Yunpeng Xu

yunpengx@andrew.cmu.edu
(+1) 412-251-9546
http://yunpengx.me

EDUCATION

Carnegie Mellon University, School of Computer Science Pittsburgh, PA
Master of Science in Embedded Software Engineering *Sep. 2016 - Dec. 2017*

- Real-Time Embedded System, Parallel Computer Architecture and Programming, Introduction to Embedded System, Introduction to Computer Systems, Architectures for Software Systems, Mobile Application Development

University of Science and Technology of China Hefei, China
Master of Biomedical Engineering, Medical Device, GPA: 86.03/100 *Sep. 2011 - Jul. 2014*
Bachelor of Electronic Information Science and Technology, GPA: 86.42/100 *Sep. 2007 - Jul. 2011*

EXPERIENCE

Research Assistant, Carnegie Mellon University, Pittsburgh, PA May. 2017 - Present

- Built an OpenCV-based moving object detection application with mono fisheye camera on Nvidia TX1 platform.
- Evaluated application performance, improved each frame's processing time by 23.5% after optimization using CUDA.

System Application Engineer, Ambarella Inc., Shanghai, China Oct. 2015 - Jul. 2016

- Designed and implemented a smart rate control library for Ambarella's S2L and S3L SDK, supported AVC/HEVC.
- Efficiently improved video compression ratio while maintaining video quality, later ported to Apple's HomeKit service.
- Designed and implemented Netlink module to transfer messages between kernel and user-space process.

Embedded Software Engineer, Galaxycore Inc., Shanghai, China Jul. 2014 - Sep. 2015

- Core Linux device driver developers for Galaxycore's video surveillance SOC.
- Implemented device driver for digital imaging sensors, and AVC and JPEG decoding modules using V4L2 framework.
- Optimized device driver for image signal processing (ISP) and AVC encoding modules based on V4L2 framework.

PROJECTS

ROS Reconfiguration Framework (C++, ROS) Jan. 2017 - Aug. 2017
Software Engineering Practicum *Carnegie Mellon University*

- Led the team to design a reconfiguration framework that allows users (ROS application developers) to swap navigation and control algorithms and parameters being used in the robot at runtime.
- Created a model that depicts the designed framework protocol and verified it using Promela and LTL property.
- Implemented all core features (reconfigure framework, node dependency) individually and performed peer reviews.

Real-Time Task Reservation Framework in Linux Kernel (C, Linux Kernel/Android) Sep. 2016 - Nov. 2016
Course project for Real-Time Embedded System *Carnegie Mellon University*

- Designed and developed a task reservation and enforcement framework including kernel modules, system calls, sysfs interface and Android NDK for task admission control on Nexus 7 tablet.
- Implemented fixed-priority processor scheduling algorithms for multi-processors using task partitioning heuristics.
- Managed energy consumption of real-time tasks by implementing power management algorithms.

Parallel Galaxy Evolution Simulator (C/C++, CUDA) Apr. 2017 - May. 2017
Course project for Parallel Computer and Architecture Programming *Carnegie Mellon University*

- Implemented a galaxy evolution simulator using both Barnes-Hut algorithm and Morton-Code algorithm.
- Both algorithms achieved more than 10x speedup by using performance bottleneck analysis and CUDA acceleration.

Easy Order (Java/Python, Android/Django) Jul. 2017 - Aug. 2017
Course project for Mobile Application Development *Carnegie Mellon University*

- Led the team to design and develop a cross-platform meal delivery application for Chinese takeout, integrated with meal ordering, meal tracking and payment features.
- Built all user interfaces using Android layouts, implemented core features and interaction with RESTful APIs.

Dynamic Storage Allocator (C) Jun. 2016 - Jul. 2016
Course project for Introduction to Computer Systems *Carnegie Mellon University*

- Implemented a Dynamic Storage Allocator including malloc, free, realloc and calloc interfaces.
- Compared three free blocks organization strategies' performance: implicit free list, explicit free list, segregated free list.
- Achieved 78% memory utilization over 29 cases using segregated list, first fit, splitting and coalescing after block freed.

SKILLS

Programming Languages: C/C++, Java > Python > Javascript > Bash, Matlab
Linux Development: Device driver, Kernel module, Task reservation & power-aware algorithms, Video Codec
Frameworks: ROS, CUDA, Android, V4L2, OpenCV, OpenMP, MPI, ISPC, Django, Bootstrap, React
Tools: Git, Makefile, GDB, Repo, Scrum, JIRA, Markdown