




Zihong Zhou

 (+86)159-1331-2531

 hearwindsaying@gmail.com

 <https://hearwindsaying.github.io/>

Summary

I am strongly interested in computer graphics, especially in physically based rendering. I attempt to learn and understand the principles behind natural phenomena appearing in the physical world. I could then synthesize realistic images in a virtual world fabricated by math and code.

Education

2017.9 – 2021.6  **B.Sc. Computer Science and Technology, South China Agricultural University**
GPA: 4.14/5 or 91.44/100 (Rank: Top 5%)

Research Publications

- 1 Zhou, Z., & Wei, L.-Y. (2020). Spherical light integration over spherical caps via spherical harmonics. In *Siggraph asia 2020 technical communications*. SA '20.  doi:10.1145/3410700.3425427

Research

Analytical Area Light Integration via Spherical Harmonics 2020.2-2020.8

- Advised by Dr. Li-Yi Wei through his Open Mentor program.
- Studied state-of-the-art methods for efficient polygonal and spherical area light integration.
- Implemented two SIGGRAPH/ToG 2018 papers: Integrating Clipped Spherical Harmonics Expansions, Analytical Spherical Harmonics Coefficients for Polygonal Area Lights.
- Published my first SIGGRAPH short paper Spherical Light Integration over Spherical Caps via Spherical Harmonics.

Experience

Rendering Engineer (Intern), 4D ShoeTech 2020.9 - 2021.6

- Worked on in-house real-time renderer designed for footwear manufacturing. Implemented Eric Heitz's Linearly Transform Cosine method for real-time shading polygonal area lights.
- Reported and fixed several bugs of the offline renderer SDK.
- Integrated a brand-new DirectX 12 renderer backend to the existing RHI (Render Hardware Interface), being compatible to the current rendering pipeline and APIs.

Rendering Engineer (Full-Time), 4D ShoeTech 2021.6 - present

- Launched a new project on reference path tracer with high visual fidelity based on DirectX 12 backend and DXR (DirectX Ray Tracing) API.
- Integrated NVIDIA's Realtime Denoiser and Intel's OpenImageDenoiser to the post-processing pipeline of the path tracer.

Service

SIGGRAPH Research Career Development Committee

2021.2 - present

A committee member in undergraduate mentorship program, helped organized different tracks of programs which are adapted for students at various levels, ranging from introductory courses studying groups (arrival) to graduate school applications for computer graphics (departure).

Side Project

Colvillea: A Physically Based GPU Ray Tracer

2018.7-Now

This is a physically based global illumination renderer I wrote for learning computer graphics. I also use it to implement papers and experiment with new ideas during my research. It relies on Nvidia's OptiX to achieve parallelism by leveraging GPU resources, resulting in high performance ray tracing rendering.

Living Room in Unity

2019.11-2019.12

This is my final project for the Virtual Reality course. My contribution is bringing Benedikt Bitterli's famous rendering scene living room to Unity and showing how we could use various rasterization techniques to approximate the beautiful image produced by the offline renderer.

Simple Photoviewer

2019.3-2019.5

A simple photoviewer written for OOP course using C++/WinRT deployed at Universal Windows Platform. Standard C++17 and XAML language are used for the project. Several optimization techniques are employed to provide the user with a smooth interaction when previewing large image files.

Vortex Ring Art

2021.11-2021.12

Inspired by a post on the vortex simulation method in the movie industry, I re-implemented the author's 2D algorithm and extended it to 3D in Taichi, a high-performance domain-specific language (DSL) for simulation. Interestingly, I ran this script within Blender to control the particle system so that I could use Blender's power for rendering and composition.

Skills

| | |
|-----------|---|
| Coding | 📌 C/C++, C++/WinRT, CUDA, C#, XAML |
| Framework | 📌 OptiX, Diligent Engine, DirectX RayTracing, NSight, Mitsuba |
| Toolkit. | 📌 \LaTeX , Git, Mathematica, Maya, Blender, Unity |
| Language | 📌 Mandarin/Cantonese (Native), English (Fluent) |