

# Yansong Yu

E-mail: yu.4083@buckeyemail.osu.edu Cell: +16142847737

## EDUCATION

<b>The Ohio State University</b>	08/2024-present
Ph.D. student in Computer Science and Engineering	
<b>Nanjing Tech University</b>	09/2021-07/2024
Master of Science in Computer Science	
<b>GPA: 3.76/4.0</b>	
<b>Nanjing University of Information Science &amp; Technology</b>	09/2017-06/2021
Bachelor of Engineering in Computer Science and Technology	
<b>GPA: 3.73/4.0 (Grade Ranking: 2/264)</b>	

## PUBLICATION

Yansong Yu, Zhenlong Du, Xiaoli Li, *High Performance Nonlinear Least Square Optimization via Domain Specific Language*, 2023 15<sup>th</sup> International Conference on Graphics and Image Processing (ICGIP 2023), 2023.

Yansong Yu, Zhenlong Du, *A GPU Computation-based Ray Tracing Engine with User-friendly and Scalable Rendering Features and Structures*, APCT 2024 (2024 3rd Asia-Pacific Computer Technologies Conference)

## RESEARCH EXPERIENCE

<b>Huawei Mindspore Neural Network Framework Operator Development</b>	07/2021-11/2022
<ul style="list-style-type: none"><li>Worked on the development of SparseToSparseSetOperation, SparseMatrixTranspose, ExtractGlimpse, and SegmentMin</li><li>Used unit test with the Google Test framework (gtest) to test the functionality of operators and employed system test by constructing large input data to evaluate the performance of operators, and adopted the Unix naming methods to ensure code readability</li></ul>	
<b>Real-time ARAP (As-Rigid-As-Possible) Mesh Deformation based on the High-performance Least Squares Computation Library Opt</b>	04/2022-07/2022
<ul style="list-style-type: none"><li>Established the front-end and back-end interaction framework and developed the back-end ARAP mesh deformation algorithm</li><li>Implemented the CPU version code for the control group (based on the Eigen library), CUDA, and OpenGL interaction logic</li><li>Completed screen-space ray tracing and ray tracing acceleration structure BVH (Bounding Volume Hierarchy), and realized Matcap rendering for achieving better visual effects at a lower rendering cost</li></ul>	

## PROJECT EXPERIENCE

<b>Compute Shader-based Path Tracing Engine</b>	11/2022-Present
<ul style="list-style-type: none"><li>Used offline rendering techniques on the CPU to obtain the Ground Truth of the rendered results, and implemented data transfer between the CPU and compute shader using OpenGL's Uniform and Storage Buffer</li></ul>	
<b>More Advanced Ray Tracing Engine</b>	11/2021-04/2022
<ul style="list-style-type: none"><li>Implemented Lambertian, Phong, Blinn-Phong reflection models, Cook Torrance specular reflection model, Disney Principled BSDF, and their respective Importance Sampling Techniques on the Shader side, implemented Next Event Estimation, and further implemented Multiple Importance Sampling techniques to further reduce variance and coverage time in the scene</li></ul>	
<b>Online Canonical LR Parser Syntax Analysis Program</b>	02/2020-07/2020
<ul style="list-style-type: none"><li>Took charge of algorithm implementation, syntax tree generation, and error list generation</li><li>Used C++ to implement the algorithm for generating the Canonical LR parser analysis table and improved the control program, error list generation, and matching process display</li><li>Visualized the syntax tree using D3.js</li></ul>	
<b>Lexical Analysis Program</b>	02/2020-07/2020
<ul style="list-style-type: none"><li>Implemented logic for removing comments and checking for closed multiline comments</li><li>Realized the functions of lexical checks, character and character-to-escape sequence matching, and special symbol checks</li><li>Generated token pairs that can be used for further processing</li></ul>	
<b>Android Game Development</b>	02/2020-07/2020
<ul style="list-style-type: none"><li>Implemented periodic difficulty level upgrades based on the score</li><li>Drew and refreshed graphics by using the Canvas in Android</li><li>Utilized a multi-thread strategy and the Handle in the sub-thread to achieve real-time refreshing of the main thread UI</li></ul>	
<b>Regular Expression to Minimal DFA Conversion</b>	01/2020-02/2020

- Used C++ to write Thompson's construction algorithm to generate NFA from regular expressions and persisted it
- Combined with  $\epsilon$ -closure theory to determinize NFA using C++ and used the Breadth First Search (BFS) algorithm to minimize DFA
- Performed a Depth First Search (DFS) traversal on the generated DFA to obtain the corresponding DOT script and further generated a PDF

#### **Big Data Algorithm Platform based on Python**

09/2019-01/2020

- Wrote the Apriori algorithm with Python, implemented the FP-Growth algorithm, and visualized the constructed results using Matplotlib's TreePlotter
- Used the sklearn toolkit to implement k-means, k-medoids, and FP-Growth algorithms, and utilized SVM classification to implement online handwritten digit recognition
- Compared the efficiency of the Apriori algorithm, Eclat algorithm, and FP-Growth algorithm on the Growerries dataset with different data sizes

#### **Image Processing System Design based on MATLAB**

02/2019-07/2019

- Used MATLAB to write threshold methods, Otsu algorithm, and other sharpening and filtering algorithms for image binarization and implemented spatial transformations for images (rotation, translation, skew, scaling) and adaptive smoothing filters
- Compared performance differences among various filtering algorithms and implemented polynomial image wrapping functionality

### **PATENT**

Opt-based Method for Robust and Efficient ARAP Deformation (publication number: CN116012548A)

### **SOFTWARE COPYRIGHT**

Computer Software Copyright Registration Certificate for the Software: Illumengine Rendering System V1.0 based on OpenGL, 2021 (registration number: 2021SR1016796)

### **GRADUATION DESIGN**

#### **Three-dimensional Rendering System**

2020-2021

- Imported obj format models and completed the basic transformations of objects (translation, scaling, rotation)
- Improved the shadow quality in open scenes using CSMs algorithm based on the implementation of real-time shadows using depth map techniques
- Implemented the Omnidirectional Shadow Maps technique for point light sources and realized soft shadows using the PCF method
- Visualized the Frustum bounding box of the main camera by implementing a debug camera
- Supported the functions of editing texture maps such as diffuse maps, specular maps, and ambient maps for model meshes
- Realized various visual effects (HDR, Blooming) and basic functionalities (normal visualization, normal flipping, etc.) for enhancing the visuals

### **AWARD AND HONORS**

Special Class Scholarship, Nanjing Tech University	2022
Provincial First Prize & National Second Prize in the 11 <sup>th</sup> C/C++ Programming Contest	2020
First-class Scholarship & Merit Student, Nanjing University of Information Science & Technology	2020
First-class Scholarship, Nanjing University of Information Science & Technology	2018
Merit Student Scholarship, Nanjing University of Information Science & Technology	2018
Third Prize in Programming Contest, Nanjing University of Information Science & Technology	2018
Excellent Prize in Mathematical Modeling of Nanjing University of Information Science & Technology	2018
Third Prize in Jiangsu Provincial Mathematical Contest	2017
Third Prize in Mathematical Contest, Nanjing University of Information Science & Technology	2017

### **SKILLS & OTHERS**

**Languages:** Chinese (native), English (proficient), Japanese (preliminary)

**Programming Languages:** Proficient in C/C++, Python, Java, MATLAB, familiar with shell language

**Computer Tools:** using MATLAB to write image processing algorithms and visual programs, and using matplotlib library in Python for visual programming

**Other Certificates:** PAT Level B Full Score Certificate, Jiangsu Provincial Computer Level 3 Certificate, National Computer Level 2 Certificate