**Steven (Guanqi) Yu**

<https://www.linkedin.com/in/yuguanqi>

<https://github.com/StephenGuanqi>

**EDUCATION**

**Carnegie Mellon University**, Pittsburgh, PA Graduated in May 2018

Master of Information Systems Management

***Courses:*** *Operating Systems, Parallel Computing, Distributed System, Cloud Computing, Practical Data Science, Computer Vision, Introduction to Machine Learning*

**Nanjing University of Information Science & Technology**, Nanjing, China Graduated in June 2016

Bachelor of Engineering degree in Information Systems Engineering

**SKILLS**

Solid **C++11**& **C** programming experience on embedded **SOC**, strong in **Linux** **multithreading & multi-process** programming and **Debugging**. Solid in Operating System architecture, experienced in **Unix system** programming, **CUDA** GPU programming and shell & python. Good math foundation in computer vision & machine learning, familiar with Caffe, Tensorflow and AWS.

**PROFESSIONAL EXPERIENCES**

**NVIDIA**, Santa Clara, California, United States July 2018 – Present

System Software Engineer, DriveWorks SDK(middleware), Autonomous Vehicle

* Board bring-up for NVIDIA DRIVE AGX Pegasus platform, contribute to multiple modules including image interop, sensor driver and ROS-like framework. Optimized multiple APIs’ performance for low latency. Top Contributor on 1st year.
* Developed a tool that can share the camera sensor images across multiple processes in a server-slave mode through IPC. Collaborate with multiple teams in application stack to integrate the feature, and successfully brought up for real-time in-car launching. Tuning server to achieve %4 cpu usage. --- C++11, multi-thread strategies, IPC, ftrace debugging, socket
* In charge of x86 development of NvSciBuf, which is a safe central allocator which takes into account HW constraints from different engines and allocates a buffer which can be accessed by all engines involved in interop. Used by multiple User Mode Drivers like NvMedia and CUDA. ---C, safety-standards, Makefile, cross-platform, design patterns
* Contribute to channel communication for nodes in our own ROS-like framework implementation. Utilized NvSci API to provide safe and efficient C data structures & C++ objects transport across different threads & processes. --- C++11, template programming, pointer arithmetic

**Wondergate Augmented Reality**, Beijing, China May 2017 – Aug 2017

Software Engineer Intern

* Developed a Gradle Build System that can cross compile C++ code into shared library for Android, iOS and Windows platform simultaneously, used by entire developers in Company’s R&D team.
* Developed iOS Augmented Reality App in ARKit and open source ARKit-Unity plugins on Github.

**National Laboratory of Pattern Recognition (Chinese Academy of Sciences)**, Beijing, China Aug 2015 – July 2016

Software Engineer Intern

* Built an Android OCR App for Taizhou Water Company to enable digital, real-time recording of water meter numbers to reduce the human error and redundancy of manual records.
* Implemented QR-code-scanning-like Android JNI App, designed C++ OpenCV image segmentation algorithm and utilized Deep Neural Net on Caffe to recognize water meter digits. Cross-compiled C++ modules on Android using NDK.

**ACADEMIC PROJECTS**

**Caching Proxy Server & Dynamic Storage Allocator in C, CMU** Sep 2017 – Dec 2017

* Built a HTTP Proxy Server in C to handle Concurrent HTTP requests, implemented LRU cache to improve efficiency.
* Implemented C standard library malloc, free, realloc and calloc. Achieve the final 74.3% space utilization and 22000 kilo-operation per second throughput. Debugging in GDB on x64 assembly code.

**Planar Tracking Augmented Reality Android App, CMU** Nov 2017 – Dec 2017

* Built planar tracking algorithm in C++11 based on Homography estimation and deployed on Android device.
* Used Kanade-Lucas feature tracking in OpenCV to calculate optical flow, and use ORB feature descriptor for relocalization. Achieve 30 frame per second tracking on 1080p resolution camera preview.

**Selected Computer Vision Course Projects in MATLAB, CMU** Sep 2017 – Dec 2017

* Bag-of-words with spatial pyramid matching in MATLAB to conduct scene classification.
* Implemented SIFT+BRIEF feature matching to stich panorama, as well 3d-reconstruction pipeline.
* Implemented Lucas-Kanade with Inverse Composition on affine transformation to track motion with fair speed.