

Matt Buckley

hire@matt.software | +1 (504) 400-8024

EDUCATION

University of Southern California | Los Angeles, CA

August 2014 – May 2018

B.S. in Computer Science, Minor in Neuroscience | Viterbi School of Engineering

GPA: 3.3 | ACT: 36 | SAT: 2370

SKILLS AND INTERESTS

8+ Years Programming Experience

- **Proficient Languages:** C++, Python, C, Bash, Javascript, Java, Lua
- **Familiar Languages:** HTML/CSS, Lisp, MATLAB, PHP
- **Software Experience:** ROS, TensorFlow, Gazebo, Unix/Linux, Google Cloud, Git, React, V-Rep, Unity

RELEVANT COURSEWORK

Artificial Intelligence	Cognitive Neuroscience	Cryptography	Linear Algebra	Neurobiology
Operating Systems	Probability Theory	Professional C++	Software Engineering	Theory of Computation

EXPERIENCE

USC Center for Advanced Manufacturing | *Undergraduate Researcher* *January 2017 – September 2017*

- Led a research team to create the winning entry for the 2017 NIST ARIAC competition.
- Personally implemented the competition solution in C++, and integrated it with ROS and Gazebo.
- Presented the solution and findings for a NIST workshop at iROS 2017 in Vancouver.

USC Robotic Embedded Systems Lab | *Undergraduate Researcher* *November 2015 – August 2016*

- Helped develop an “active articulation” model to determine robot grasp joint geometry using ROS and C++.
- Worked to develop a firmware-level quadrotor swarm simulator in V-Rep using C++, C, Lua, and ROS.
- Helped improve quadrotor tracking software that processes data from a Vicon system, using C++ and MATLAB.

USC Aerial Robotics Team Leadership | *High-Level Software Team Leader* *August 2015 – May 2017*

- Managed a software team working on dynamic path planning, object detection and analysis, global state simulation, optical flow tracking, and more. Directed the design of the system layout for handling and connecting these components.
- Oversaw the integration of these systems with one another and with the robot’s control routines using ROS.

PROJECTS

ARIAC Competition *January 2017 – July 2017*

- Led a research team that designed and submitted the winning solution to the NIST ARIAC Competition. The competition was designed to spur advancement in Agile robotics, enabling industrial robots to dynamically plan and execute different kinds of tasks depending on their environment and sensor quality, rather than being single-purpose.
- Personally implemented the solution in C++, integrating it with the Gazebo simulator and ROS operating system.
- Invited to speak at iROS 2017 in Vancouver as part of winning, to present the strategy and methods developed.

Stochastic Robot Simulation *August 2014 – May 2017*

- Created and updated an event-based simulation for a system of interacting robots with added noise using C++, to model the possible outcomes of different given configurations. This was done to allow dynamic prediction of robots’ locations and states at a given time using limited information, for the International Aerial Robotics Competition.

Aerial Robot Path Planning *August 2015 – January 2018*

- Expanded upon the earlier event-based simulation by using it for intelligent planning. This planner, written in C++, estimates the outcome of sets of potential action sequences in order to pick initial moves with highest utility.

Quadrotor Swarm Simulation *June 2016 – August 2016*

- Worked to develop a simulation of a quadrotor swarm to allow rapid and safe testing of new control code. This involved integrating the drones’ C hardware code with C++ ROS and V-rep interfaces, as well as building an accurate physics model to realistically simulate their behavior.

Active Articulation Model *November 2015 – May 2016*

- Worked to expand an interactive model implemented in Python and C++ that determines the joint geometry connecting different objects by repeatedly finding and attempting the most “informative” manipulations with a robotic arm.

SkinCare *November 2015*

- Worked to develop an application that can recognize skin disorders from pictures using optical Machine Learning from the TensorFlow library to help the user self-diagnose different conditions, for HackSC 2015.