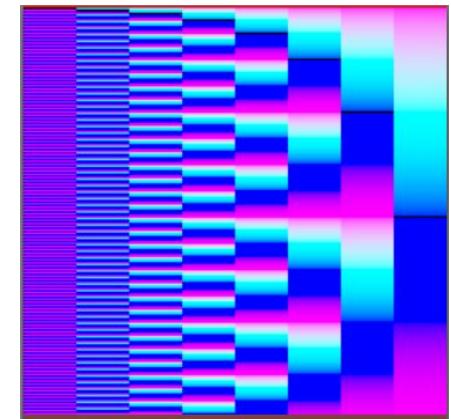
"Butterfly" DFT of size 2



Log N butterfly layers

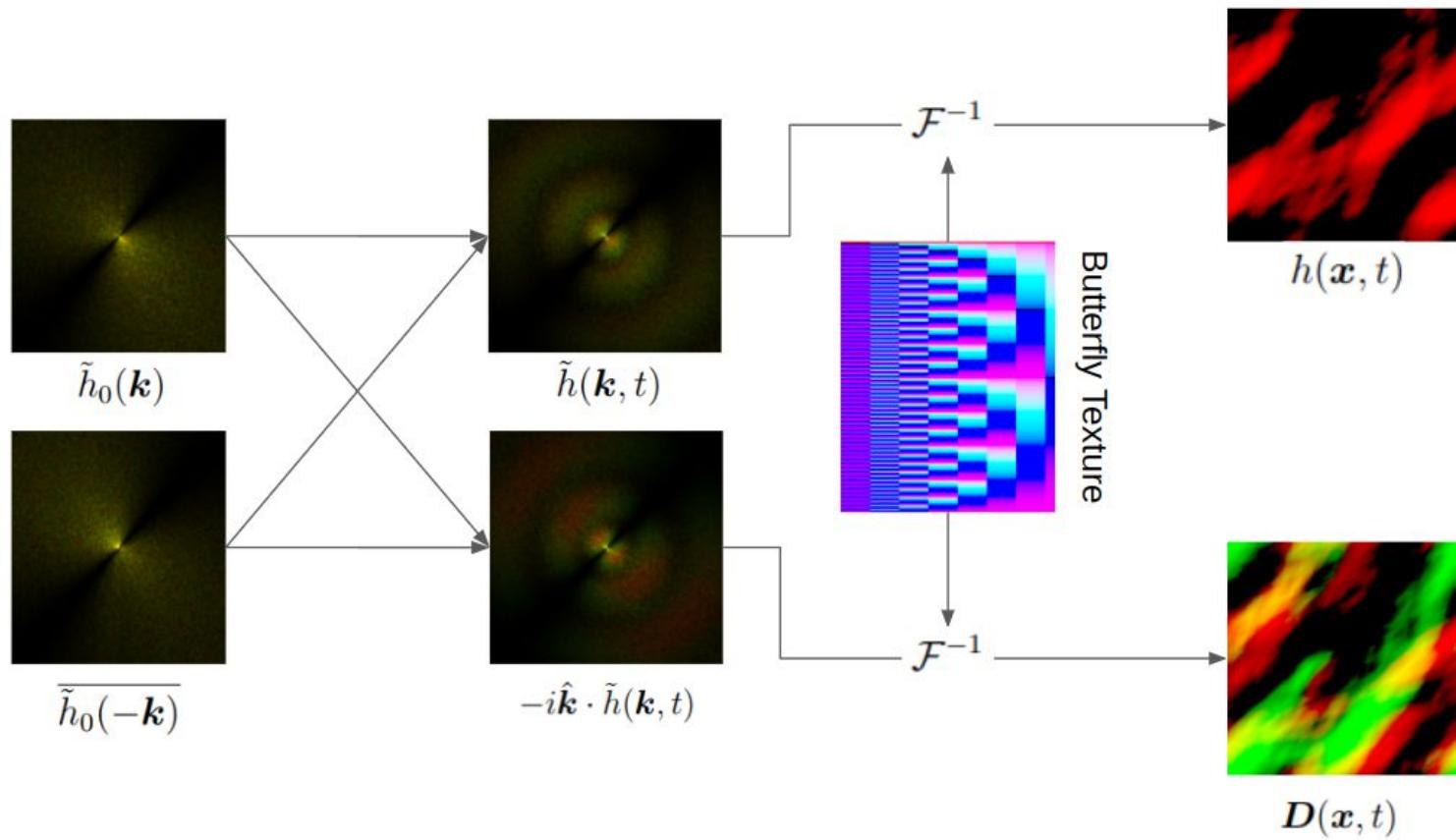
$$\tilde{h}_P(-\mathbf{k}, t) = \overline{\tilde{h}_P(\mathbf{k}, t)}$$

Hermitian property

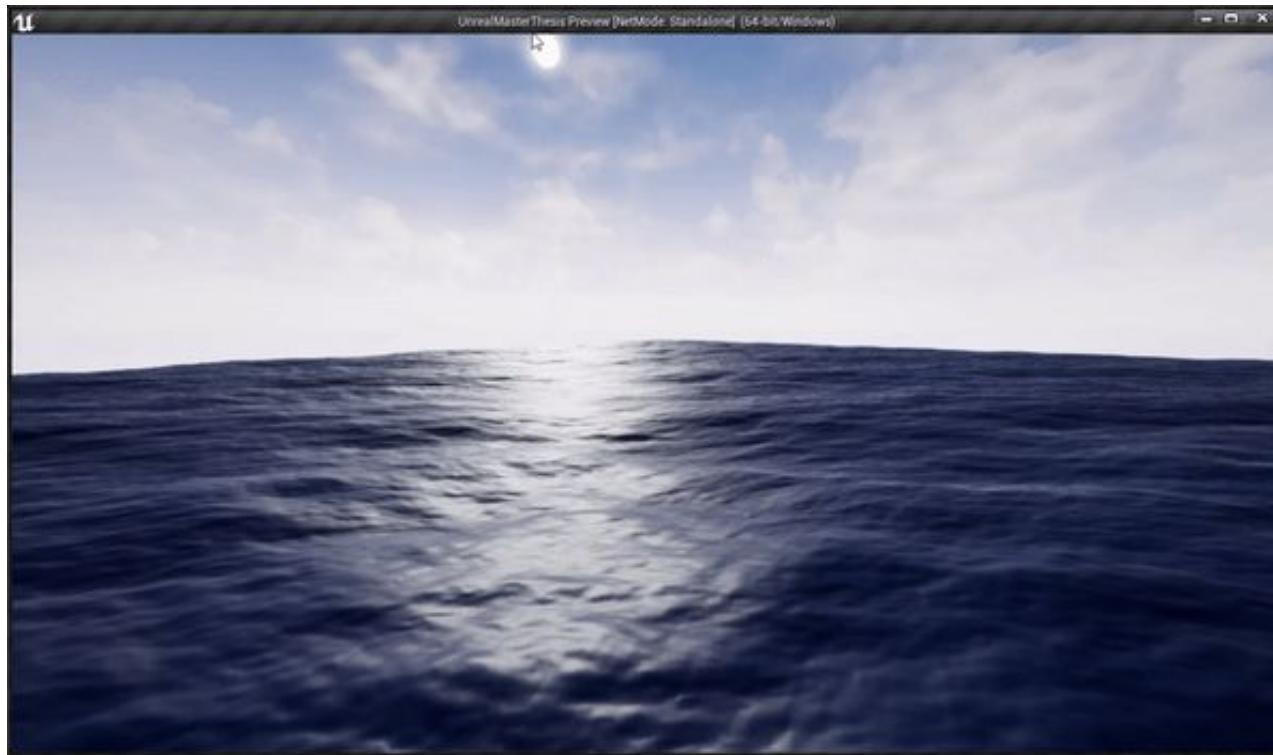


Butterfly Texture

Non-interactive Ocean — Ocean Wave Spectrum

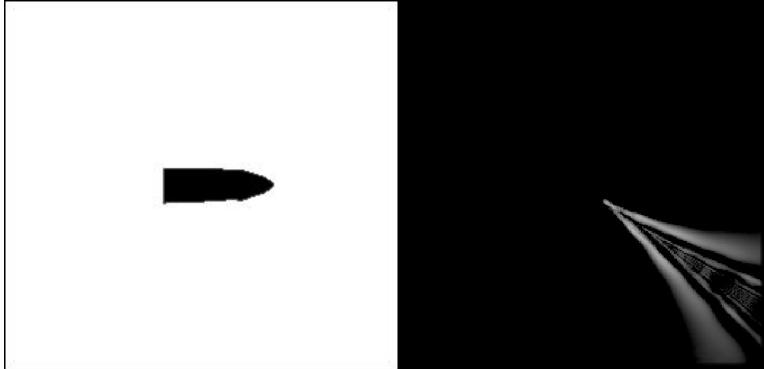


Non-interactive Ocean — Ocean Wave Spectrum



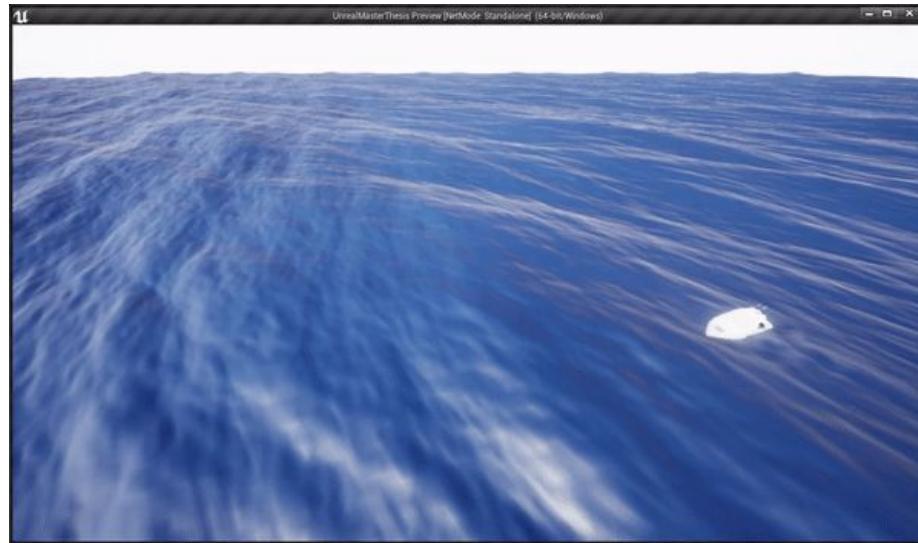
Interactive Ocean — eWave

- Linearized Bernoulli's Equation
- Simplistic Displacement
- Accurate Wave Propagation



2D masking

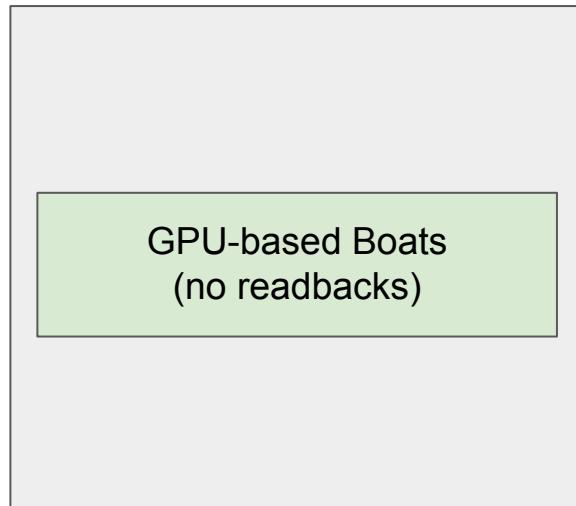
Wake texture



Boat Physics

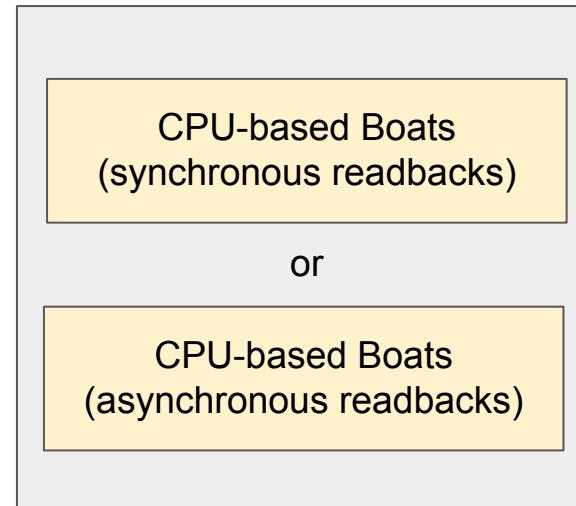
- Based on article by Jacques Kerner
- Used in the AAA game Just Cause 3
- Forces:
 - Gravity
 - Buoyancy
 - Slamming Force
 - Artificial Dampening
 - User Control

Evaluating the Framework



Our Framework

vs.



State of the Art

Research Questions:

- In practice, what is the significance of the latency-induced simulation errors?
- Could this simulation accuracy be recovered without hurting performance?

Research Questions:

- **RQ1:** In practice, what is the significance of the latency-induced simulation errors?
- **RQ2:** Could this simulation accuracy be recovered without hurting performance?

Research Questions:

- **RQ1:** In practice, what is the significance of the latency-induced simulation errors?
- RQ2: Could this simulation accuracy be recovered without hurting performance?

Research Questions:

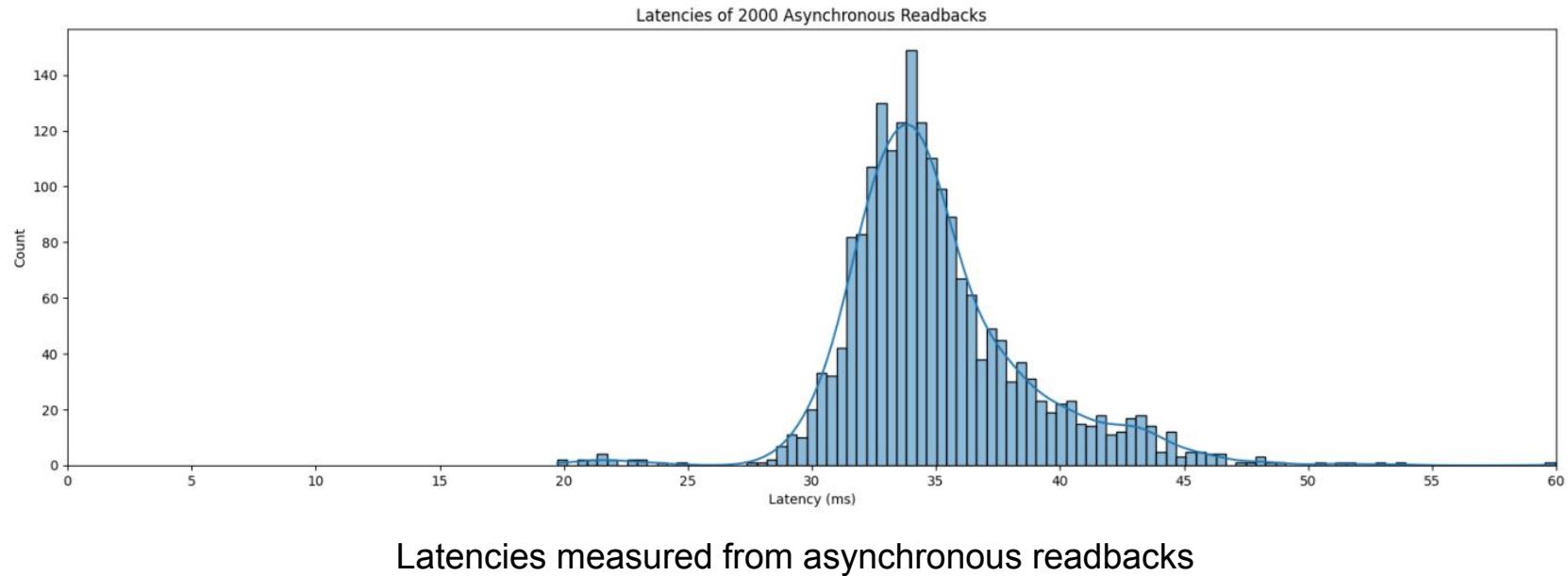
- **RQ1:** In practice, what is the significance of the latency-induced simulation errors?
- RQ2: Could this simulation accuracy be recovered without hurting performance?

RQ1 — Significance of Latency?

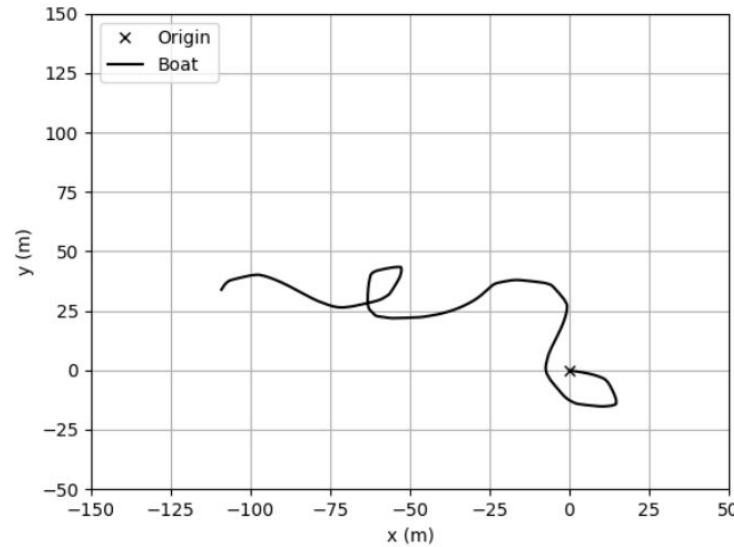
- 0 latency = baseline
 - synchronous CPU-based boats
 - GPU-based boats
- Asynchronous readbacks:
 - Fixed latencies (1, 2, and 3 frames)
 - “Organic” latency

RQ1 — Significance of Latency?

- Organic delay: ~1-2 frames of latency (~16.7 ms frame budget)

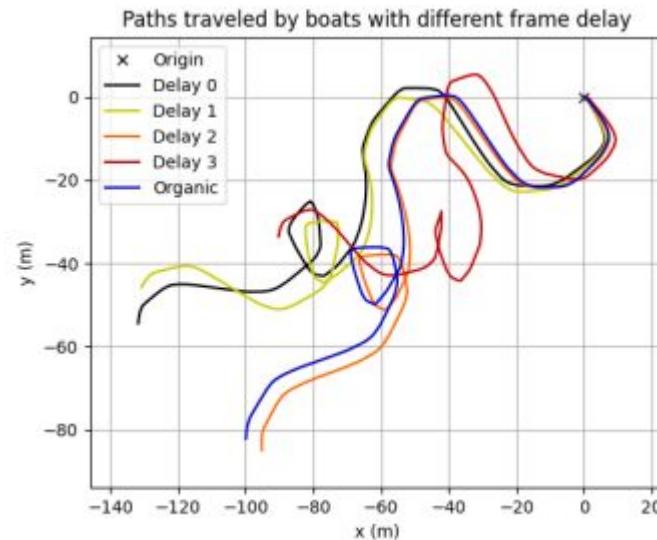


RQ1 — Significance of Latency?



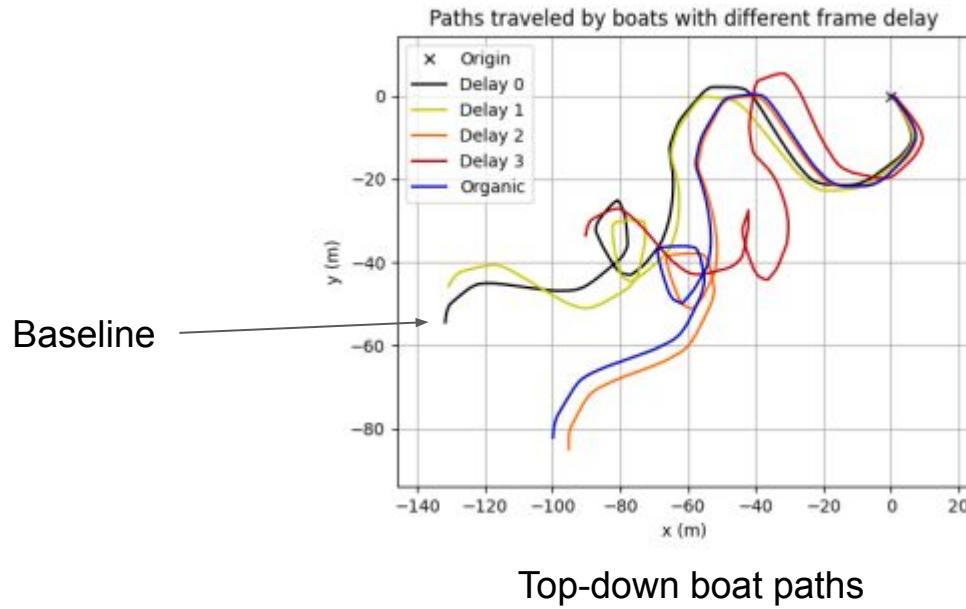
Example top-down boat path

RQ1 — Significance of Latency?

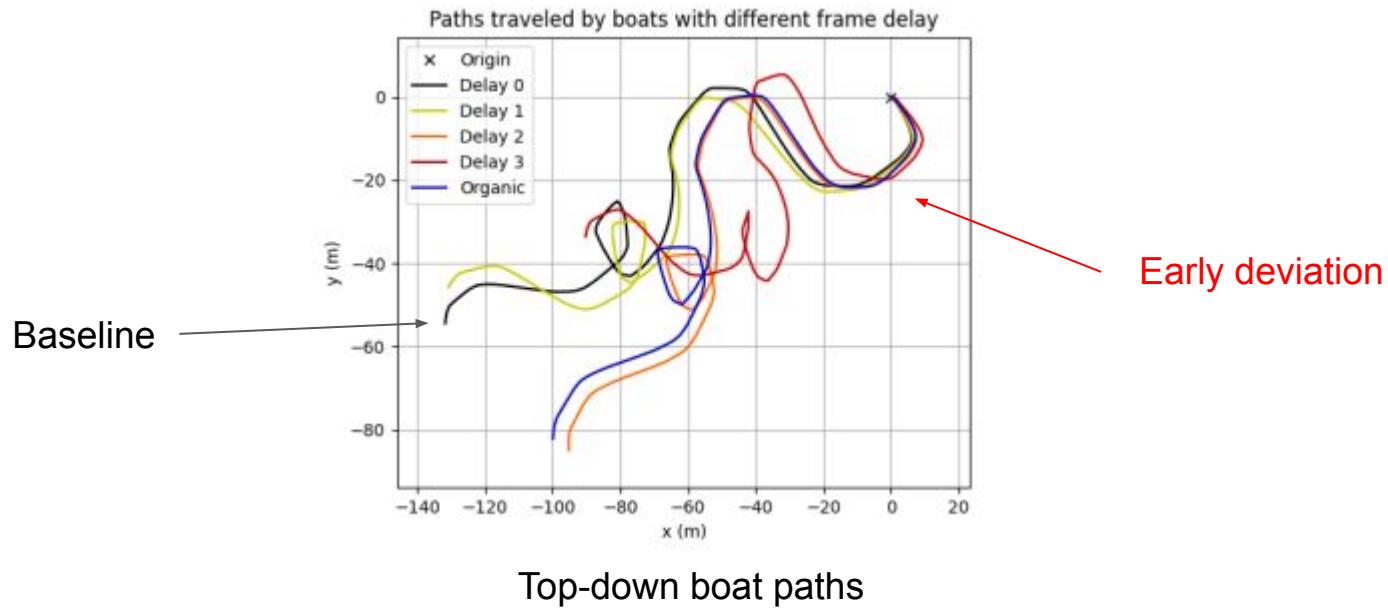


Top-down boat paths

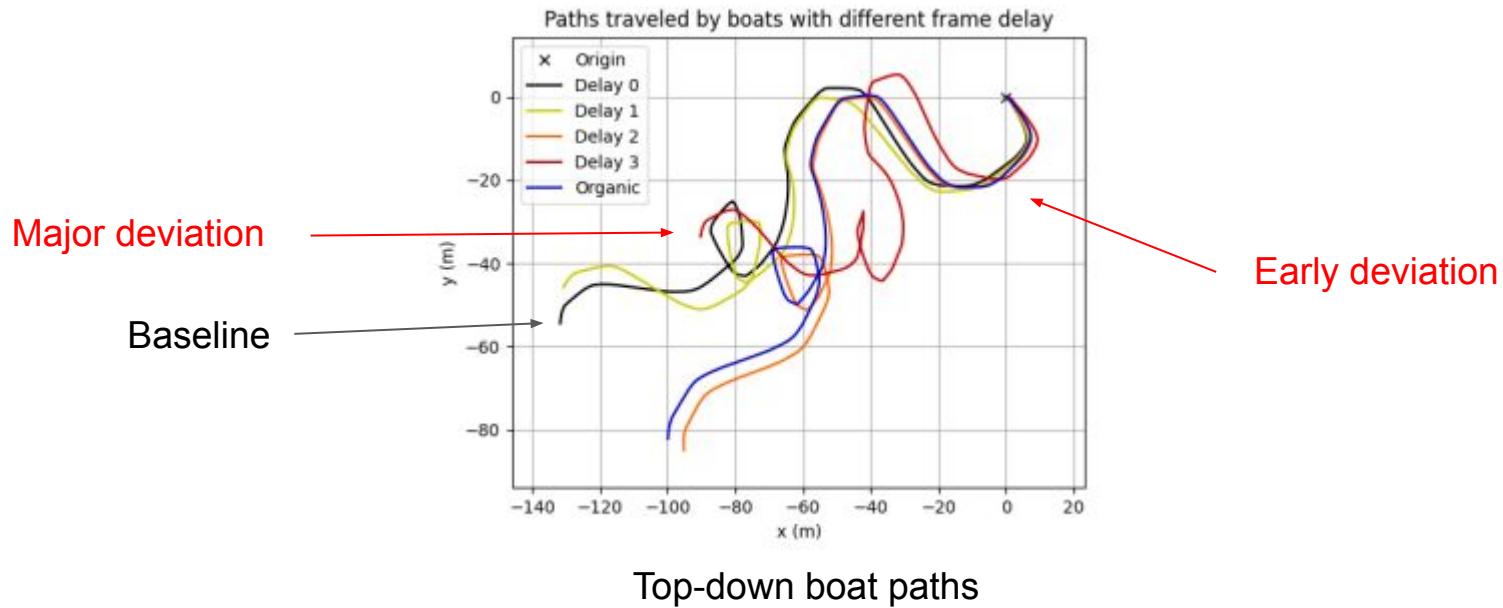
RQ1 — Significance of Latency?



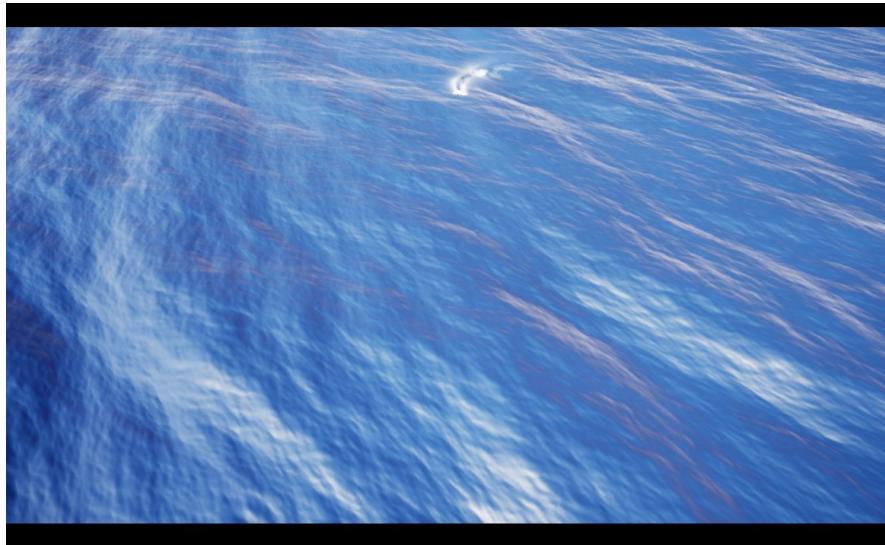
RQ1 — Significance of Latency?



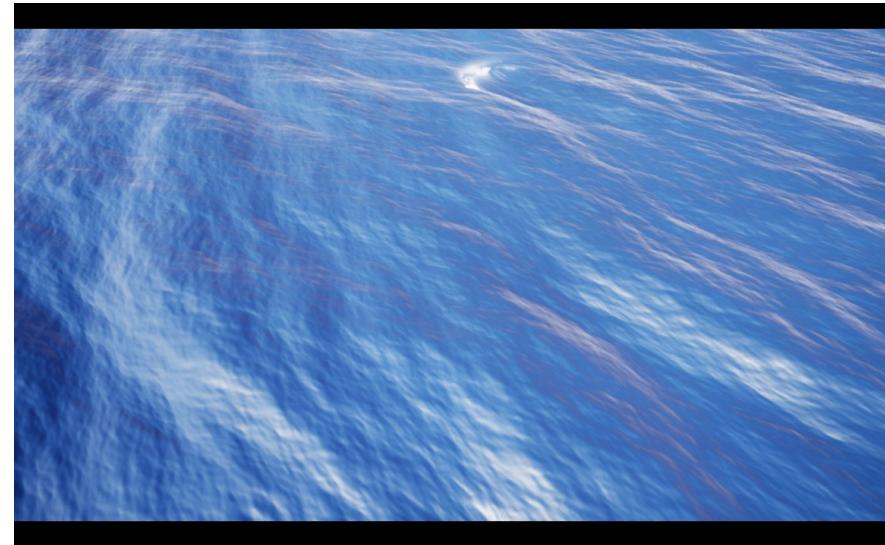
RQ1 — Significance of Latency?



RQ1 — Significance of Latency?



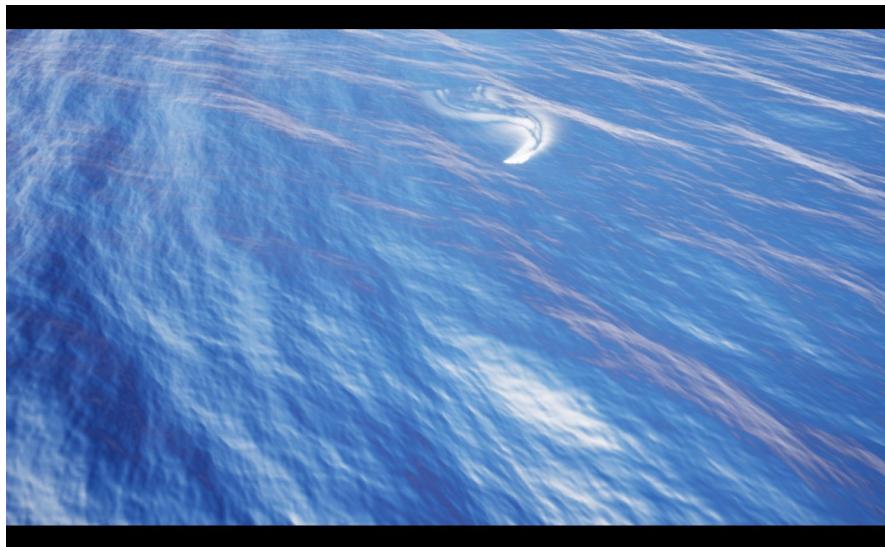
No latency



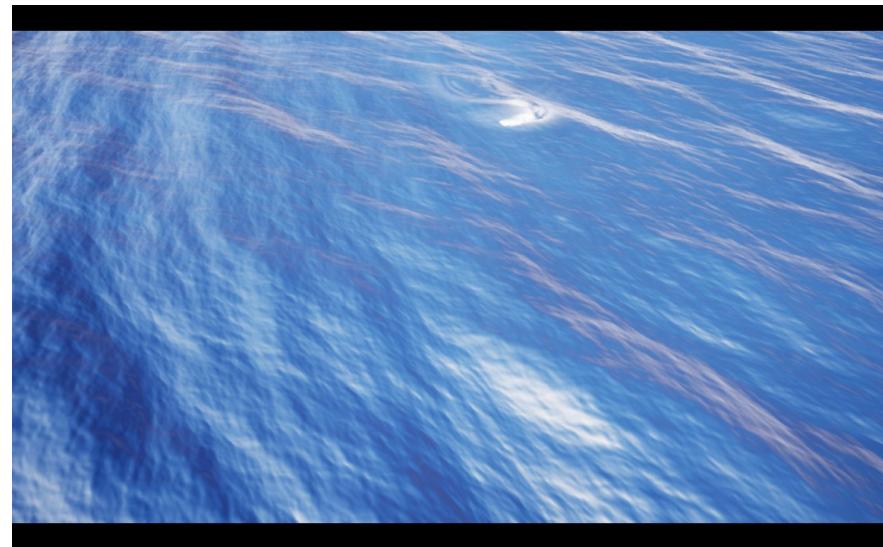
3 frames of latency

Frame 200

RQ1 — Significance of Latency?



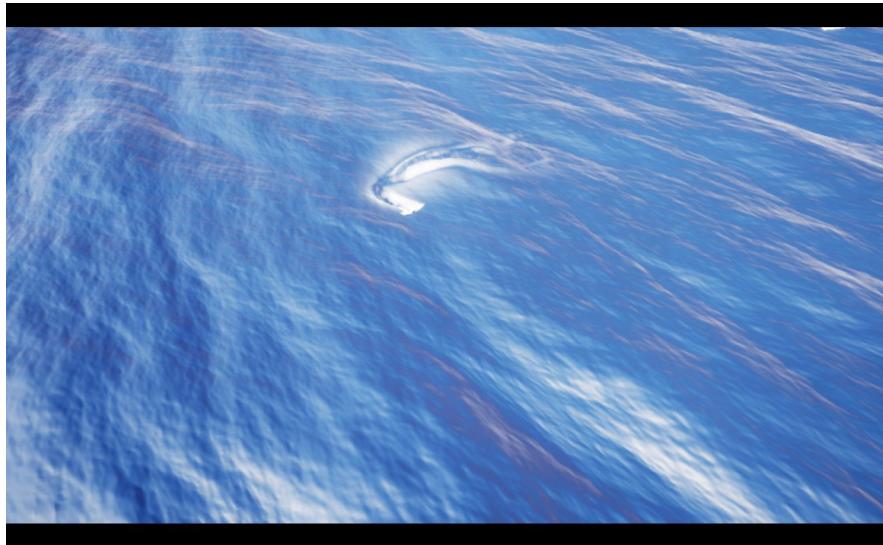
No latency



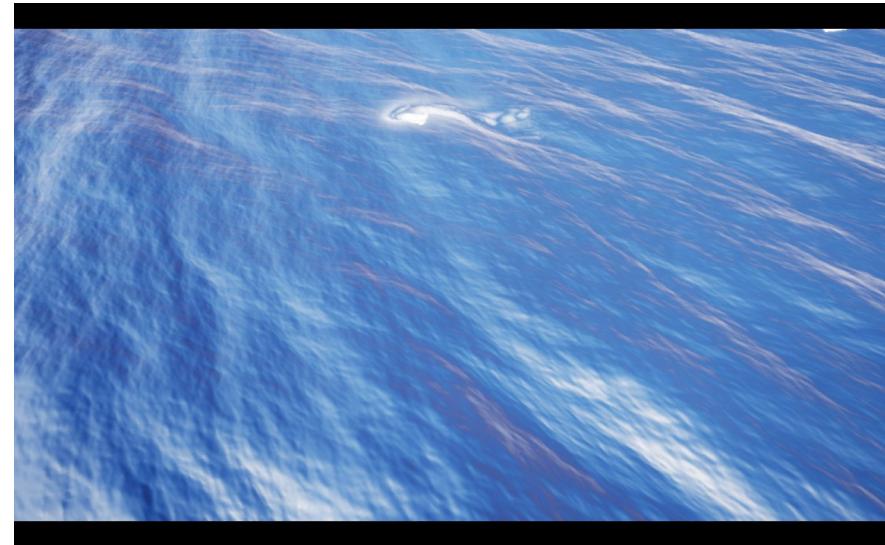
3 frames of latency

Frame 400

RQ1 — Significance of Latency?



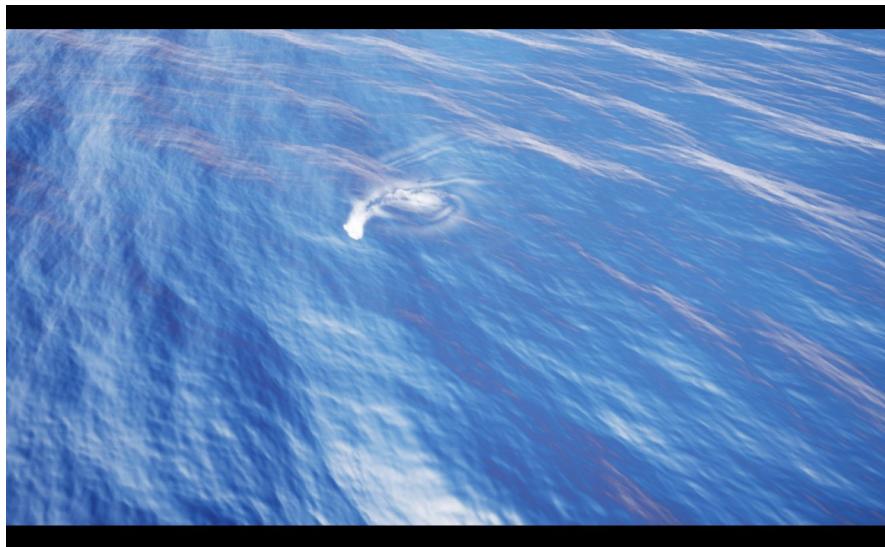
No latency



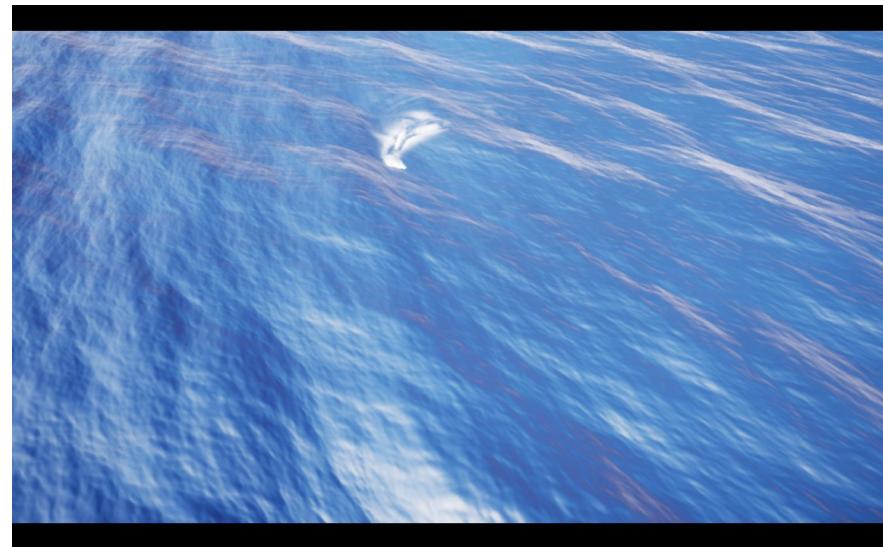
3 frames of latency

Frame 600

RQ1 — Significance of Latency?



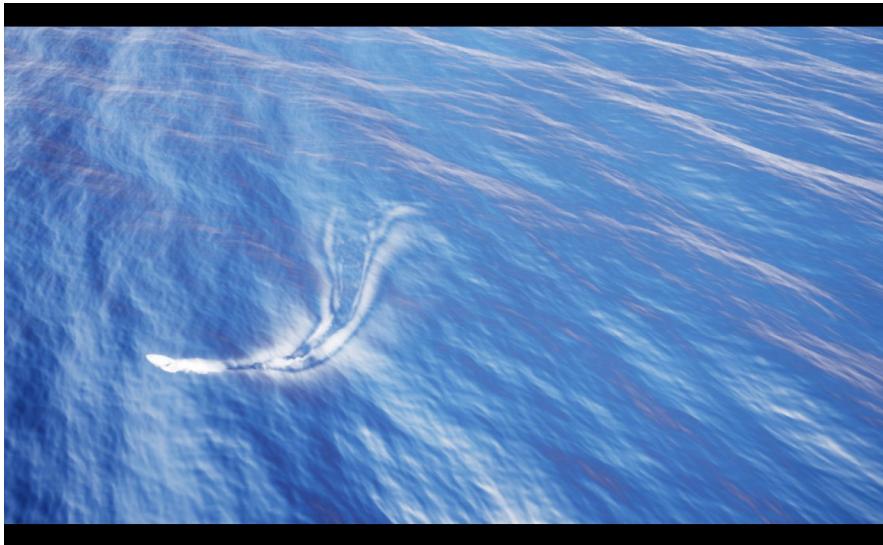
No latency



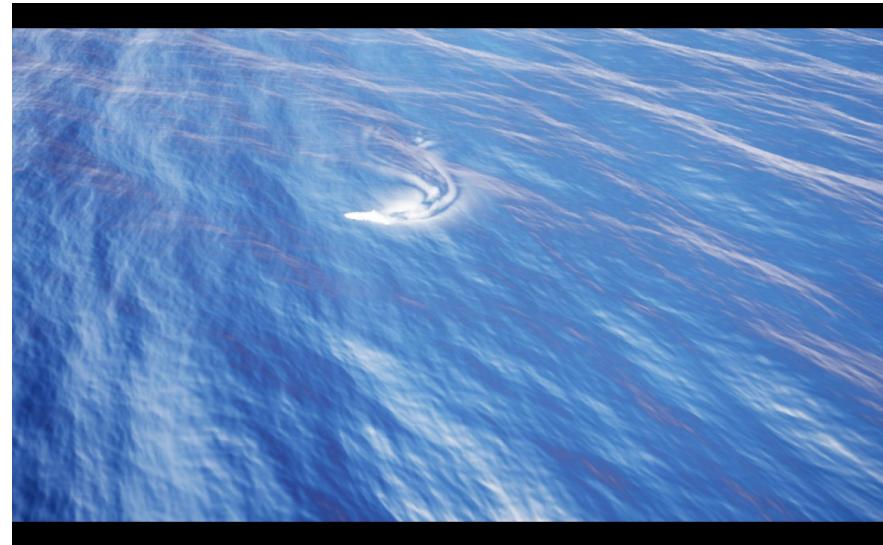
3 frames of latency

Frame 800

RQ1 — Significance of Latency?



No latency



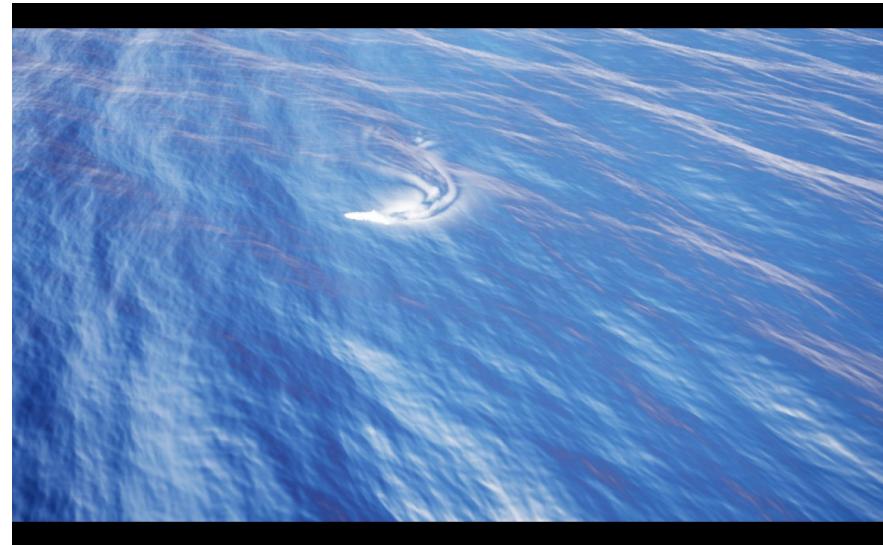
3 frames of latency

Frame 1000

RQ1 — Significance of Latency?



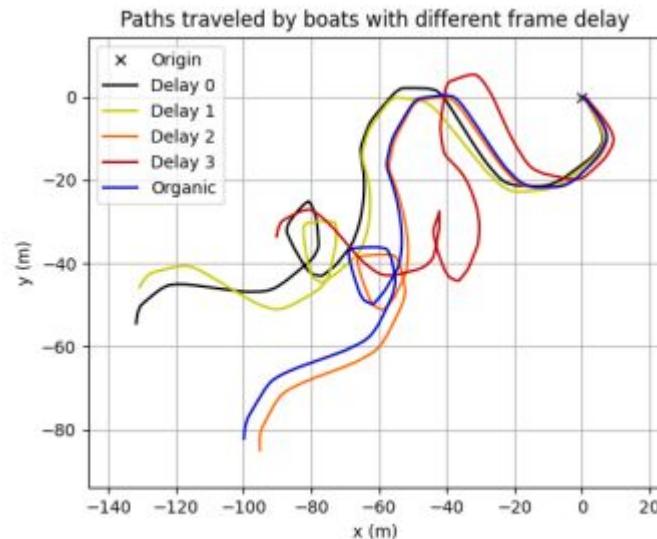
No latency



3 frames of latency

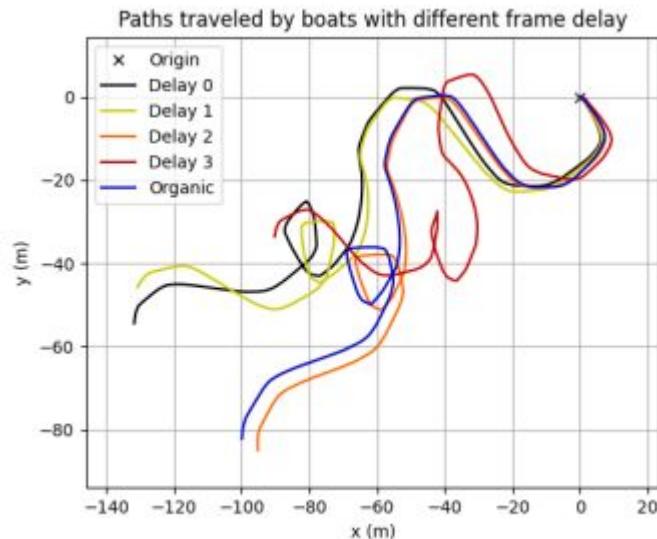
Frame 1000 = 20 seconds

RQ1 — Significance of Latency?

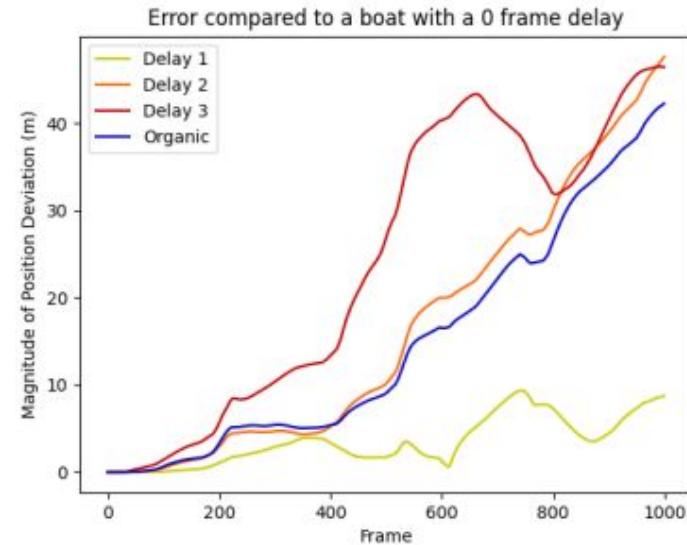


Top-down boat paths

RQ1 — Significance of Latency?

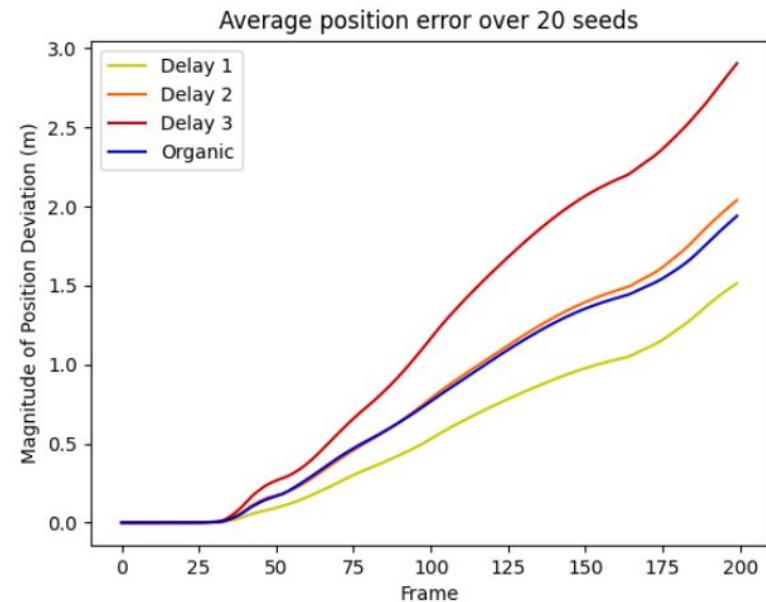
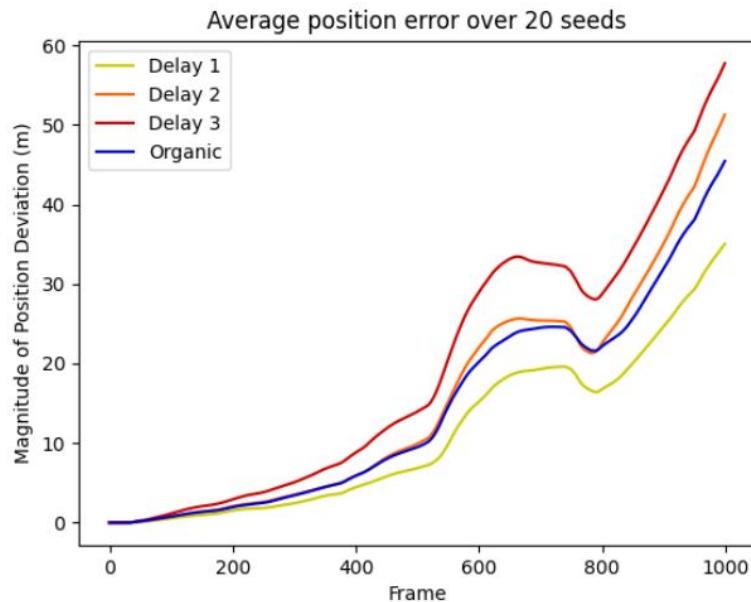


Top-down boat paths



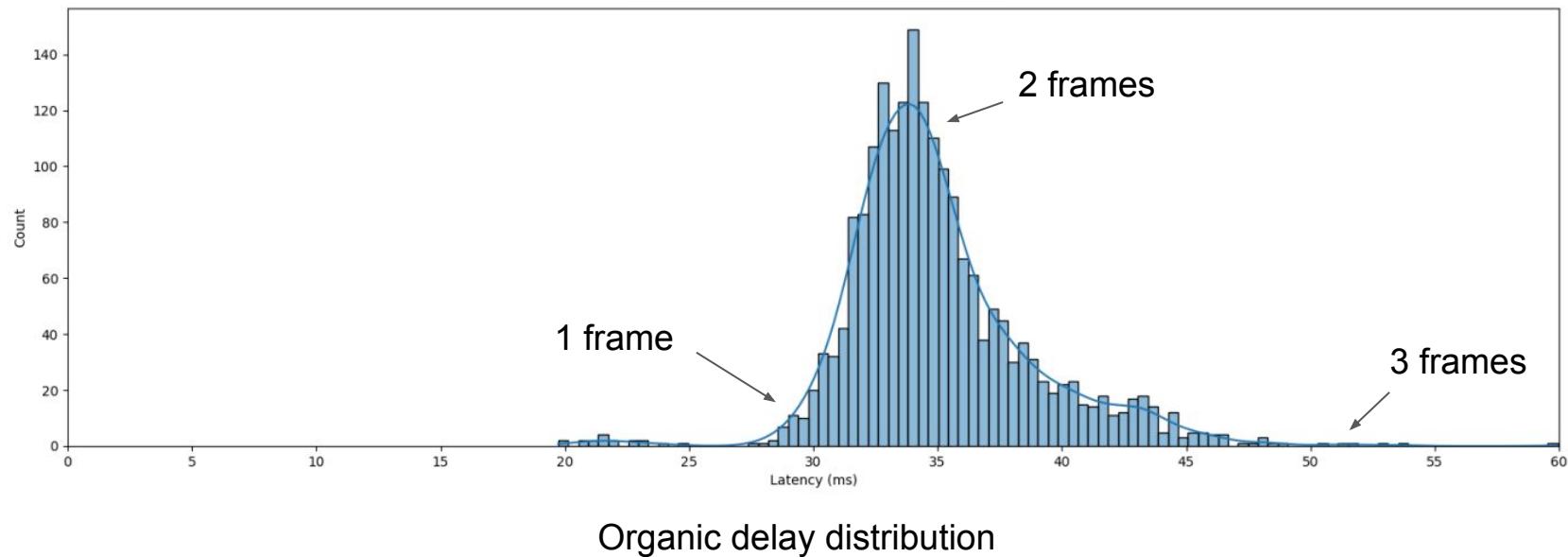
The corresponding errors

RQ1 — Significance of Latency?



RQ1 — Significance of Latency?

- Organic delay is non-deterministic
- This is undesirable in competitive contexts



RQ1 — Significance of Latency?



The Crew 2
(2018)

RQ1 — Significance of Latency?



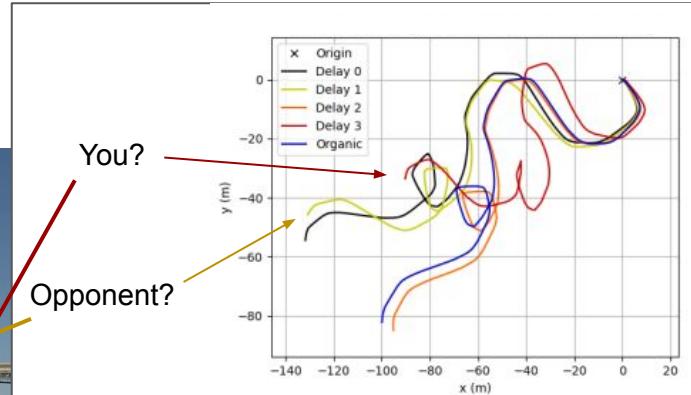
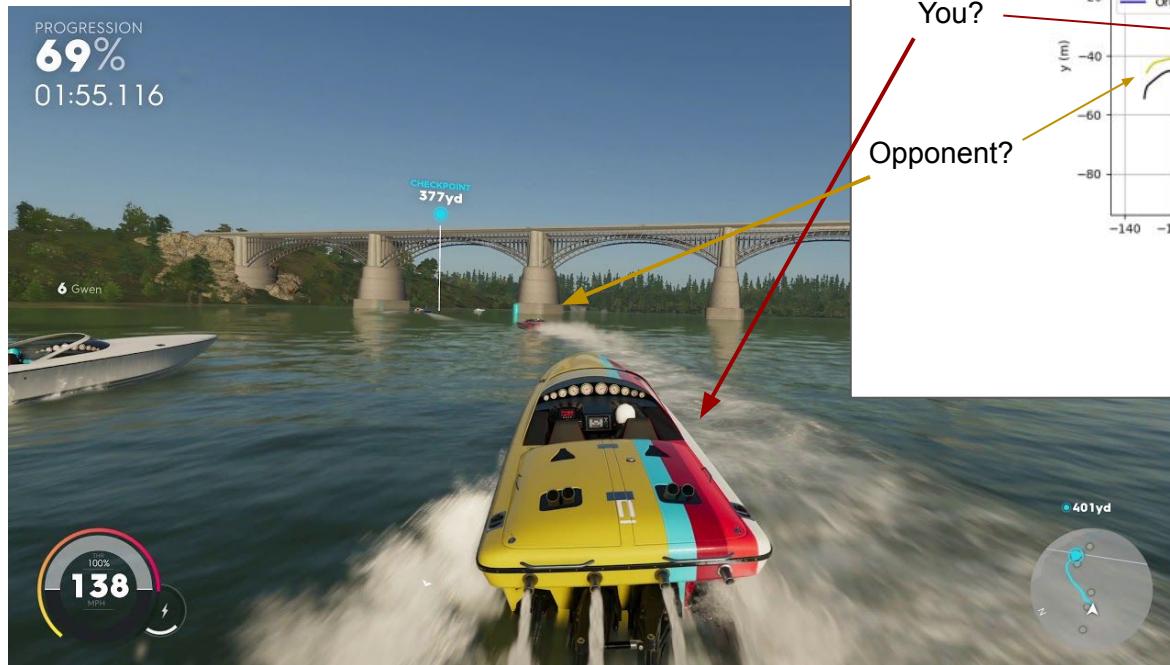
The Crew 2
(2018)

RQ1 — Significance of Latency?



The Crew 2
(2018)

RQ1 — Significance of Latency?



The Crew 2
(2018)

RQ1 — Significance of Latency?

- Visual Artifacts

60 Frames Latency



RQ1 — Significance of Latency?

3 Frames Latency



Frame 1

RQ1 — Significance of Latency?

3 Frames Latency



Frame 2

RQ1 — Significance of Latency?

3 Frames Latency



Frame 3

RQ1 — Significance of Latency?

3 Frames Latency



Frame 4

RQ1 — Significance of Latency?

3 Frames Latency



Frame 5

Research Questions:

- RQ1: In practice, what is the significance of the latency-induced simulation errors?
- **RQ2**: Could this simulation accuracy be recovered without hurting performance?

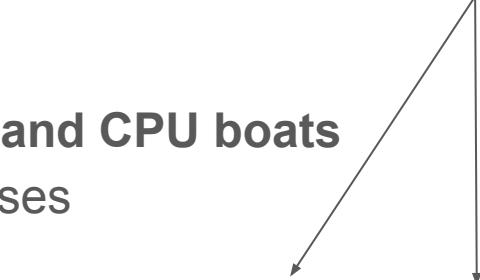
RQ2 — Performance

- Unable to use Unreal Insights for rendering pipeline
- RenderDoc used for individual shaders
- Frame cost in different scenarios estimated in Unreal
 - CPU (sync) vs CPU (async) vs GPU
 - One vs two boats

A: Intel i7-2700K 3.50GHz, NVIDIA GTX 1070 Ti
B: AMD Ryzen 7 3.20GHz, NVIDIA GeForce RTX 3060 (Laptop)

RQ2 — Performance

- **Very similar performance for GPU and CPU boats**
- Bottleneck is the cost of the FFT passes



Setting	Setup Ȣ		Setup Ȣ	
	GPU	CPU	GPU	CPU
Empty scene	2.215	0.000	3.087	0.000
Only the (non-interactive) ocean	2.843	0.015	3.250	0.010
Ocean + 1 CPU-based boat (sync)	5.879	3.051	5.195	2.237
Ocean + 1 CPU-based boat (async)	3.111	0.406	3.940	0.326
Ocean + 1 GPU-based boat	2.924	0.354	3.801	0.252
Ocean + 2 CPU-based boats (sync)	8.389	5.587	6.252	3.773
Ocean + 2 CPU-based boats (async)	3.256	0.663	4.771	0.452
Ocean + 2 GPU-based boats	3.052	0.516	5.120	0.387

(costs in milliseconds)

RQ2 — Performance

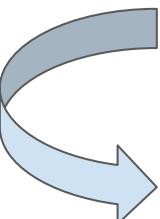
- **Very similar performance for GPU and CPU boats**
- Bottleneck is the cost of the FFT passes

Setting	Setup Ȣ		Setup Ȣ	
	GPU	CPU	GPU	CPU
Empty scene	2.215	0.000	3.087	0.000
Only the (non-interactive) ocean	2.843	0.015	3.250	0.010
Ocean + 1 CPU-based boat (sync)	5.879	3.051	5.195	2.237
Ocean + 1 CPU-based boat (async)	3.111	0.406	3.940	0.326
Ocean + 1 GPU-based boat	2.924	0.354	3.801	0.252
Ocean + 2 CPU-based boats (sync)	8.389	5.587	6.252	3.773
Ocean + 2 CPU-based boats (async)	3.256	0.663	4.771	0.452
Ocean + 2 GPU-based boats	3.052	0.516	5.120	0.387

(costs in milliseconds)

RQ2 — Performance

- **Very similar performance for GPU and CPU boats**
- Bottleneck is the cost of the FFT passes



Setting	Setup Ȣ		Setup Ȣ	
	GPU	CPU	GPU	CPU
Empty scene	2.215	0.000	3.087	0.000
Only the (non-interactive) ocean	2.843	0.015	3.250	0.010
Ocean + 1 CPU-based boat (sync)	5.879	3.051	5.195	2.237
Ocean + 1 CPU-based boat (async)	3.111	0.406	3.940	0.326
Ocean + 1 GPU-based boat	2.924	0.354	3.801	0.252
Ocean + 2 CPU-based boats (sync)	8.389	5.587	6.252	3.773
Ocean + 2 CPU-based boats (async)	3.256	0.663	4.771	0.452
Ocean + 2 GPU-based boats	3.052	0.516	5.120	0.387

(costs in milliseconds)

RQ2 — Performance

- Very similar performance for GPU and CPU boats
- **Bottleneck is the cost of the FFT passes**

Feature	Setup Ȣ	Setup Ȣ
Update non-interactive ocean	195.584	285.008
Update GPU boat	44.032	60.416
Update wakes	324.608	535.552
FFT (256×256)	84.992	145.408

(costs in microseconds)

RQ2 — Performance

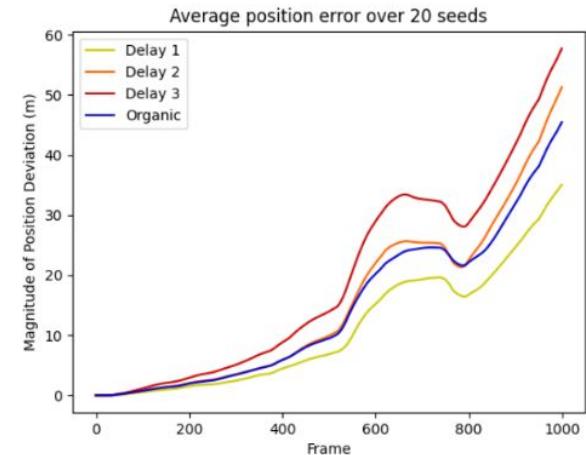
- Very similar performance for GPU and CPU boats
- **Bottleneck is the cost of the FFT passes**

Feature	Setup Ȣ	Setup Ȣ
Update non-interactive ocean	195.584	285.008
Update GPU boat	44.032	60.416
Update wakes	324.608	535.552
FFT (256×256)	84.992	145.408
cuFFT (256×256):	47.2	17.4

(costs in microseconds)

Conclusion

- Latency has significant impact on simulation state
- Promising results from prototype:
 - circumvent latency errors
 - without sacrificing performance
- Results should be applicable to any similar ocean simulation



Improving the Accuracy of FFT-based GPGPU Ocean Surface Simulations

A master's thesis in computer science and engineering

