Yuanbo Li

646-821-2178 Email:yuanbo_li@brown.edu
Github: https://github.com/Liyb2002 Website: https://liyb2002.github.io/

Education

Brown University **(Sep 2022 - May 2024)**
M.S in Computer Science (Visual Computing Track)
Relevant Courses: Computer Graphics, Advanced Computer Graphics, Deep Learning

Columbia University, Columbia College **(Sep 2018 - May 2022)**
B.A in Mathematics
Relevant Courses: Advanced Programming, Natural Language Processing, Cloud Computing, Analysis, Abstract Algebra

Research Interests

Computer Graphics, Geometry Processing, NeuralSymbolic Methods, Simulation, AR/VR, Rendering

Publications

PossibleImpossibles: Exploratory Procedural Design of Impossible Structures, (under review)
Eurographics 2024, Full Paper, **Yuanbo Li**, Tianyi Ma, Zaineb Aljumayaat, Daniel Ritchie

Research Experience

Visual Computing Group, Brown University **(Sep 2022 - Sep 2023)**
TL;DR: We designed an exploratory system to generate 3D structures that appear to be impossible.
Advisor: Prof. Daniel Ritchie

- Explored taxonomy of impossible structures, introduced a procedural language, and designed a procedural model to generate the structures
- Designed scoring functions for characterizing visually pleasing impossible structures, and applied Sequential Monte Carlo to guide the search for output space of the result

Visual Computing Group, Brown University **(Oct 2023 - NOW)**
TL;DR: We proposed a novel method to infer 3D impossible structures from 2D images
Advisor: Prof. Daniel Ritchie

- Used neural networks to predict depth maps for impossible structures
- Extract rules regarding connectivity of the structure in 3D space, and use neurally guided Sequential Monte Carlo to reconstruct the impossible structure in 3D space.

Visual Computing Group, Brown University. **(Feb 2023 - NOW)**
TL;DR: We proposed methods to find color assignments to make labels visually salient in an AR setting.
Advisor: Prof. James Tompkin

- Design algorithms to calculate and interpolate pixel values based on background image. Implemented the algorithms using Unity shader, compute shader, and thread dispatching
- Designed and implemented parts of a neural network to predict label colors, including masking, optimization, loss functions, and palette color choosing.

Collaborative Prediction Market Lab, Columbia University **(Sep 2021 - Jan 2022)**
TL;DR: We applied blockchain network to record market predictions.
Advisor: Prof. Siddhartha Dalal

- Built App backend using Django framework and REST API. Deployed service on AWS EC2. Retrieved on-chain data using web3js for analysis.
- Implemented AMM algorithm to reduce the slippery rate by 15% each transaction.

Work Experience

ARPA Technology **(May 2022 - Aug 2022) /   (June 2020 - Aug 2020)**

Software Engineer Intern

- Designed and implemented Behavior Driven Tests for Randcast, a distributed random number generating system for Blockchain Networks, and wrote documentations for the system
- Designed smart contract for Bella.fi, a DeFi protocol, based on Open Zeppelin contract libraries

Tencent CSIG **(May 2021 - Aug 2021)**

Technical Product Manager Intern

- Participated in architecture design for blockchain-based IPFS (InterPlanetary File System).
- Wrote 10+ pages documentation and 30+ page whitepaper to help build on IPFS system.

Tencent Videos **(June 2019 - Aug 2019)**

Product Manager Intern

- Used python to write web-crawler for 4 websites and gather view information for a total of 100+ articles on Tencent videos
- Used python (data cleansing) to analyze performance of different 100+ Key Opinion Leader

Teaching

Advanced Graphics, TA, (prof. Daniel Ritchie) , Brown Computer Science Department, Spring 2024
Calculus III, TA, (prof. Daniele Alessandrini) , Columbia Mathematics Department, Fall 2021

Selected Projects

Path Tracer (Graphics) [C++] Implemented a path tracer with 4 basic types of BRDFs, soft shadows, Russian Roulette path termination and event splitting with BRDF importance sampling.
Mesh Processing (Graphics) [C++] Implemented mesh subdivision, simplification, denoising, and remeshing.
Finite Element Simulation (Graphics) [C++] Implemented finite element simulation with internal elastic and viscous damping forces, collision detection, and RK4 integration.
As-Rigid-As-Possible [C++] Dissected and implemented paper “As-Rigid-As-Possible Surface Modeling”(2007).
Stylized Caustics (Graphics) [C++] Dissected and implemented paper “Stylized caustics: progressive rendering of animated caustics”(2016).
Impressionism Filter (Graphics) [C++] Dissected and implemented paper “Processing images and video for an impressionist effect”(1997).
PointNet (Computer Vision) [Python] Dissected and implemented paper “PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation” (2016)
Formal Proof of L-Systems (Programming Languages) [Lean] Used Lean4 (a formal proof language) to define L-Sytems and make proofs on output types and terminations.

Talks

Inverse Procedural Modeling, Brown Visual Computing Group, Nov 2023
Generating 3D impossible structures via guided Sequential Monte Carlo, Brown Visual Computing Group, Feb 2023 / Nov 2023
Mathematics behind M.C Escher’s Tessellations, Columbia Mathematics Seminar, Feb 2022
Introduction to Markov chain basics and applications, Columbia Mathematics Seminar, Oct 2021

Skills

Programming Languages: C++, Python, Javascript, C#, Java, Lean4
Libraries/Frameworks: OpenGL, Tensorflow, PyTorch, Unity, threeJS, Pyrender, Django
Tools: Blender, Adobe Illustrator, Adobe Photoshop, Keyshot, Mesh Lab, AWS, Final Cut Pro