Song Zhang

425-351-6164 song.zhang@utah.edu linkedin.com/in/song-zhang github.com/guiqi134

Technical Skills

Technologies: Computer Graphics, Real-time Rendering, Physically-based Rendering
Languages: C/C++, Python, C#, Javascript
Developer Tools: VS & VS Code, Unity, Git
Frameworks: OpenGL, Falcor, DirectX, CUDA

Education

University of Utah **Sep. 2020 – May 2022**
Master of Science in Computer Science (GPA: 3.8) *Salt Lake City, UT*

New York Institute of Technology **Sep. 2016 – May 2020**
Bachelor of Science in Computer Science *New York, NY*

Relevant Coursework

- Introduction to Computer Graphics
- Interactive Computer Graphics
- Math of Computer Graphics
- Parallel Programming Many-Core
- Visualization for Scientific Data
- Graduate Algorithms
- Virtual Reality
- Distributed Systems
- Computer Architecture

Projects

Using ReSTIR for Area Light Soft Shadows | *C++, Falcor, Slang* **Sep. 2021**

- Modified origin Percent-Closer Soft Shadows (PCSS) method by changing the final PCF step.
- Integrated this new PCSS method with ReSTIR for shadow approximation.
- Some comparison tests are being done with other shadow mapping methods like Variance Shadow Map (VSM), Exponential Shadow Map (ESM), and Moment Shadow Map (MSM).

Real-time Area Light Implementation with Linear Transformed Cosines | *C++, OpenGL* **Feb. 2021**

- Realized four types of area lights – rectangular, cylindrical, elliptic and ellipsoid using the techniques described in the LTC paper.
- Designed two scenes using ImGui and OpenGL:
  - In the first scene, users can adjust the coefficients of each light as well as the properties of plane material to see how those changes will affect the final rendering.
  - In the second scene, many dynamic area lights were randomly generated to show their interaction with the GGX material plane.

BRDF Visualization System | *C++, Falcor* **Feb. 2021**

- Using Falcor designed a real-time interactive BRDF visualization system, in which the BRDF Lobe was constructed using SDF with ray marching.
- The entire system implements UE4's BRDF material and provides a material ball to show the final render results.

GPU Ray Tracing | *C++, Cuda* **Feb. 2021**

- Using CUDA to transfer the computation of each Ray from CPU to GPU and achieving a 17.4x speedup in rendering.
- The final rendered image implements 3 spherical materials: Lambertian, Metal and Glass with a depth of field effect.

Unity Cross-platform Standalone VR basketball Game | *C#, Unity* **Sep. 2020**

- Implemented VR basketball game running on Vive Index, Oculus and Windows Mixed Reality using UnityXR API.
- Participated in the implementation of character movement, picking and throwing the ball.
- Built appropriate textures, materials, sound and physics models for each object in the scene.