



# SurfelPlus

A project by Zhen Ren, Ruipeng Wang and Jinxiang Wang



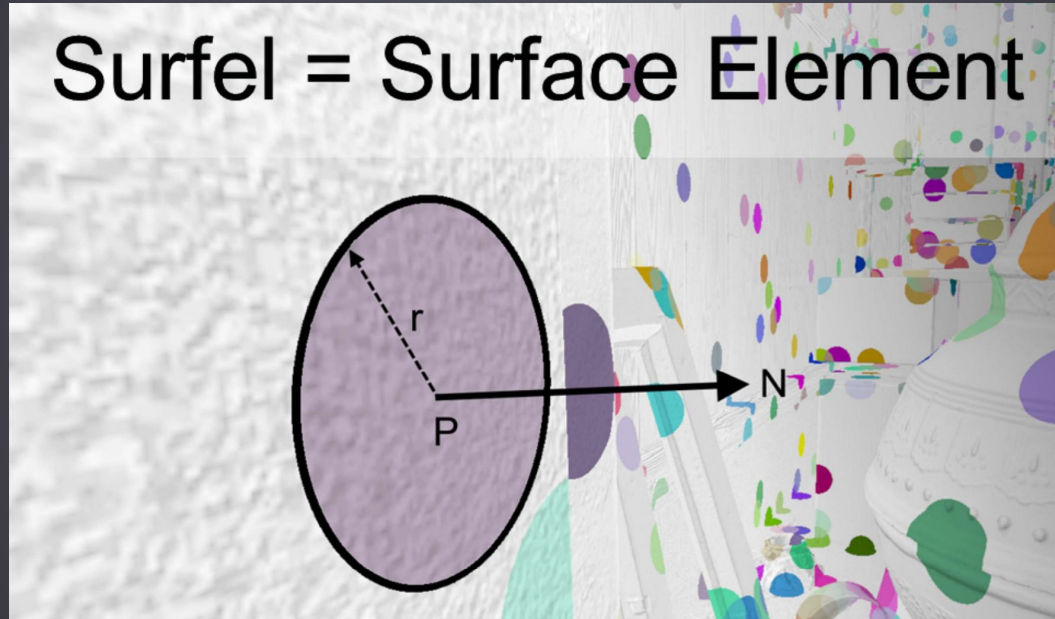
Penn  
Engineering  
UNIVERSITY of PENNSYLVANIA

# Introduction



SurfelPlus is a real-time dynamic global illumination renderer

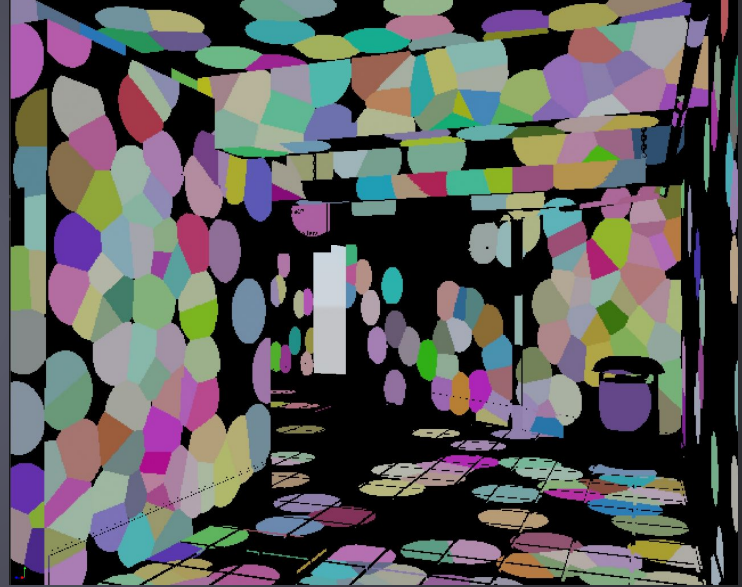
# What is “surfel” ?



Surfel = radius + position + normal

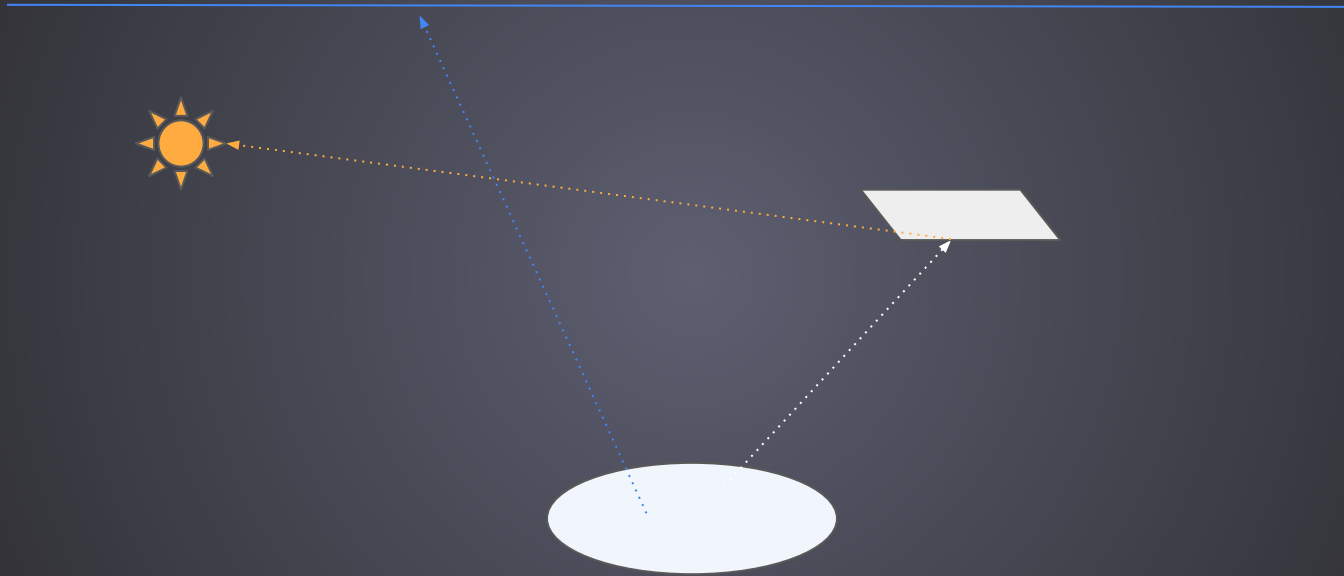


# Algorithm Overview



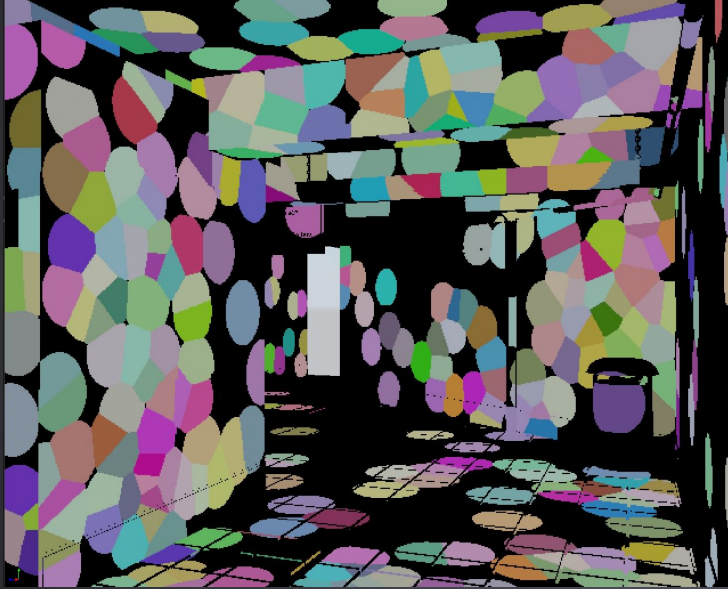
We attach surfels to object surfaces and use it as scene cache

# Algorithm Overview



Surfels would shoot rays and gather environment lighting throughout frames

# Algorithm Overview



Each pixel can then use nearby surfels to calculate indirect diffuse lighting

# Render Passes Overview

## Prepare Stage

Gbuffer Pass

## Surfel Calculation Stage

Surfel Prepare Pass

Surfel Update Pass

Cell Info Update Pass

Cell to Surfel Update Pass

Surfel Ray Trace Pass

Surfel Integrate Pass

Surfel Generation & Evaluation Pass

## Reflection Calculation Stage

Reflection Trace Pass

Spatial-Temporal Filtering Pass

Bilateral Filtering Pass

## Integrate Stage

SSAO Pass

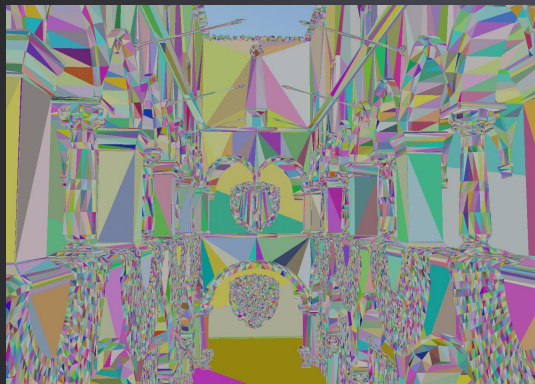
Light Integrate Pass

TAA Pass

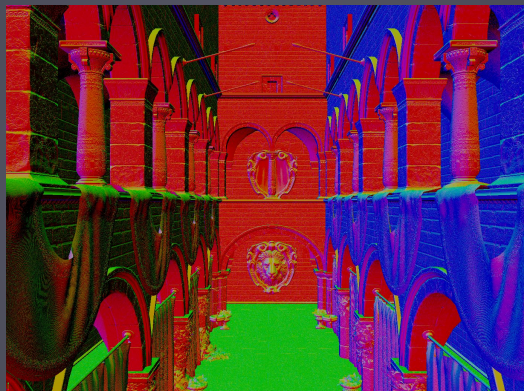
Tone Mapping Pass

## Frame End

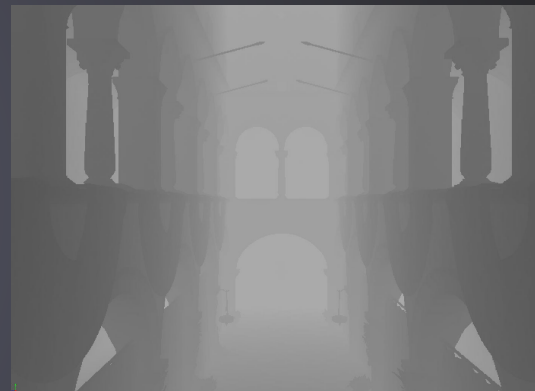
# Prepare Stage



Visibility



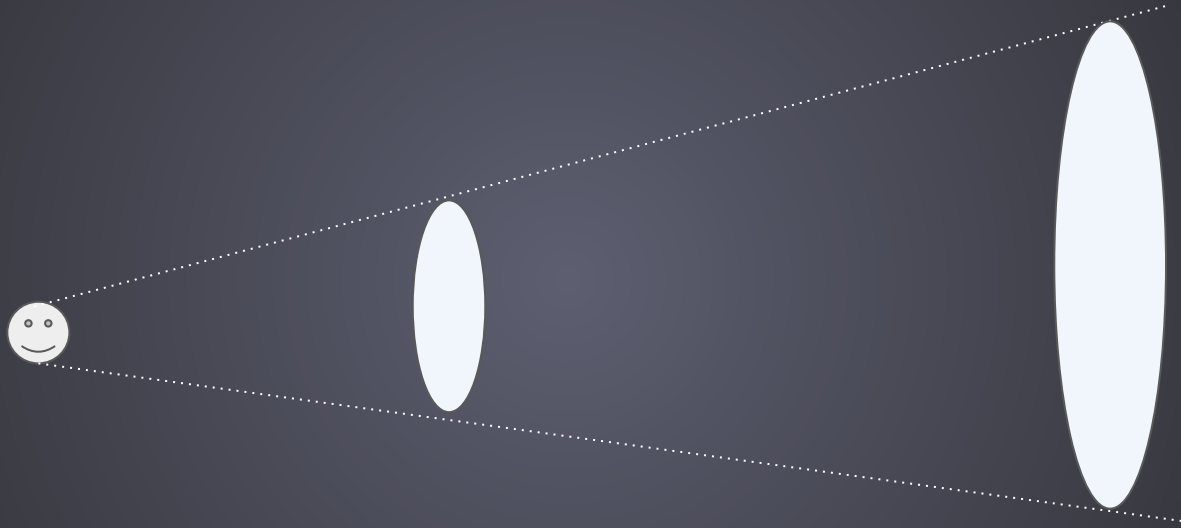
Compressed  
Normal



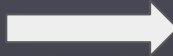
Depth



# Surfel Calculation Stage



Surfel Prepare Pass



Update Surfel radius based on distance  
so that surfels sizes stay same on screen

# Surfel Calculation Stage

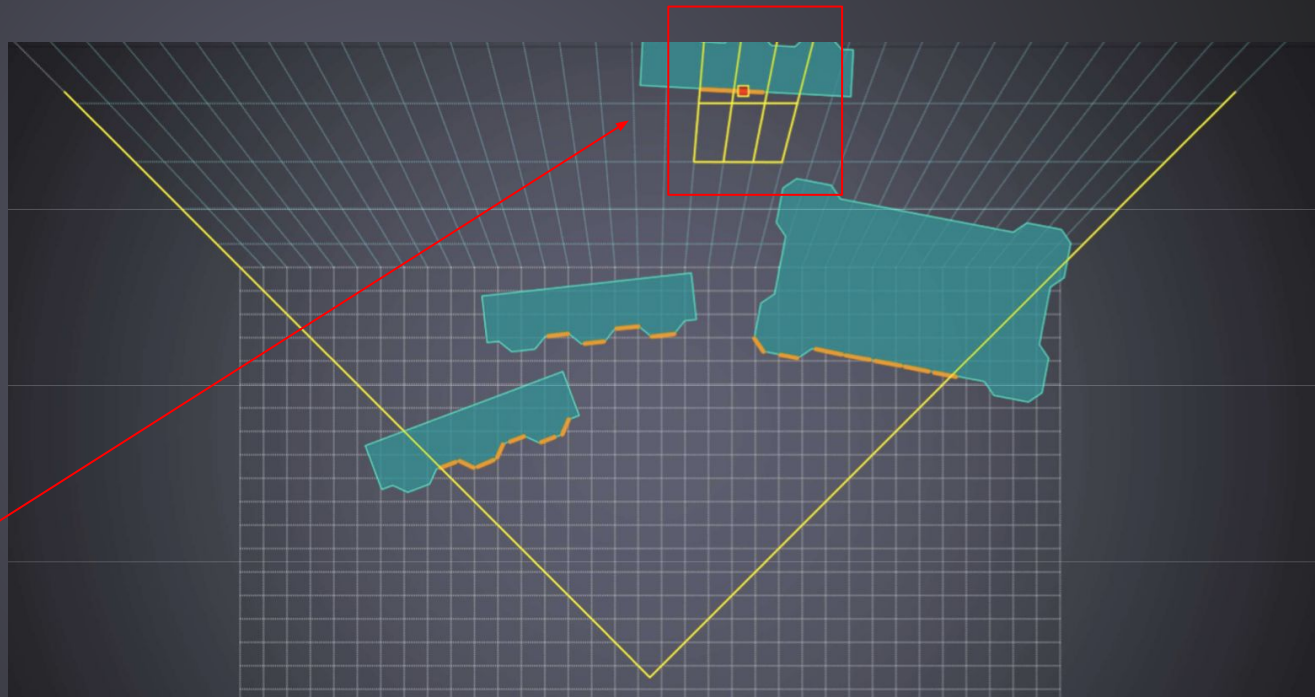
Surfel Update Pass

Cell Info Update Pass



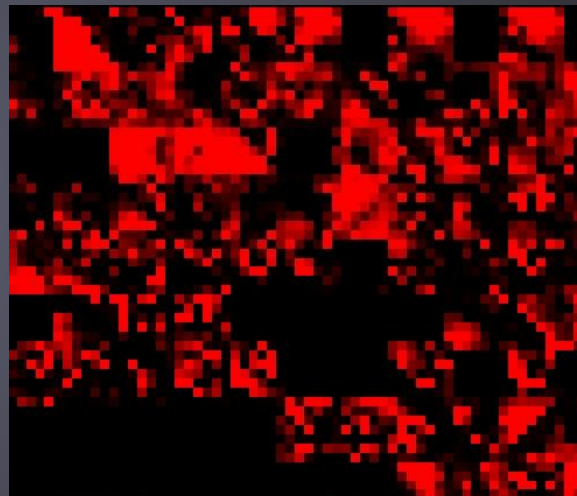
Construct Grid-based  
Acceleration Structure

Allocate Surfels in Cell



# Surfel Calculation Stage

- Surfel Ray Trace Pass
  - Generate rays to collect scene lighting
  - Use irradiance map to do ray guiding

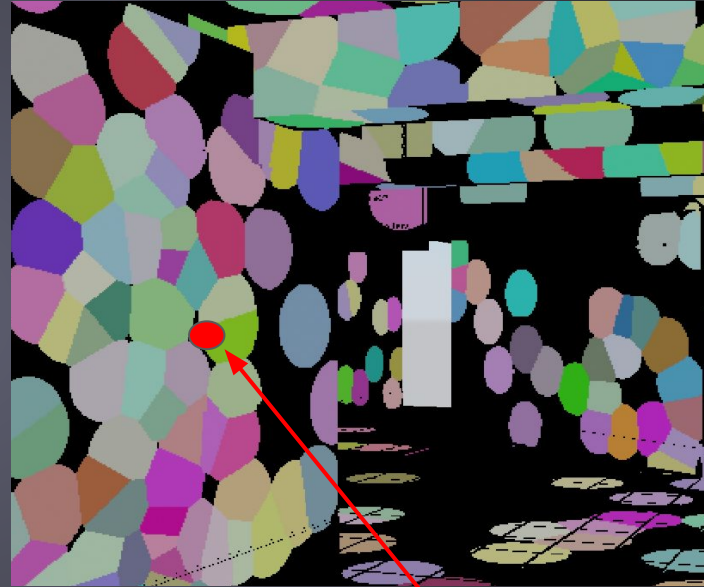


irradiance map

# Ray Trace Optimization

## Finalize path using surfels

Use surround surfels to give path a contribution. This step accelerates convergence.





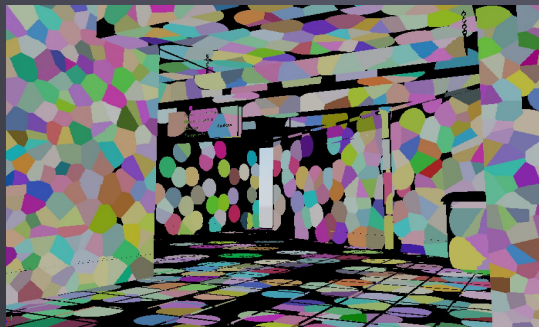
# Surfel Integrate Pass

- Gather rays
- Share Irradiance
- Stochastic Sample



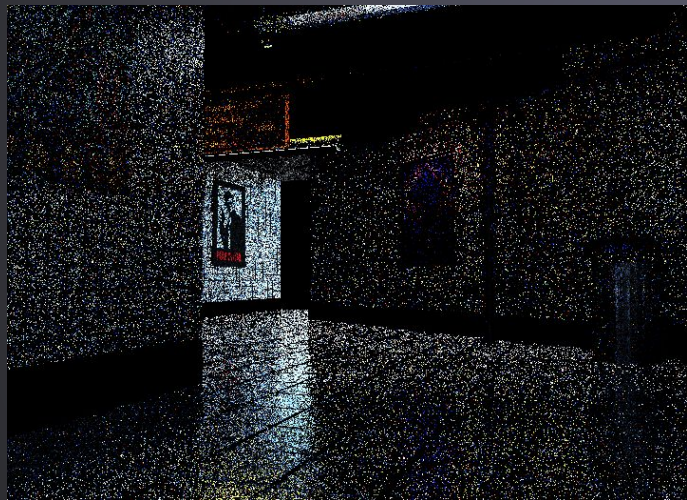
# Surfel Generation & Evaluation Pass

- Generate surfels based on screen coverage
- Recycle surfels if it's too dense
- Evaluate pixel lighting using surrounding surfels
- Use half resolution map to optimize performance

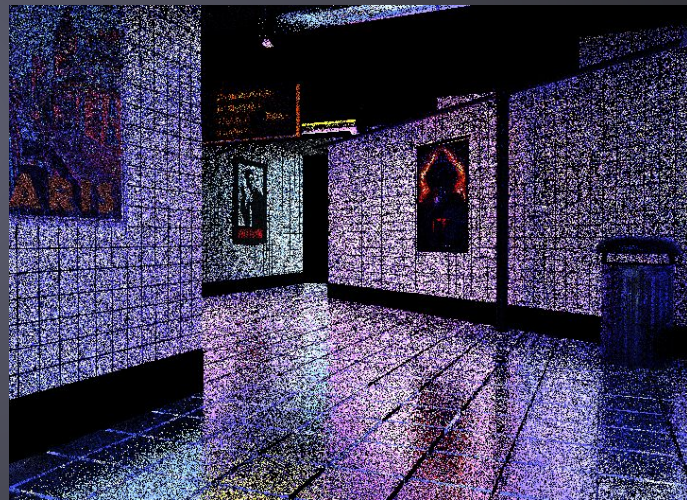


# Reflection Calculation Stage

- Use RIS to generate samples
- Terminate path using surfels
- Half resolution texture



6 Bounces **without** surfel termination



1 Bounce **with** surfel termination

# Filtering & Denoising

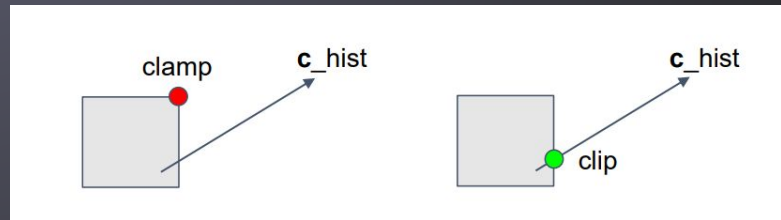
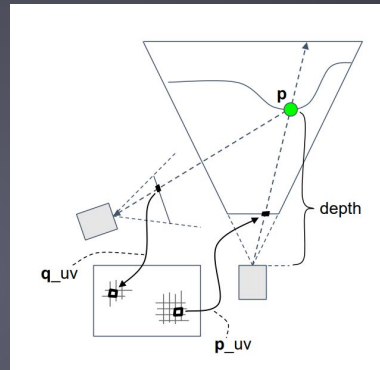
- Spatial filtering
- Temporal accumulation
- Bilateral filtering





# Temporal Reprojection Anti-Aliasing

- Reprojection:
  - previous frame UV
- Neighbor Clipping:
  - Clip color towards history color.
- Sharpen:
  - Apply Laplace operator to the final color



# Temporal Reprojection Anti-Aliasing

TAA Off



TAA (Unsharpened)



TAA (Sharpened)

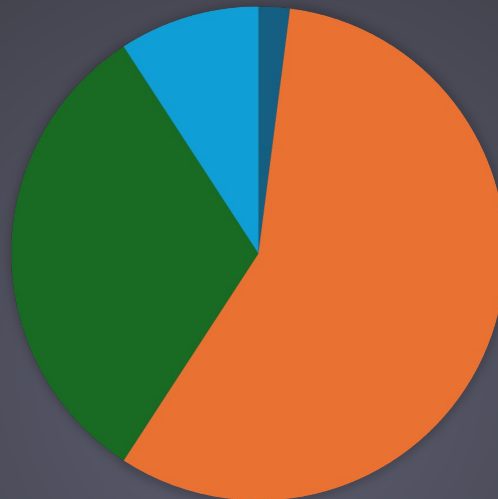


## PERFORMANCE ANALYSIS

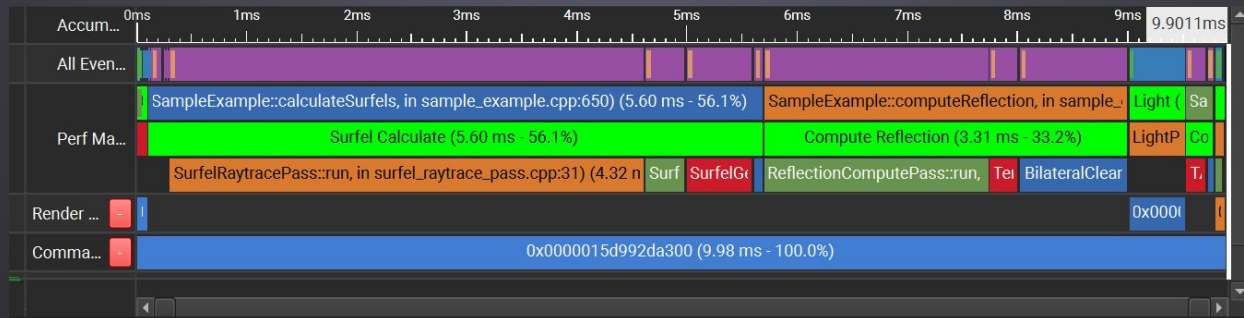
Integrate Stage 9%  
Prepare Stage 2%

Reflection  
Calculate  
32%

Surfel  
Calculation  
57%



About 5 ~ 10 ms  
per frame



# Conclusion

## Advantages:

- No precomputation
- GI in real-time
- Can combine with some advanced techniques like Restir & Stochastic light cut

(Possible future work)



## Limitation:

- Take some time to converge
- Only low frequency lighting



# Platform & Base Code

Platform: Tested on Windows, Nvidia GPU with hardware ray tracing support.

Graphics API: Vulkan 1.3 & GLSL

Base Code: [Nvidia VK\\_RAYTRACE Renderer](#)





Live Demo!



# Thanks for Watching!

A project by Zhen Ren, Ruipeng Wang and Jinxiang Wang



Penn  
Engineering  
UNIVERSITY of PENNSYLVANIA

# References

<https://www.ea.com/seed/news/siggraph21-global-illumination-surfels>

[https://advances.realtimerendering.com/s2024/content/EA-GIBS2/Apers\\_Advances-s2024\\_Shipping-Dynamic-GI.pdf](https://advances.realtimerendering.com/s2024/content/EA-GIBS2/Apers_Advances-s2024_Shipping-Dynamic-GI.pdf)

<https://media.contentapi.ea.com/content/dam/ea/seed/presentations/dd18-seed-raytracing-in-hybrid-real-time-rendering.pdf>

<https://github.com/W298/SurfelGI>

<https://www.ea.com/frostbite/news/stochastic-screen-space-reflections>

[https://old.reddit.com/r/opengl/comments/96api8/has\\_anyone\\_successfully\\_implemented\\_groundtruth\\_ao/e40d2ie/](https://old.reddit.com/r/opengl/comments/96api8/has_anyone_successfully_implemented_groundtruth_ao/e40d2ie/)

<https://intro-to-restir.cwyman.org/>



Zhen Ren

- [LinkedIn](#)
- [renzhen@seas.upenn.edu](mailto:renzhen@seas.upenn.edu)

Ruipeng Wang

- [ruipeng@sas.upenn.edu](mailto:ruipeng@sas.upenn.edu)
- [wang-ruipeng.github.io](https://github.com/wang-ruipeng)

Jinxiang Wang

- [LinkedIn](#)
- [andyjw99@seas.upenn.edu](mailto:andyjw99@seas.upenn.edu)

