Dingding Zheng

Penn Engineering Research & Collaboration Hub (PERCH)
3401 Grays Ferry Ave, Philadelphia, PA 19146
(445) 888 1792
GitHub

Education

Aug 2018– May 2020 **Master of Science, Robotics**, *University of Pennsylvania*, United States.
Advisor: Prof. Dan Koditscheck & Prof. Mark Yim
Coursework: Advanced Robotics, Autonomous Racing, Linear system, Model Predictive Control, Machine Perception, Deep Reinforcement Learning, Convex Optimization

Sep 2014 – June 2018 **Bachelor of Engineering, EE**, *Donghua University*, China.
Advisor: Prof. Fang Han

Research

Interests

Multi-agent Systems, Autonomous UAV, Human Robot Collaboration, Dynamics and Control, SLAM.

Experience

Jun 2019 – Now **Graduate Research Assistant**, GRASP ModLab, University of Pennsylvania.
Topic: Reconfiguration of Multi-modular Robots (SMOREs)
Advisor: Prof. Mark Yim
- Research on 3D reconfiguration efficiency of modular robot.
- Initialized multi-camera detection system using "Vicon" and "Apriltag".
- Used ROS tuw pacakage and implemented a MPC controller to do multi-robot path planing and tracking.
- Skills Used: ROS, Pytorch, C++, Vicon Motion Capture System.

Aug 2019 – Oct 2019 **Graduate Research Assistant**, PRECISE Lab, University of Pennsylvania.
Topic: Human Robot Interaction & Safety Guarantee for Multi-agent System
Advisor: Prof. Osbert Bastani
- Added car dynamics model into OpenAI multi-agent particle environment.
- Implemented MPC controller to simulate the human decision making and trained robot car using MADDPG algorithm.
- Implemented "Human Social Force Model" to simulate the human decision making and MPC controller to control robot car.
- Stills Used: Pytorch, Tensorflow, OpenAI.

Projects

Spring 2020 **Advanced Robotics (MEAM 620)**, University of Pennsylvania.
Grade: A, Instructor: Prof. Vijay Kumar & Dr. James Paulos & Prof. CJ. Taylor

- Designed a geometric non-linear PID controller to let the quadrotor reaches desired goal without collision.
- Down-sampled the path derived from A* to get waypoints and implemented "Minimum Jerk" algorithm to get optimal trajectory.
- Implemented "Complementary Filter" and RANSAC to get accurate estimated states.
- Implemented "Error-state Kalman Filter" to estimate the pose of quadrotor given by data from IMU and onboard stereo pair.
- Reconstructed a 3D environment model given data from EuRoc.
- Used: Python, ROS, CrazyFlie 2.0, Vicon Motion Capture System, EuRoc dataset.

Spring 2020 **Autonomous Racing (ESE 615)**, University of Pennsylvania.
Grade: A, Instructor: Prof. Rahul Mangharam

- Designed and implemented algorithms to let the "F1tenth" racing car finish loops as soon as possible. Competed with cars from instructor and other teams.
- Implemented "Time-to-Collision" algorithm to ensure emergency self-braking.
- Implemented "Wall Following" algorithm to let the car keep a fixed distance to the wall.
- Implemented "Reactive Method: Follow the Gap" algorithm for obstacle avoidance.
- Implemented "Point-to-Line Iterative Closest Point (PLICP)" to estimate the pose of racing car given by data from IMU, Lidar and VESC.
- Generated a 2D map of Upenn Levine 2nd floor using "Google Cartographer".
- Implemented "Pure Pursuit" algorithm to let the car follow the trajectory.
- Implemented "RRT*" algorithm and created local occupancy map to let the car avoid obstacles more efficiently.
- Implemented "Minimum Curvature" and "Covariance Matrix Adaptation Evolution Strategy (CMA-ES)" algorithm to get optimal racing trajectory.
- Implemented "Obstacle-Dependent Gaussian Potential Field" algorithm to do obstacle avoidance.
- Set up several ROS environments to test the performance of different algorithms.
- Used: Python, C++, ROS, F1tenth Racing Car, Lidar, Google Cartographer.

Fall 2019 **Vison & Learning (CIS 680)**, University of Pennsylvania.
Grade: B+, Instructor: Prof. Jianbo Shi

- Generated adversarial images using deep neural network.
- Implemented "YOLO v1" to do extremely fast real time multi object detection.
- Implemented "Mask-RCNN", which combines object detection and semantic segementation into a per-pixel object detection.
- Implemented a family of generative models including: "Variational Autoencoder (VAE)" and "Generative Adversarial Network (GAN)".
- Implemented "Clipped Proximal Policy Optimization (Clipped PPO)" algorithm and applied it in OpenAI racing-car v0.
- Used: Python, OpenAI gym, Pytorch.

Fall 2019 **Deep Reinforcement Learning (ESE 680)**, University of Pennsylvania.
Grade: A, Instructor: Dr. Santiago Paternain & Dr. Miguel Calvo-Fullana

- Implemented Policy Gradient method and added Baselines.
- Implemented Actor-Critic algorithms, eg. Temporal Difference Learning.
- Implemented Actor-Critic method and applied it in OpenAI gym Acrobot v1.
- Used: Python, OpenAI gym, Pytorch.

Spring 2019 **Machine Perception (CIS 580)**, University of Pennsylvania.
Grade: A-, Instructor: Prof. Kostas Daniilidis

- Implemented projective geometry and homographies method to project "Upenn Engineering" logo onto the goal during football match and made a video.
- Reconstructed a 3D model from two 2D images using RANSAC.
- Implemented "DoG Filter" to do scale invariant detection.
- Used: MATLAB, Python, Pytorch.

Awards

2016-2017 **Excellent Academic Performance (Top 10%)**, Information Science Department, Donghua University.

2015 **Outstanding Undergraduate Student (Top 3%)**, Information Science Department, Donghua University.

Skills

Programming Languages: Python, C/C++, MATLAB
Tools: LaTeX, ROS, Ubuntu, Vicon Motion Capture System, Windows, Mathematica, Lingo
Deep Learning Frameworks: Pytorch, Tensorflow
Packages: Numpy, Cvxopt, Scipy, OpenCV, Matplotlib, Pandas, etc.