|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ZHENYU YANG** | | | | | | | | |
|  |  | | | | |  | | |
| Goleta, CA 93117 | +1 (312) 888-0068 | | | | | zhenyuyang@umail.ucsb.edu | | |
|  | | | | | | | | |
| **Education** | | | | | | | | |
| University of California, Santa Barbara (UCSB), Santa Barbara | | | | Sep 2015 - Expected: June 2018 | | | | |
| M.S., Electrical and Computer Engineering | | | | GPA: 3.86/4.0 | | | | |
| Advisor: Kwang-Ting (Tim) Cheng | | | |  | | | | |
|  | | | |  | | | | |
| Illinois Institute of Technology (IIT), Chicago | | | | Aug 2013 - May 2015 | | | | |
| B.S., Electrical Engineering | | | | GPA: 3.92/4.0 | | | | |
|  | | | | | | | | |
| North China Electric Power University (NCEPU), Beijing | | | | Sep 2011 - July 2013 | | | | |
| B.S., Electrical Engineering (Transferred to IIT to complete) | | | | GPA: 3.71/4.0 | | | | |
|  | | | | | | | | |
| **Internships** | | | | | | | | |
| **Continental Advanced Lidar Solutions, Carpinteria, CA** | | | ***Algorithm Intern*** | | | | | **June 2016 – May 2017** |
| * *Project: Street scene real-time reconstruction in Autonomous Driving(C++ & Matlab)* | | | | | | | | |
| * Implemented a real-time point cloud viewer to render a street scene filmed by 4 Lidar cameras | | | | | | | | |
| * Simulated a virtual Lidar camera to investigate the effect of rain drops on UV rays | | | | | | | | |
| * Implemented Fast Point Feature Histogram(FPFH) algorithm and tested with raw point cloud data | | | | | | | | |
| * Interfaced Matlab core to pthread programs to boost computation of FPFH by 37% | | | | | | | | |
|  | | | | | | | | |
| **Projects** | | | | | | |  | |
| **Drone Cinematography, UCSB** | | | | | | | **March 2016 – November 2017** | |
| * *An autonomous drone flight program to track objects and achieve desired camera frame composition(C++)* | | | | | | | | |
| * Devised a vision-based feedback control system with inverse Jacobian control scheme | | | | | | | | |
| * Simulated the proposed tracking system in Unity and Gazebo | | | | | | | | |
| * Implemented the vision tracking system in Linux ROS with DJI Onboard SDK | | | | | | | | |
| * Interfaced the tracking system between a drone and an Android application | | | | | | | | |
| * Co-authored a paper “An Autonomous Drone Cinematography System for Action Scenes” accepted by *IEEE International Conference on Robotics and Automation* 2018 | | | | | | | | |
|  | | | | | | | | |
| **Paper Town - Android Application, UCSB** | | | | | | | **January 2017 – May 2017** | |
| * *A location-based story sharing android application(Java)* | | | | | | | | |
| * Implemented an user location management module and tested it with GPS emulator | | | | | | | | |
| * Worked with the UX team of 2 people to fulfill UI designs of sliding card views and map views | | | | | | | | |
| * Optimized an interactive google map module, decreased the location loading time by 20% | | | | | | | | |
|  | | | | | | | | |
| **Android-based AR Navigation, UCSB** | | | | | | | **January 2016 – March 2016** | |
| * *An augmented reality application for street navigation with Vuforia AR library(Java)* | | | | | | | | |
| * Utilized motion sensors, compass and GPS to obtain the spatial location of the phone | | | | | | | | |
| * Built a real-time decoder to process path-points JSON data received from Google Maps APIs | | | | | | | | |
| * Restructured and simplified the app navigation hierarchy based on feedback from 62 users | | | | | | | | |
|  | | | | | | | | |
|  | | | | | | | | |
| **LEGO Mindstorms Projects, UCSB** | | | | | **July 2015 – December 2015** | | | |
| * *An experimental project to implement multiple control schemes on mobile robots(Matlab)* | | | | | | | | |
| * Built an LQR control system to balance and stabilize a two-wheel robot | | | | | | | | |
| * Constructed an inverse Jacobian matrix from the state-space representation of an omni-wheel robot and used it to drive the robot to follow pre-defined trajectories | | | | | | | | |
| * Devised a transpose Jacobian control scheme to control the end-effector of a two-joint arm robot | | | | | | | | |
|  | | | | | | | | |
| **Skills** | |  | | | | | | |
| **Programming Languages:** | | C++, Java, Python, Matlab, C# | | | | | | |
| **Tools:** | | Visual Studio, Matlab, ROS, vim, git, Android Studio, Unity | | | | | | |