Quantum Coin Method for Numerical Integration

N. H. Shimada and T. Hachisuka

Introduction (4 min)

- Motivation
- Raytracing
- Monte Carlo

Previous research (8 min)

- Grover's | Abrams and Williams'
- How to accelerate?
- Why noisy result?

Our research (6 min)

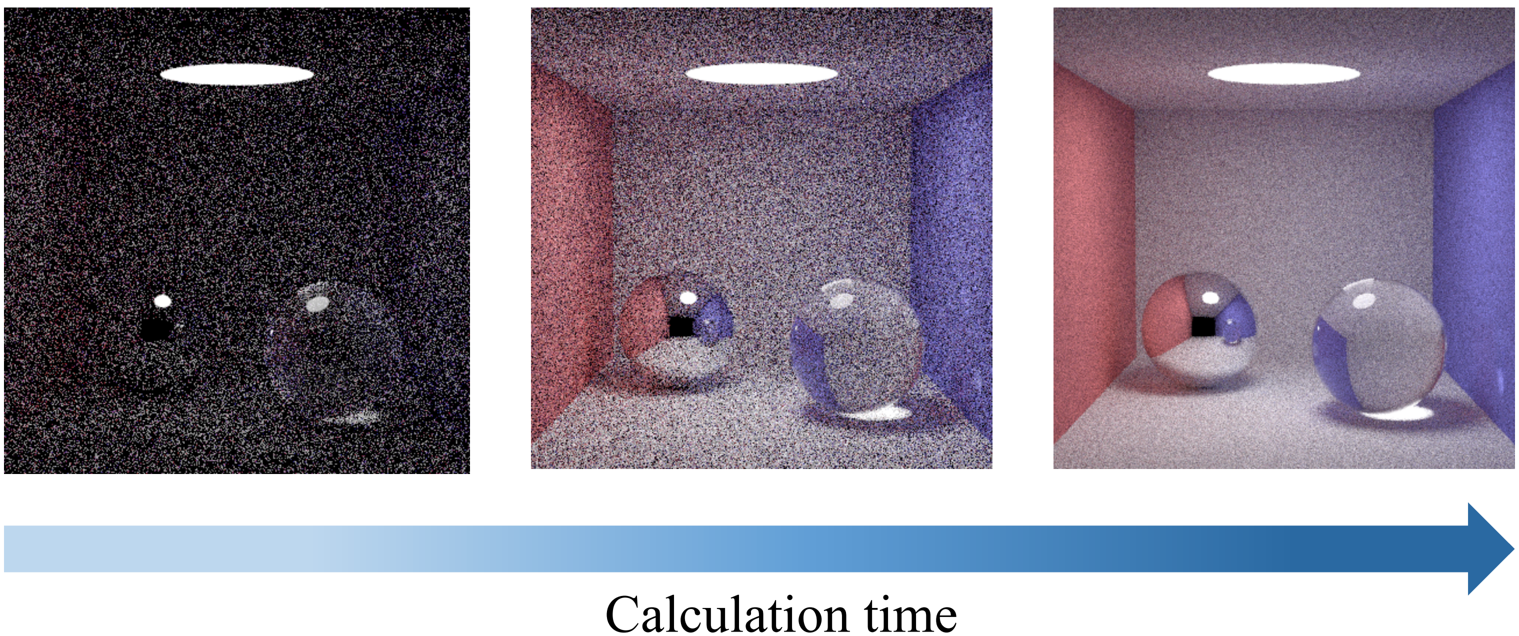
- Basic idea
- Our improvements
- Future work

Introduction

Motivation

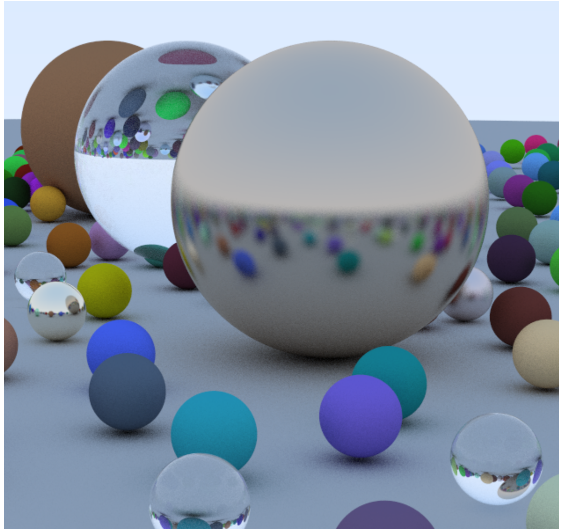
Quantum Computation for Computer Graphics

Noise of Raytracing



Raytracing

- Photorealistic images
  - Many applications (Movies, Games, VRs)



Spheres [1]

Volumetric Light Transportation [2]

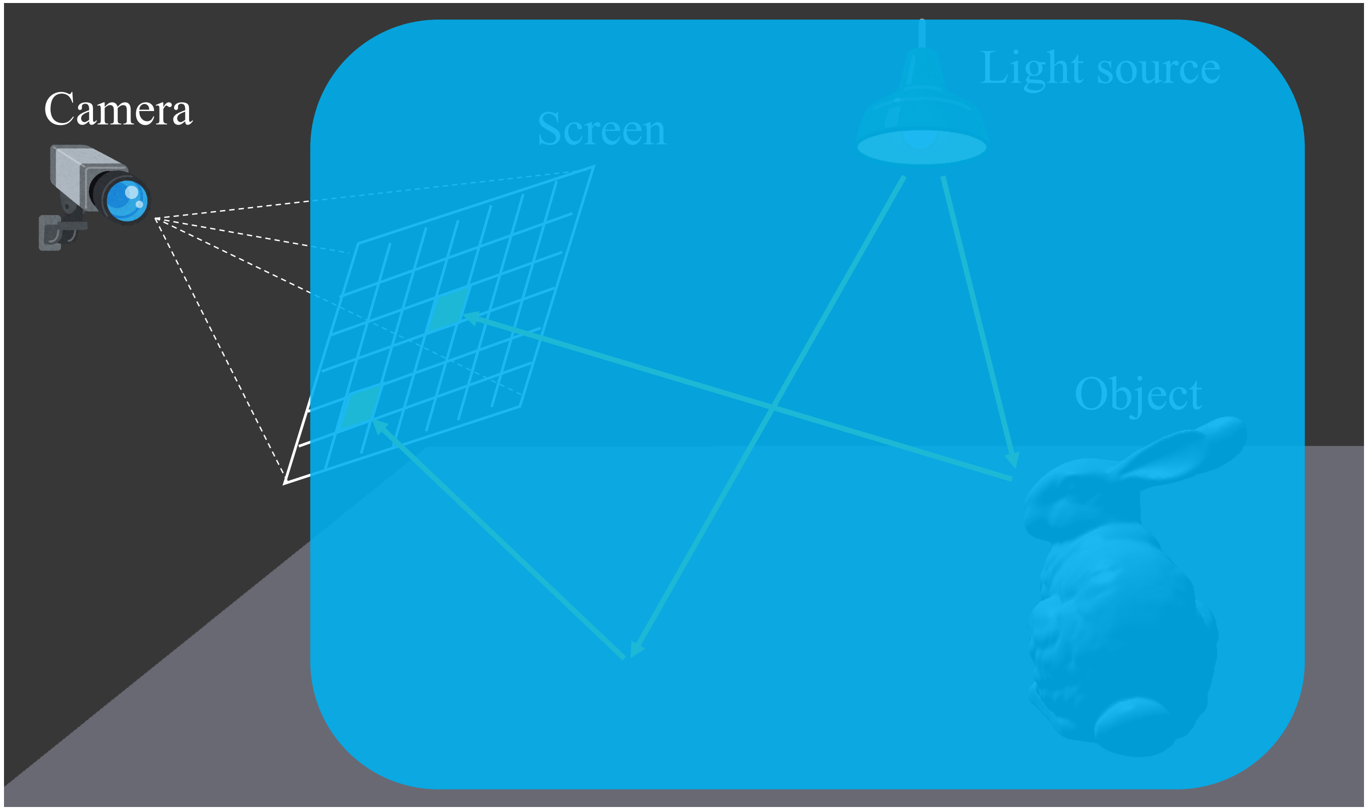
DirectX Raytracing with RTX [3]

[1] Implemented by the author
[2] J. Křivánek, et al., SIGGRAPH 2014
[3] Unreal Engine 2018

Raytracing

- Physically-based rendering
  - Modelling light in real world
  - Raytracing function $f(x)$

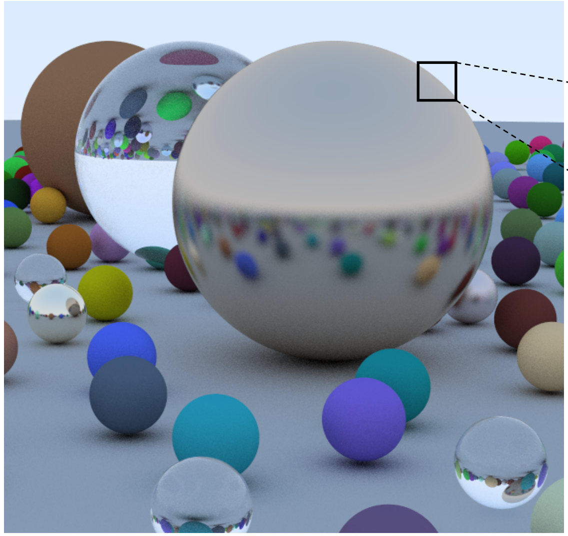
x: coordinates on screen
$f(x)$: intensity of light at x

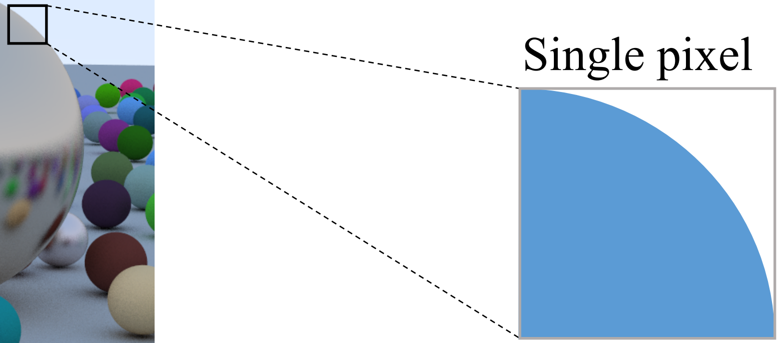


$\equiv f(x)$

Raytracing

- Noise of pixels
  - Numerical integration

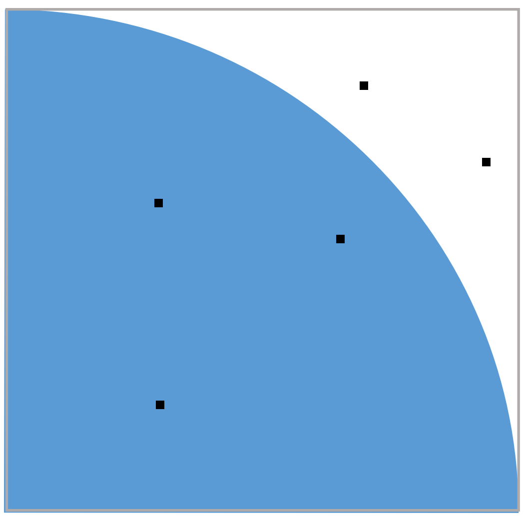




$$\int_S f(x)\, dx$$

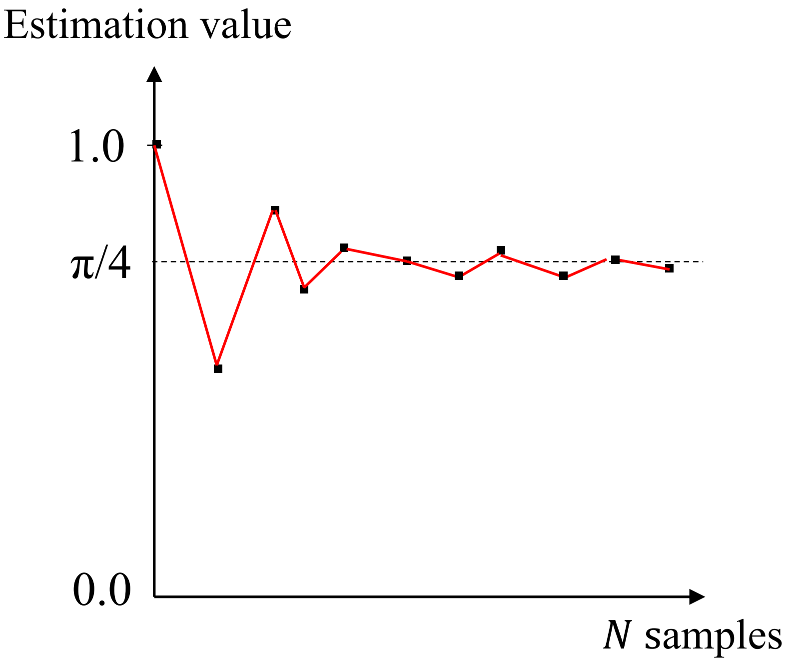
Raytracing

- Noise of pixels
  - Numerical integration
  - Monte Carlo estimation



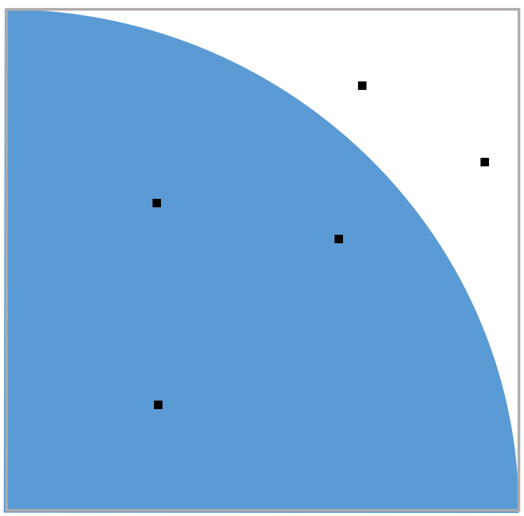
$$\int_S f(x)\,dx \simeq \frac{1}{N}\sum_{i=1}^{N} f(x_i)$$

Estimation error



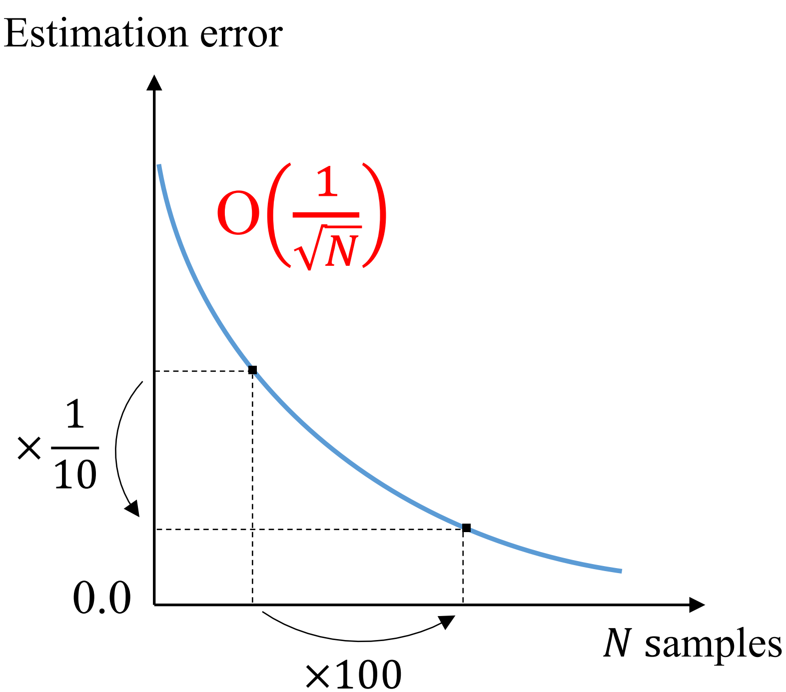
Raytracing

- Noise of pixels
  - Numerical integration
  - Monte Carlo estimation



$$\int_S f(x)\,dx \simeq \frac{1}{N}\sum_{i=1}^{N} f(x_i)$$

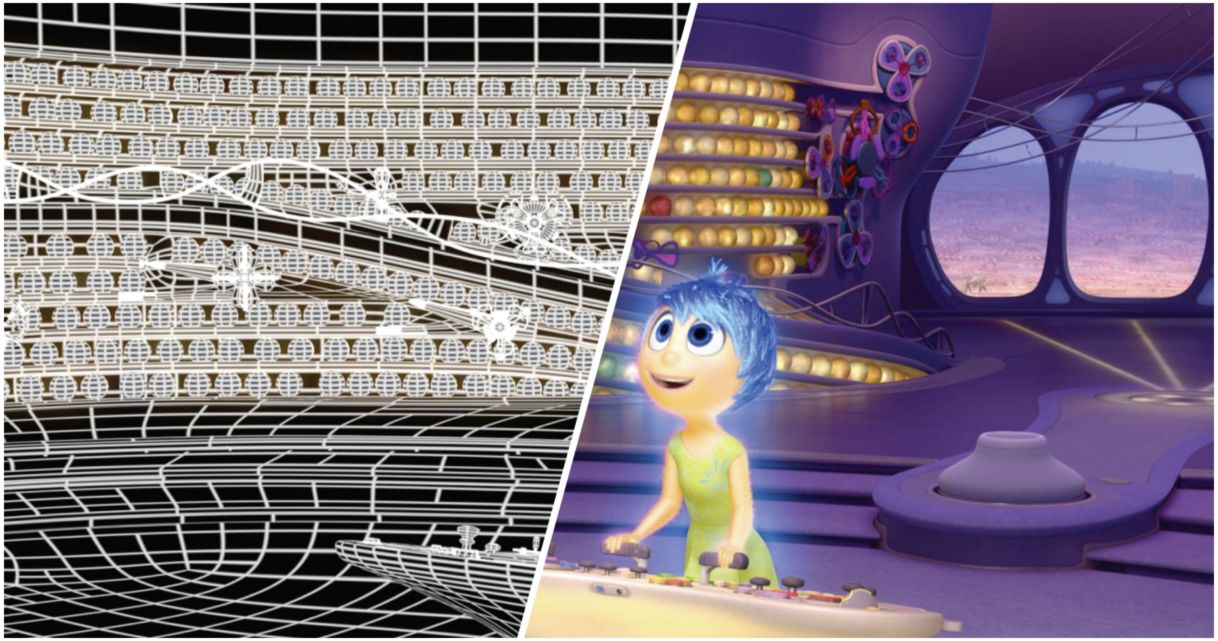
↑ Estimation error



slowly convergence …

Raytracing

- Industry application
  - PIXAR's animation movies
    1 day for 1 frame
    A huge 'render farm', but 2 years for rendering



3D scene → 2D image

*The Science Behind PIXAR

Introduction

- Raytracing:
  - Numerical integration + Blackbox function f(x)

- Numerical integration
  - Monte Carlo produces noise of pixels.
  - Faster convergent algorithms are important.

- Can we accelerate it with a Quantum Computer?
  - Yes …

Introduction (4 min)

- Motivation
- Raytracing
- Monte Carlo

Previous research (8 min)

- Grover's | Abrams and Williams'
- How to accelerate?
- Why noisy result?

Our research (6 min)

- Basic idea
- Our improvements
- Future work

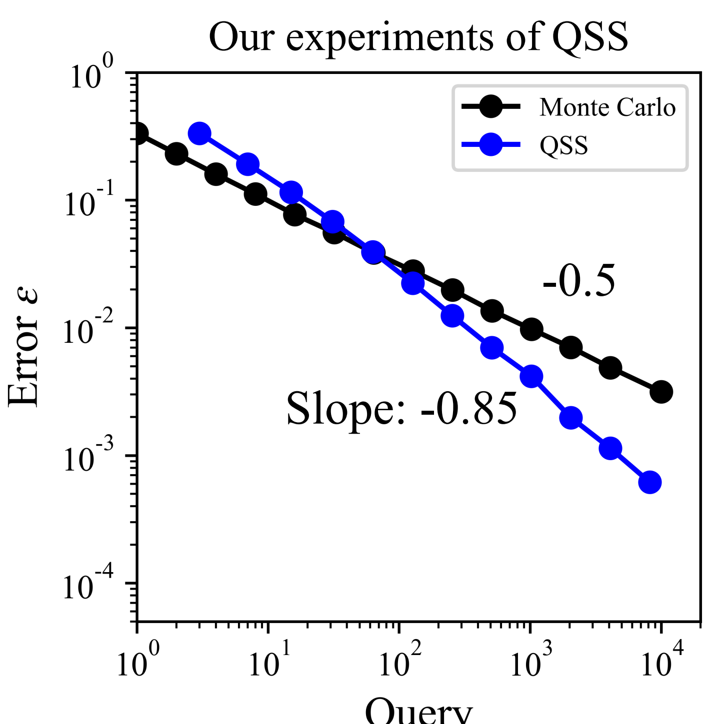
Previous Researches

- Grover's [1]

  O(1/N) error with N queries
  Using AA+QFT
  Implemented by Johnston [3] (QSS)

- Abrams and Williams' [2]

  O(1/N) error with N queries
  Using AA+Hybrid operation
  Implemented by no one



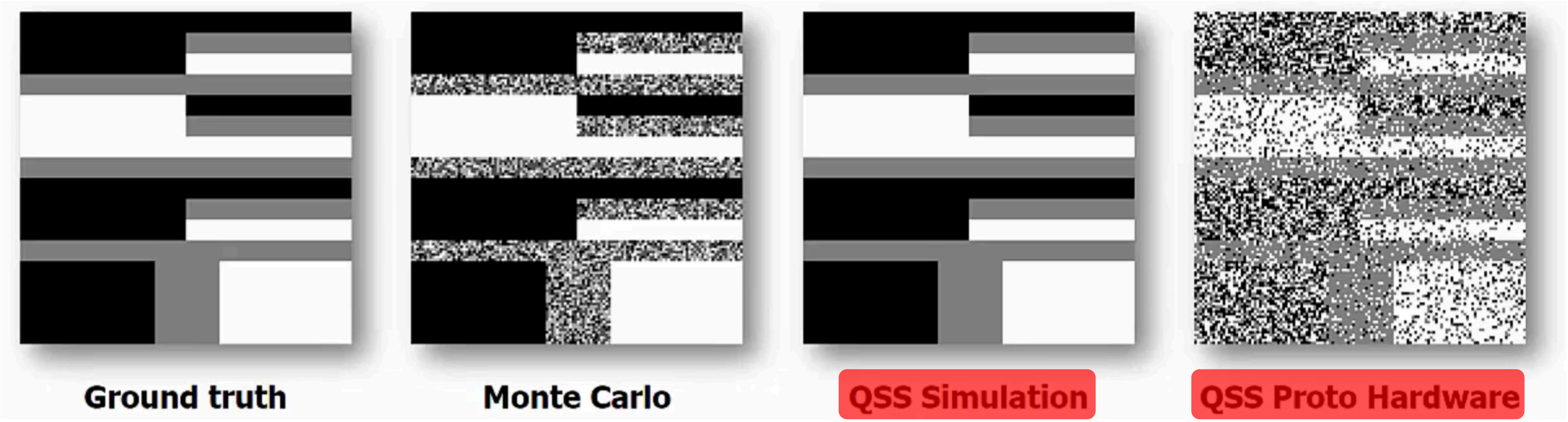
[1] STOC 1998
[2] arXiv 1999
[3] SIGGRAPH 2016

Previous Researches

- Grover's [1]
  - O(1/N) error with N queries
  - Using AA+QFT
  - Implemented by Johnston [3] (QSS)

- Abrams and Williams' [2]
  - O(1/N) error with N queries
  - Using AA+Hybrid operation
  - Implemented by no one

Johnston's experiments [3]



[1] STOC 1998
[2] arXiv 1999
[3] SIGGRAPH 2016

Previous Researches

Can we accelerate it with a Quantum Computer?

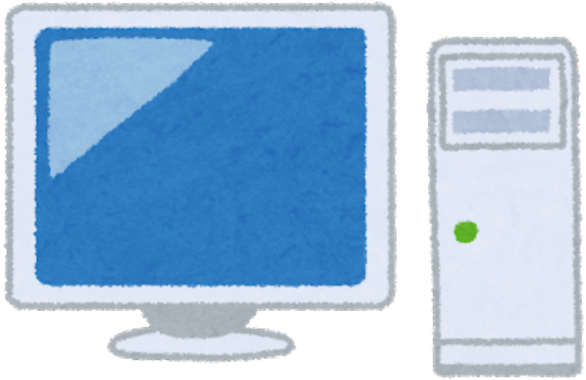
- Yes (on a simulator)

How to accelerate? (on a simulator)

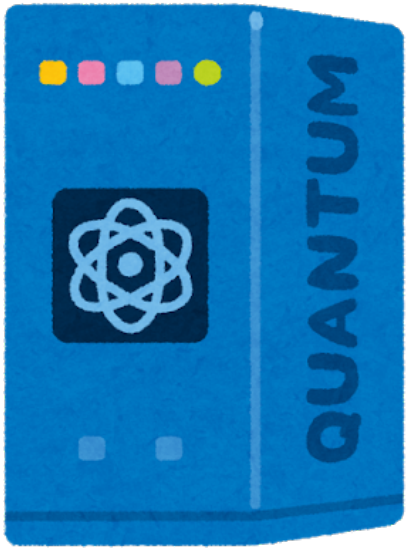
Why noisy result? (on an actual quantum computer)

How to accelerate?

Classical



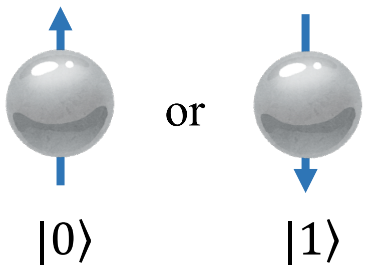
Quantum



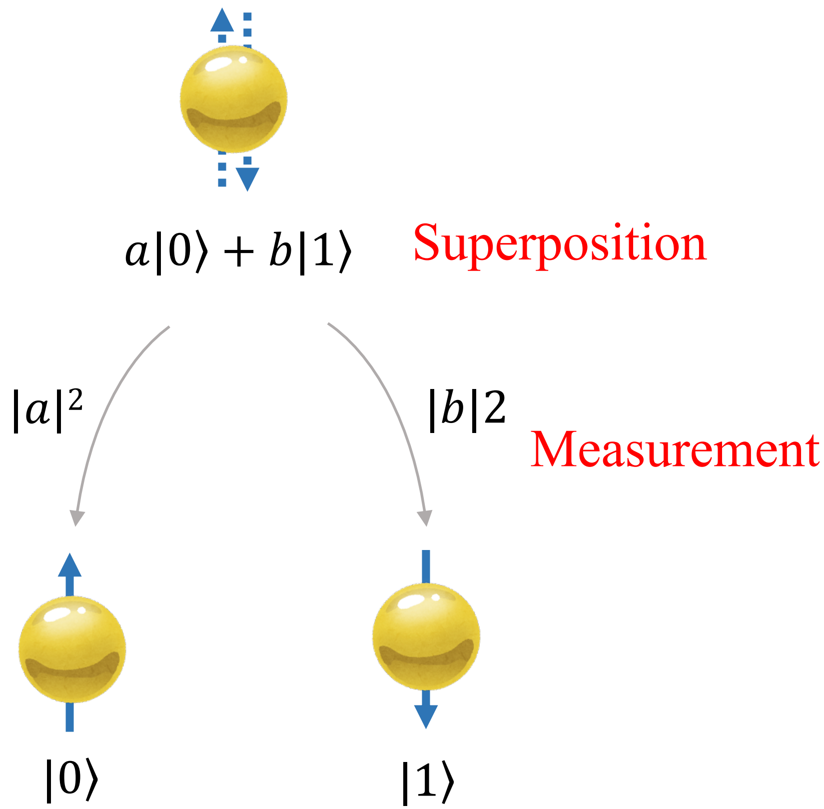
How to accelerate?

- A single bit

Classical



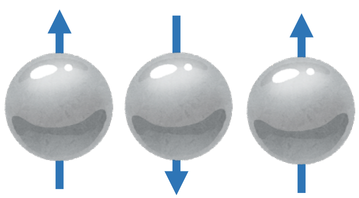
Quantum



How to accelerate?

- Multibit

Classical

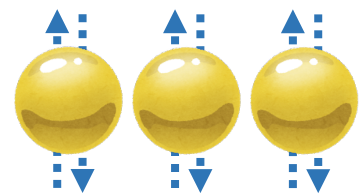


$|0\rangle \quad |1\rangle \quad |0\rangle$

$= |010\rangle$

1 state

Quantum



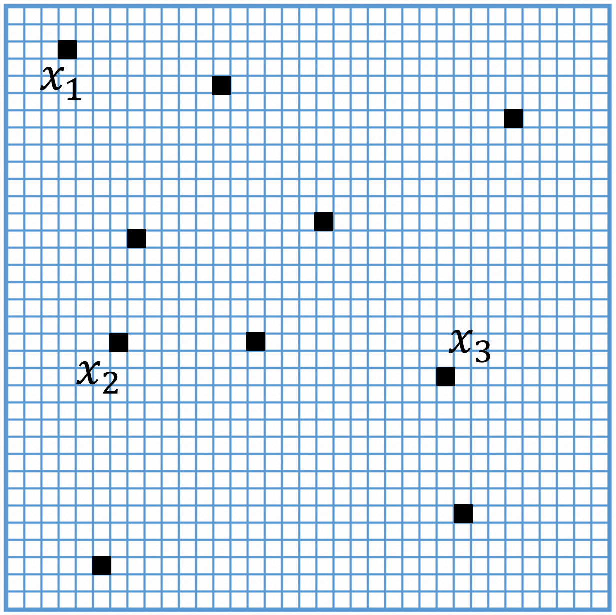
$(|0\rangle + |1\rangle)\,(|0\rangle + |1\rangle)\,(|0\rangle + |1\rangle)$

$= |000\rangle + |001\rangle + |010\rangle + |011\rangle$
$+ |100\rangle + |101\rangle + |110\rangle + |111\rangle$

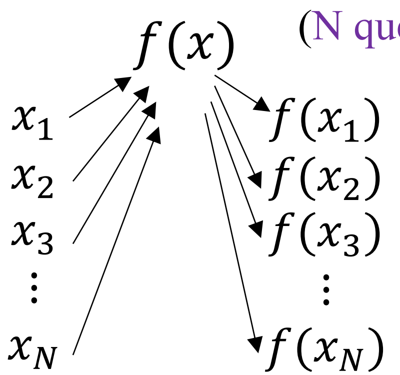
8 (=2^3) states

How to accelerate?

Monte Carlo

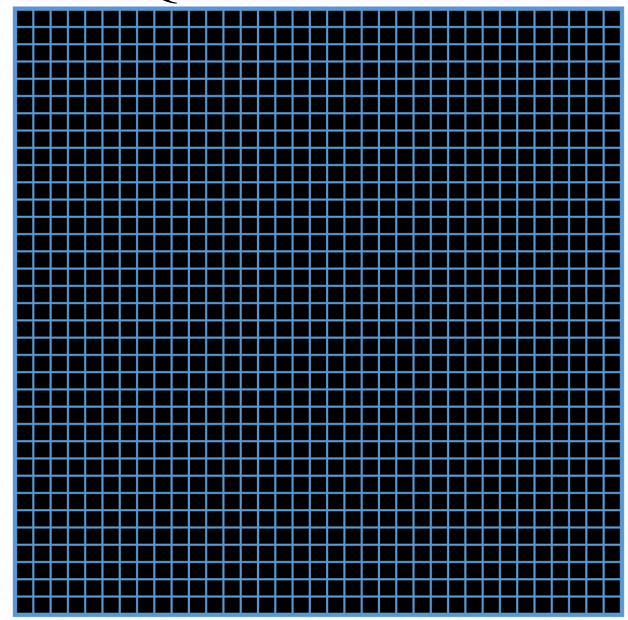


Fine grid with numerical precision
(ex. O(10^{-7}) in floating point)



(N queries)

Quantum case



Quantum circuit: $|x\rangle \longrightarrow f(x)|x\rangle$

(1 query!!)

$$\frac{1}{\sqrt{N}}\left(\begin{array}{l} \;\;|x_1\rangle \\ +|x_2\rangle \\ +|x_3\rangle \\ +\cdots \\ +|x_N\rangle \end{array}\right) \dashrightarrow \frac{1}{\sqrt{N}}\left(\begin{array}{l} \;\;f(x_1)|x_1\rangle \\ +f(x_2)|x_2\rangle \\ +f(x_3)|x_3\rangle \\ +\cdots \\ +f(x_N)|x_N\rangle \end{array}\right)$$

super position

How to accelerate?

Function $\hat{\mathcal{F}}$

$|x\rangle \longrightarrow f(x)|x\rangle$ (constant number of operations)

$$\frac{1}{\sqrt{N}}\begin{pmatrix} |x_1\rangle \\ +|x_2\rangle \\ +|x_3\rangle \\ +\cdots \\ +|x_N\rangle \end{pmatrix} \dashrightarrow \frac{1}{\sqrt{N}}\begin{pmatrix} f(x_1)|x_1\rangle \\ +f(x_2)|x_2\rangle \\ +f(x_3)|x_3\rangle \\ +\cdots \\ +f(x_N)|x_N\rangle \end{pmatrix} \dashrightarrow \frac{f(x_1)+f(x_2)+\cdots f(x_N)}{N}|00\ldots0\rangle + |others\rangle$$

$= f_{true}$ (True value)

How to accelerate?

$\hat{\mathcal{F}}$ (1 query) makes

$$|\psi\rangle = f_{true}|0\rangle + \sqrt{1 - f_{true}^2}|1\rangle$$

Are we finished?

- No!
- Result of measurement $|0\rangle$ or $|1\rangle$

How to know f_{true}?

- Makes many $|\psi\rangle$s and measure them
- Estimates $|f_{true}|^2$ from the ratio of $|0\rangle$
- N queries for $O(1/\sqrt{N})$ error (= same as Monte Carlo)
- Need some non-trivial tricks for faster convergence

How to accelerate?

- Amplitude Amplification (AA)

$$\mathcal{F}: |\psi_1\rangle = \cos\theta\,|0\rangle + \sin\theta\,|1\rangle$$

$\downarrow$ (2 queries)

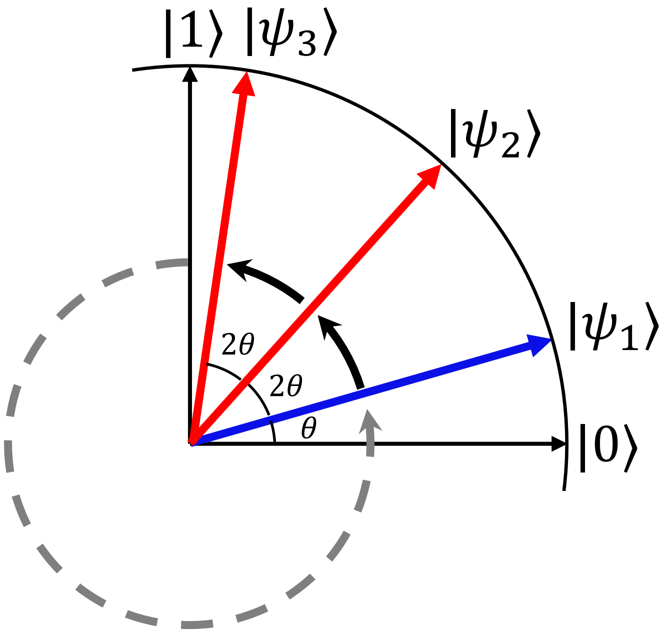
$$|\psi_2\rangle = \cos 3\theta|0\rangle + \sin 3\theta\,|1\rangle$$

$\downarrow$ (2 queries)

$$|\psi_3\rangle = \cos 5\theta|0\rangle + \sin 5\theta\,|1\rangle$$

$\downarrow$

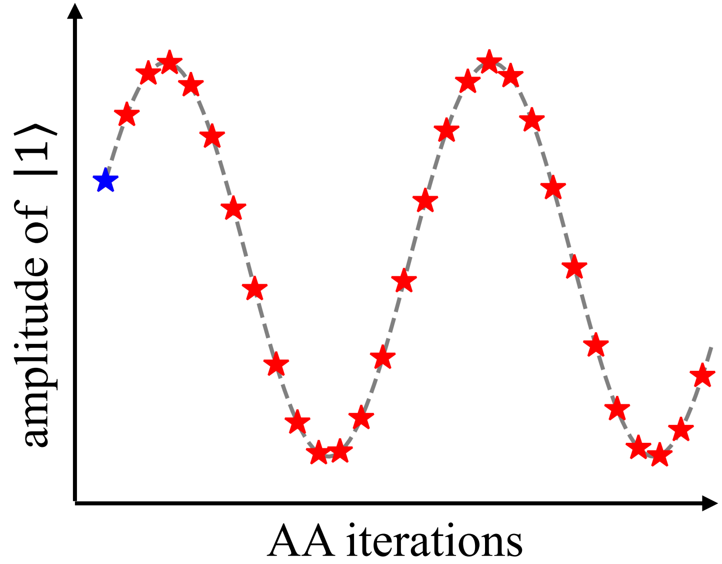
$\vdots$



- Quantum Fourier Transform (QFT)

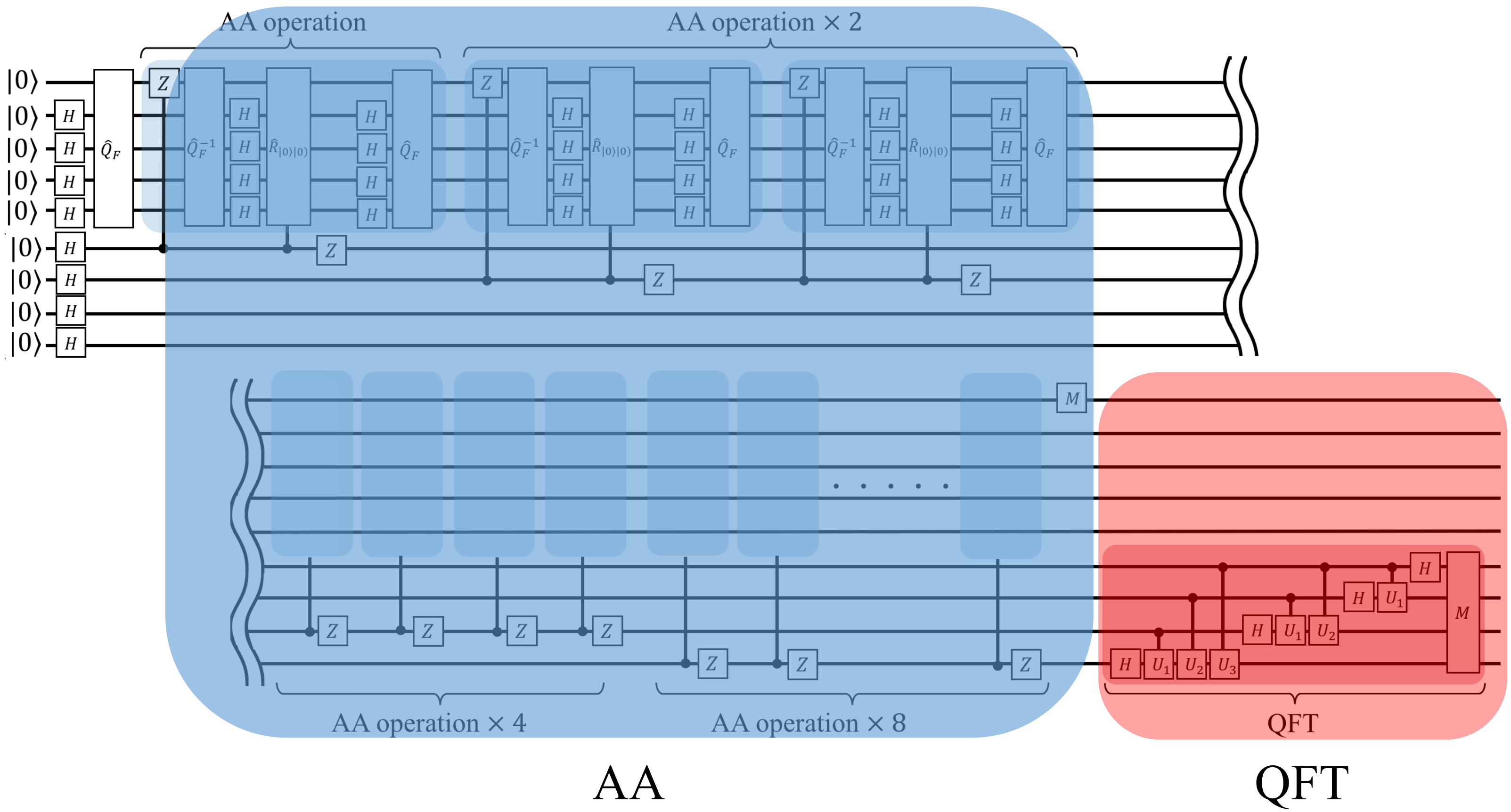
Estimate θ from an series of $|\psi_i\rangle$

O(1/N) error with N queries



Why noisy result?

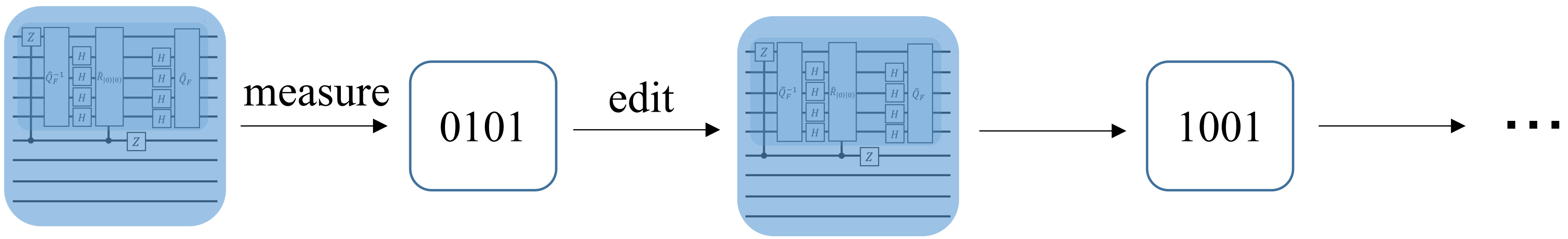
- QSS (AA+QFT)



Why noisy result?

- QSS
  - Deep depth circuits
  - Superposition is broken with time
  - → Noisy result

- Quatum-classical hybrid algorithm
  - Repeat {Quantum, Measurement, Classical}
  - Small depth circuits
  - → Noise-tolerant



small quantum circuit

Introduction (4 min)

- Motivation
- Raytracing
- Monte Carlo

Previous research (8 min)

- Grover’s | Abrams and Williams’
- How to accelerate?
- Why noisy result?

Our research (6 min)

- Basic idea
- Our improvements
- Future work

Our Research

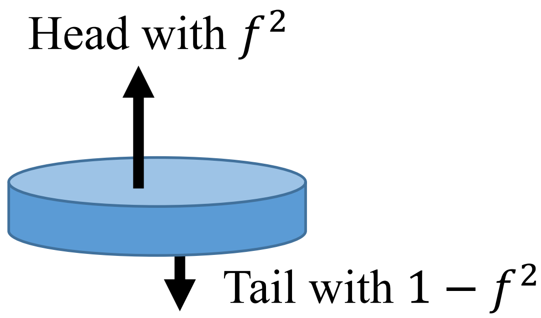
- Grover's [1]
  - O(1/N) error with N queries
  - Using AA+QFT
  - Implemented by Johnston [3]
  - (QSS)

- Abrams and Williams' [2]
  - O(1/N) error with N queries
  - Using AA+Hybrid operation
  - Implemented by us
  - (QCoin)

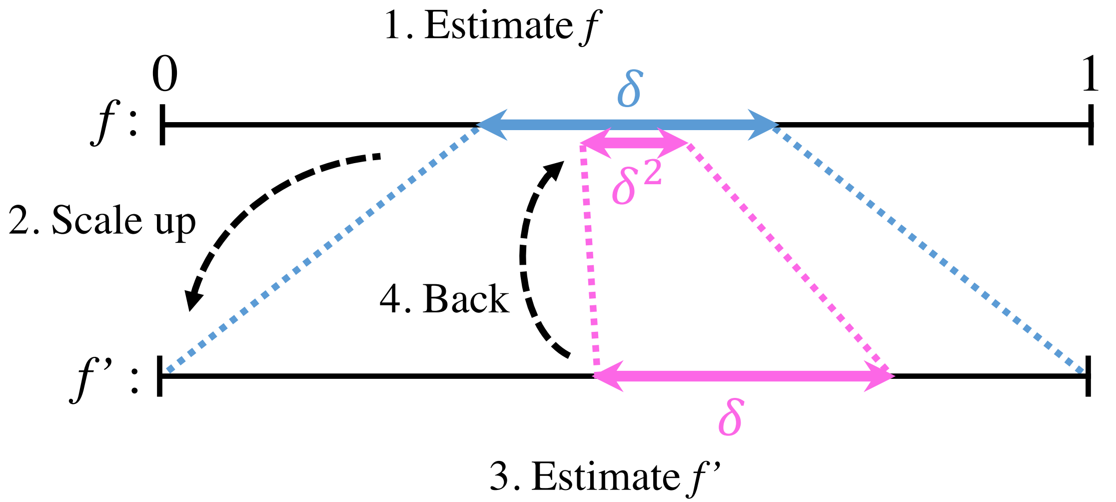
Abrams and Williams' idea

Quantum Coin

$$\mathcal{F}:\ |\psi\rangle = f|0\rangle + \sqrt{1-f^2}|1\rangle$$



- 1st step (Monte Carlo)
  δ error with $O(1/\delta^2)$ queries
- 2nd step (AA)
  Scale up by AA operation
- 3rd step (Monte Carlo)
  δ error with $O(1/\delta^2)$ queries
- 4th step (Classical calculation)
  Back to original scale
  → δ^2 error!



Abrams and Williams' idea

- Estimation error

  $\delta \to \delta^2 \to \delta^3 \to \ldots \to \delta^k$

- Queries

  $\begin{cases} \text{Monte Carlo: } O(1/\delta^2) \\ \text{AA: } O(1/\delta) + O(1/\delta^2) + \ldots + O(1/\delta^k) \end{cases}$

  $\to$ Total: $O(1/\delta^2) \times \{O(1/\delta) + O(1/\delta^2) + \ldots + O(1/\delta^k)\}$

  $= O(1/\delta^k)$ (if $k \to \infty$)

- Convergence rate

  δ^k error with $O(1/\delta^k)$ queries

  $\Leftrightarrow$ O(1/N) error with N queries

Our improvements

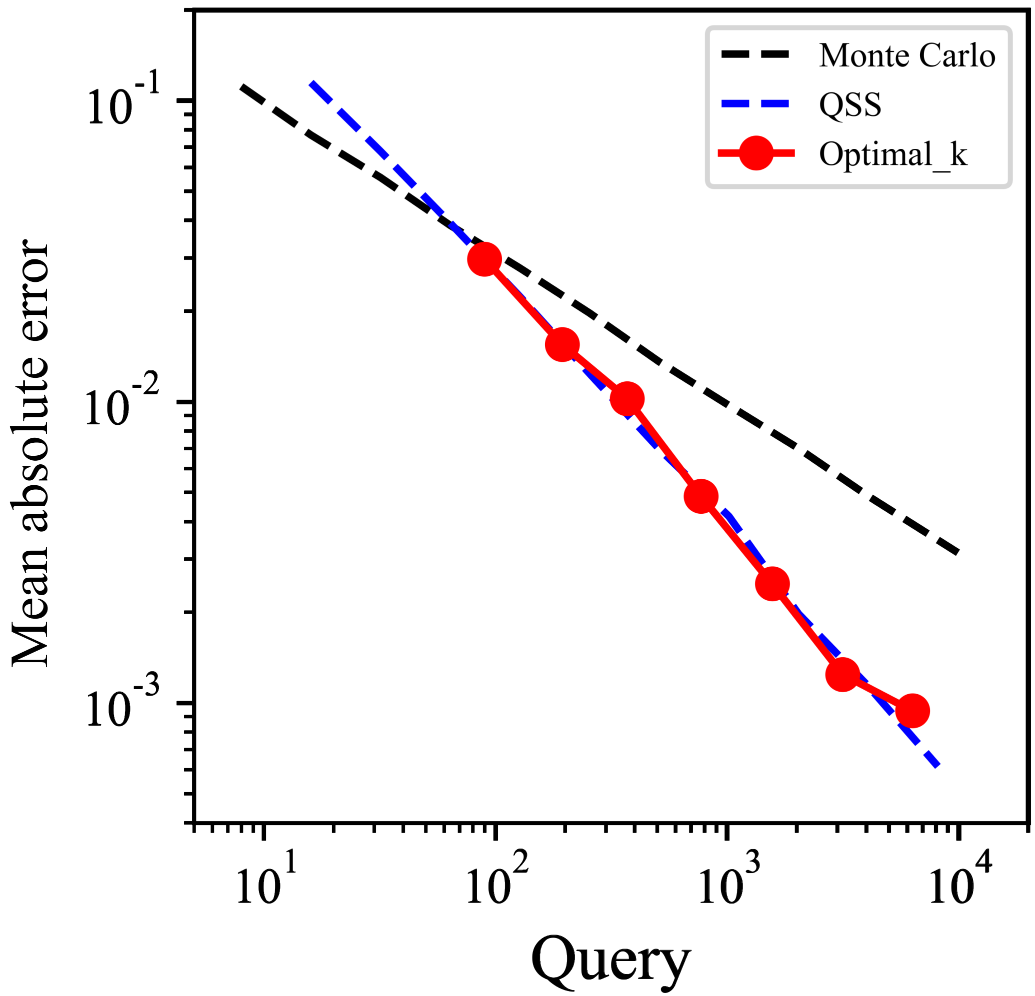
- Some concerns
  - Convergence rate with the usual number of queries in practice
  - Estimate f^2, not f
  - In Monte Carlo step, true value will be out of estimation range

- Numerical experiments
  - Implemented for the first time
  - Confirmed the error convergence compatible to QSS's

Our improvements

- Numerical experiments

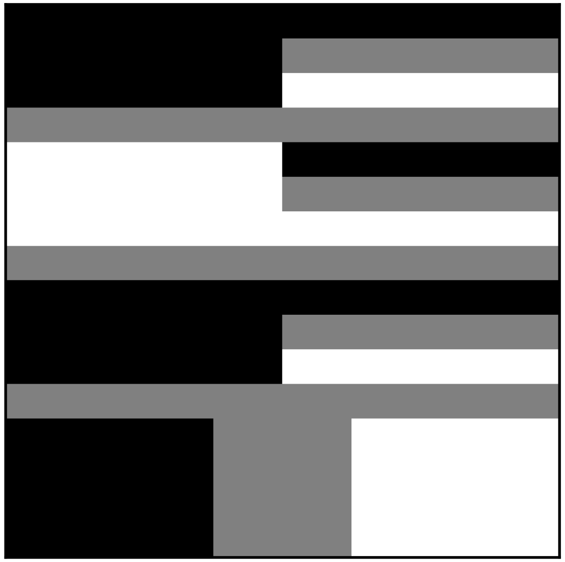
  3000 samples of random f_{true} for each point



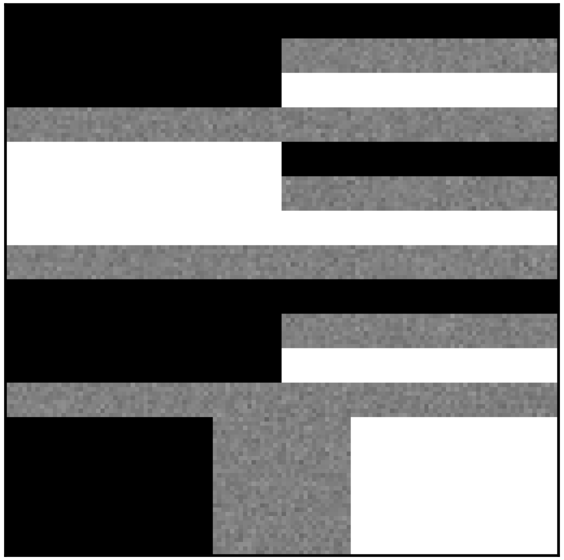
Our improvements

- Supersampling image

Ground truth

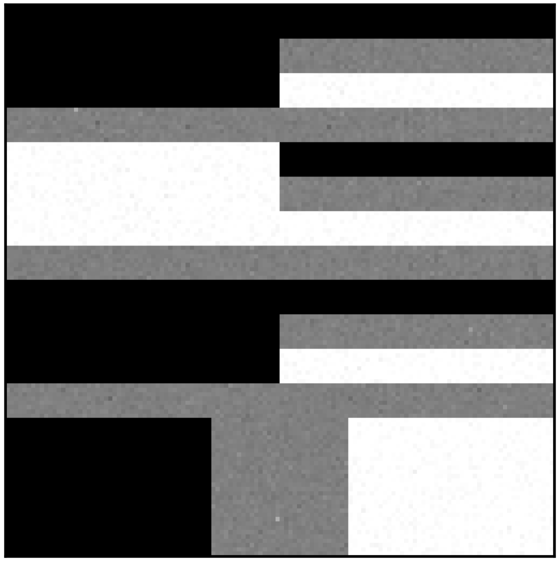


Monte Carlo



0.026

QCoin
on simulator



0.014

(Average error in gray pixels)

Our improvements

- Some advantages

  Hybrid algorithm with small quantum circuits

  → Noise-tolerant

  → Few qubits

- Experiments on an actual quantum computer

  Implemented for the first time

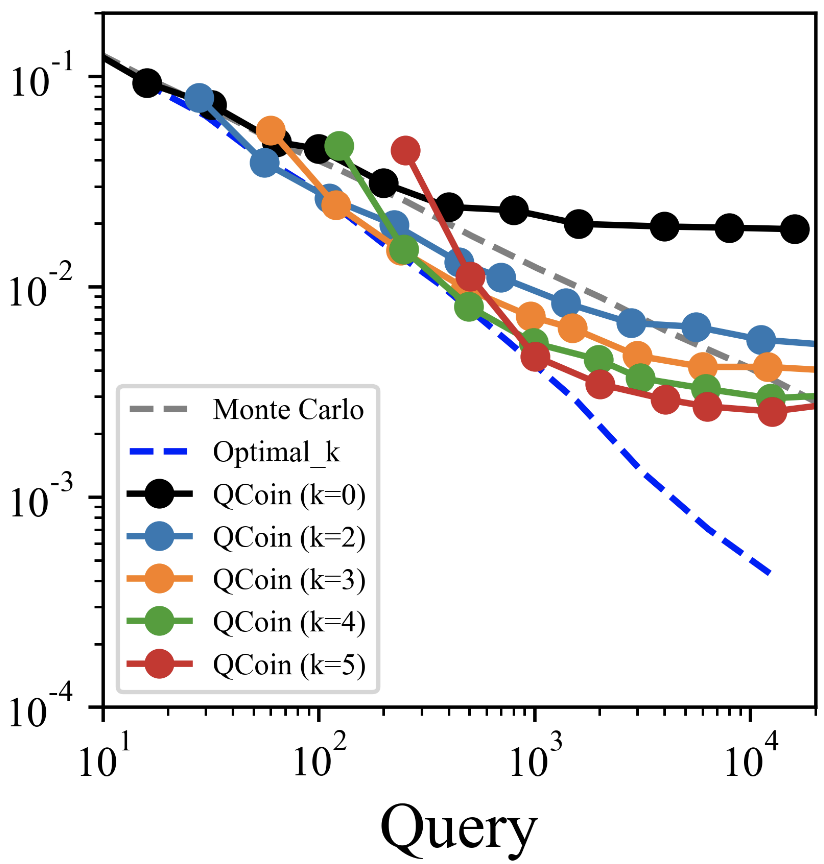
  Confirmed the advantages

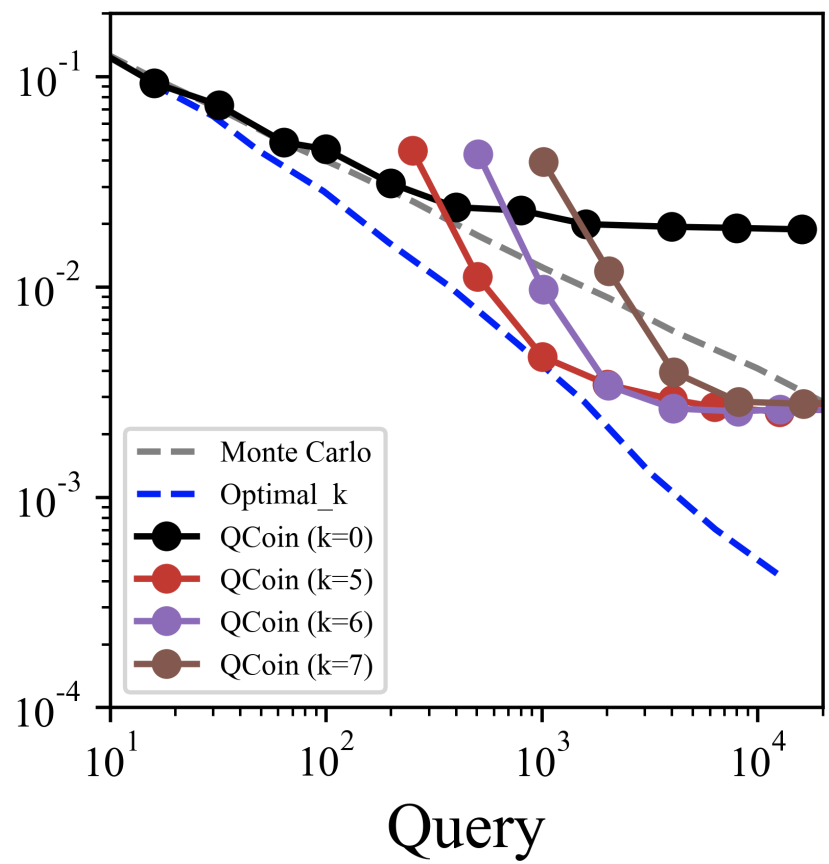
Our improvements

- On an actural quantum computer (IBMQ 5 qubits in 2019)

  ≤ 1000 queries: Good

  > 1000 queries: Bad (← too large quantum circuits)

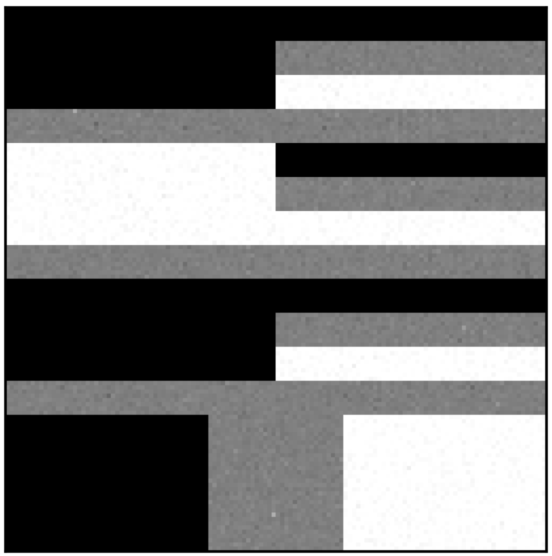




Our improvements

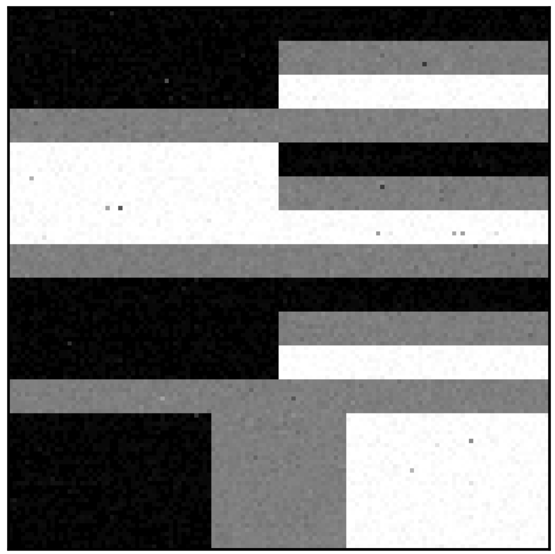
- Supersampling image (with 240 queries)

QCoin
on simulator



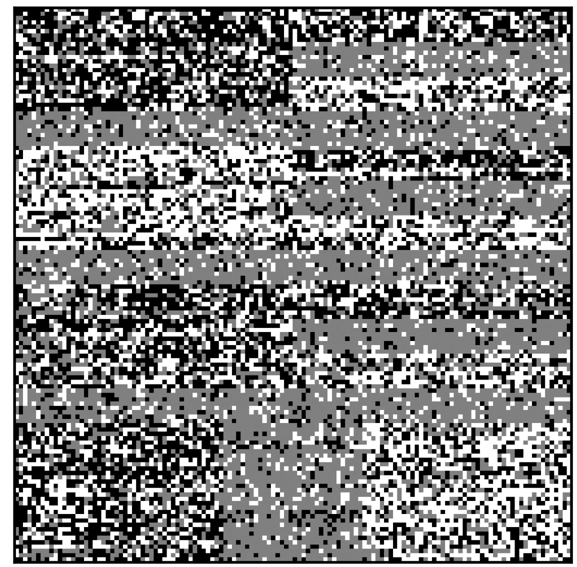
0.014
(Average error in gray pixels)

QCoin
on quantum computer



0.015

QSS
on quantum computer



Future work

- Implementation of a Raytracing function $f(x)$
  - Be hard due to hardware limitation
    - Much more qubits
    - No longer feasible on a simulator
  - Non-trivial way of calculating a function

- Combined with "Quasi-Monte Carlo"
  - Sampling with a low-discrepancy sequence
  - $O(\log(N)^s/N)$ error

- "Adaptive sampling" and "Denoising"
  - Save calculation cost referring to neighbor pixels
  - Useful for application of CG in the future