Joseph R. Morrissey

jrm15@illinois.edu ▪ 847-744-0107 ▪ https://jmorrissey23.github.io/Personal-Website/ ▪Park Ridge, Illinois

EDUCATION

University of Illinois Urbana-Champaign *Cumulative GPA:* ***3.89/4.00***
Bachelor of Science in Computer Science *Expected: May 2025*

Honors: Edmund J. James Scholar, Dean's List

Languages: Python, C/C++, SQL, Lua, HTML, CSS

Relevant Coursework

- Systems Programming, High Frequency Trading, Distributed Systems, Data Structures, Algorithms, Database Systems, Deep Learning, Computer Architecture

WORK EXPERIENCE

Novaspect Inc. Schaumburg, IL
Software Engineer Intern *May 2024 – August 2024*

- Developed a comprehensive model of a brewing process using C#, SQL, CSS, and TrakSYS, minimizing operator errors by +20% and reducing labor by 2+ hours for each job
- Created a robust inventory tracking system using C# , SQL, and TrakSYS, saving the client thousands of dollars by ensuring timely orders and preventing unnecessary purchases
- Accomplished effective project management by participating in daily stand-ups and utilizing Jira and Agile methodologies, resulting in improved team collaboration and an increase in on-time project delivery

Software Engineer Intern *May 2023 – August 2023*

- Developed a historian synchronization microservice for an industrial automation software platform using TCP/UDP data pipes and Lua socket programming, resulting in a $5,000 client proof of concept
- Expanded over 100 endpoints for a remote debugging API, enabling low latency integration with platform proprietary libraries, faster error detection and troubleshooting, and an overall smoother development experience
- Integrated an internal user story development interface with a microservice architecture platform using Lua and React JSON Schema to optimize Agile workflow and streamline client communication

PROJECTS

Market Ticker Plant

- Architected a homemade ticker plant capable of listening for specific stock symbols and parsing millions of bytes of market data
- Measured the latency of the ticker plant using a network switch and three raspberry pi's – one to tcpreplay a pcap file, one as the ticker plant, and one to find the time difference between receiving the market data and the ticker plant response
- Reduced the average latency by 0.1 seconds through various raspberry pi tuning methods, which also resulted in a tighter spread of values

Painting without Paint

- Implemented *Image Analogies* (Hertzmann et al.), which takes an unfiltered-filtered picture pair and an unfiltered image as input, and returns an output image that completes the analogy
- Generated feature vectors over the gaussian pyramids of the luminance channels of each image. Compared an approximate nearest neighbor search using a KDtree with a pixel that minimizes an objective function to find the best matching pixel for the output image
- Combined the Image Analogy method with two other group members implementation of *Color transfer between images* (Reinhard et al.), allowing us to produce images that mimic an artists' style

ORGANIZATIONS

Association for Computing Machinery **Phi Gamma Delta Fraternity**
Member (September 2021 – Present) *Member (February 2022 – Present)*