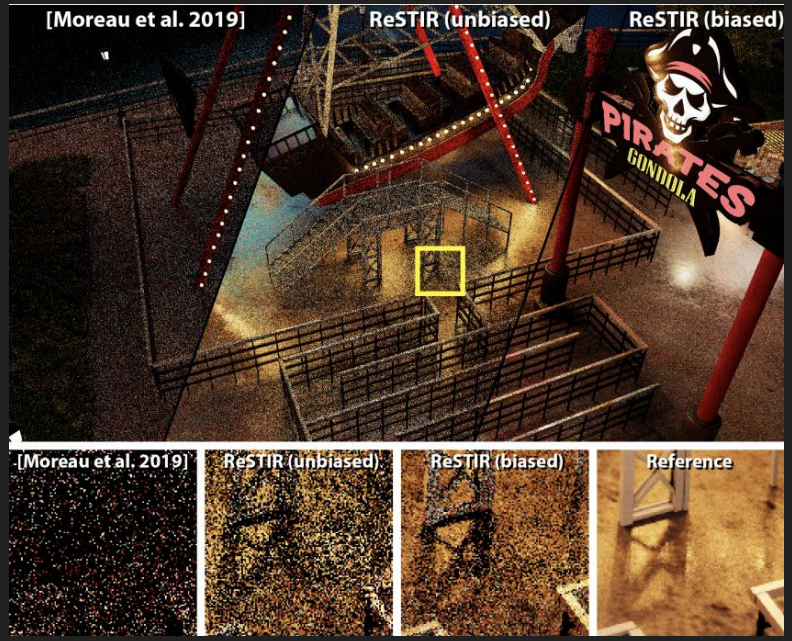
ReSTIR with DirectX Ray Tracer

Sydney Miller, Sireesha Putcha and Thy Tran

ReSTIR Overview

For each pixel:

1. Select 1 light from 32 randomly chosen lights
2. Shoot a shadow ray from the light to pixel. If it is obscured, discard selection.
3. Compare the light used in the last iteration to the light from step 2. Choose one.



4. Compare the lights from random adjacent pixels to light from step 3. Choose one.
5. Shoot a shadow ray to light from step 4 and shade pixel.

Use a combination of Weighted Reservoir Sampling (WRS) and Resampled Importance Sampling (RIS) to select and compare lights.

ReSTIR: Decide Sampled Light per Pixel (WRS and RIS)

- Each pixel has a reservoir that holds: current light and sum of all weights seen
- Each light candidate has a weight, corresponding to their chance of being chosen as the current sampled light for a pixel
    - weight = (bsdf * light_emittance * lambert * solid_angle) / light_pdf

Spatial Reuse

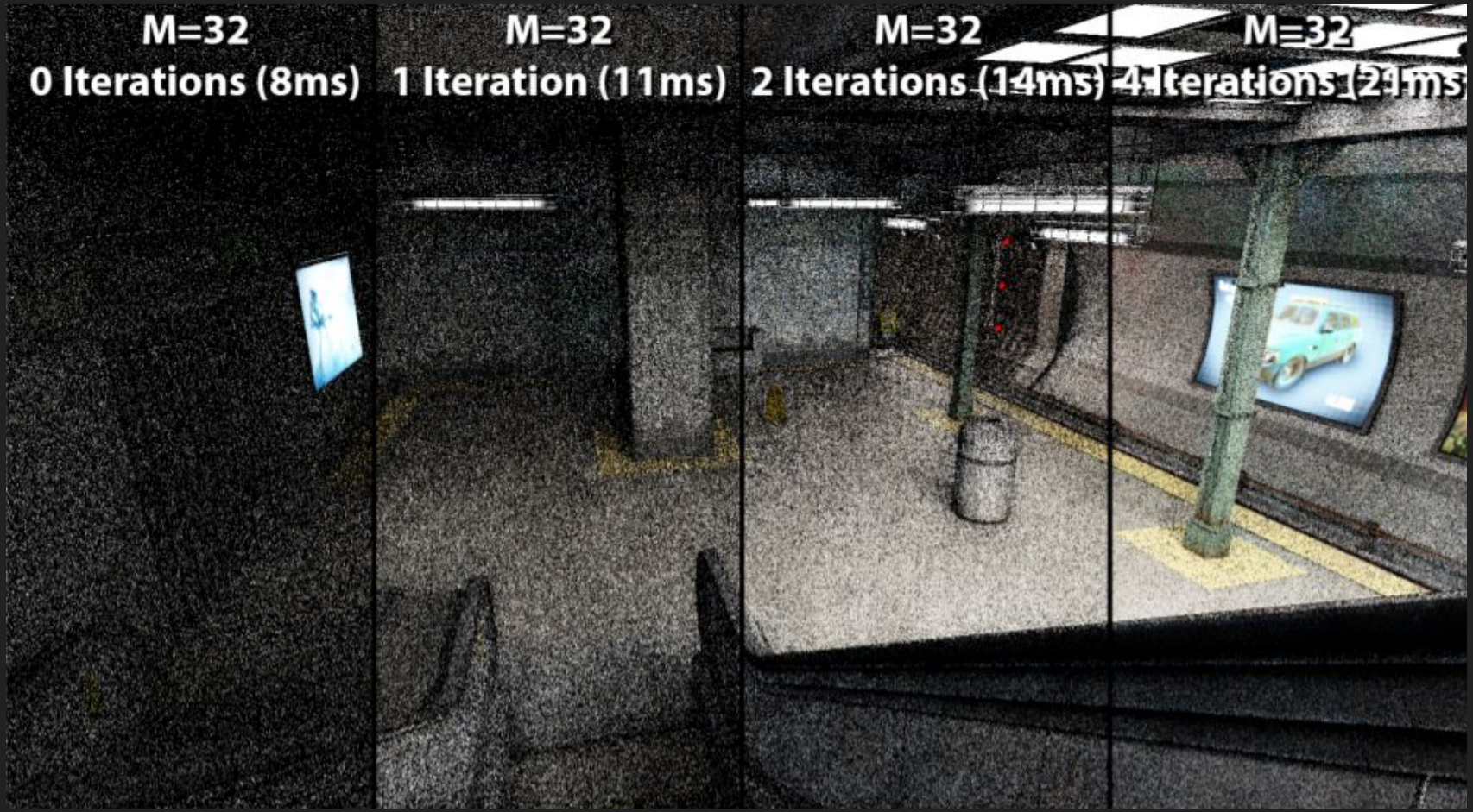
- Generate M (= 32) candidates for every pixel using RIS.
- Store the resulting reservoirs in an image-sized buffer.
- Each pixel selects k(= 5) of its neighbors and combines their reservoirs with its own.

Temporal Reuse

- Images often rendered as part of an animation sequence.
- Prior frame can provide useful info for reuse.
- Store each pixel’s final reservoir after rendering it.
- Feed forward these reservoirs for reuse in sequentially forward frames.

Visibility Reuse

- Evaluate the visibility of the selected sample of each reservoir
- Hence, occluded samples will not propagate to neighboring pixels
- Discard the reservoir if a sample is occluded


M=32
0 Iterations (8ms)
M=32
1 Iteration (11ms)
M=32
2 Iterations (14ms)
M=32
4 Iterations (21ms)

Ray Tracing With DirectX

- Followed tutorials and read through base code for DirectX ray tracing
  - Rendering pipeline
    - setting multiple passes
    - setting up buffers
    - passing variables into shaders
  - Basic ray tracing
    - direct lighting
    - global illumination
- Uses Falcor library by NVIDIA
  - DirectX ray tracing abstraction
  - Complex scene/mesh loading

Image rendered with our current directX ray tracer

Goal for the Next Milestone

- Implement our ReSTIR knowledge in our DirectX ray tracing code
- Point Lights
- Non-moving frames

ReSTIR: Decide sampled light with neighbors

- Each pixel has a reservoir
- Each reservoir holds: current light and sum of all weights seen
- Combine reservoirs of adjacent pixel light streams without increasing storage
- Treat all reservoirs as new samples where weight = sum of weights seen
- The resulting reservoir will have a chosen light and weight
- Really fast

ReSTIR Overview

The NVIDIA video and slides do not list out

- how to sample a light given the initialized 32 candidates per pixel
- how to decide between a pixel's current sampled light and its neighbors' sampled light
- how to decide between a pixel's current sampled light and its past frames' sampled light

→ need to look through the research paper and SIGGRAPH videos on ReSTIR