Eyes of Argus

M. Kaan Tasbas (CE/CS), Tri Nguyen (CS), Vinh Huynh (CS), Laura Gonzalez (EE)

Professor: Hanqi Zhuang

College of Engineering & Computer Science
Florida Atlantic University

EGN 4952C Engineering Design 2

Introduction

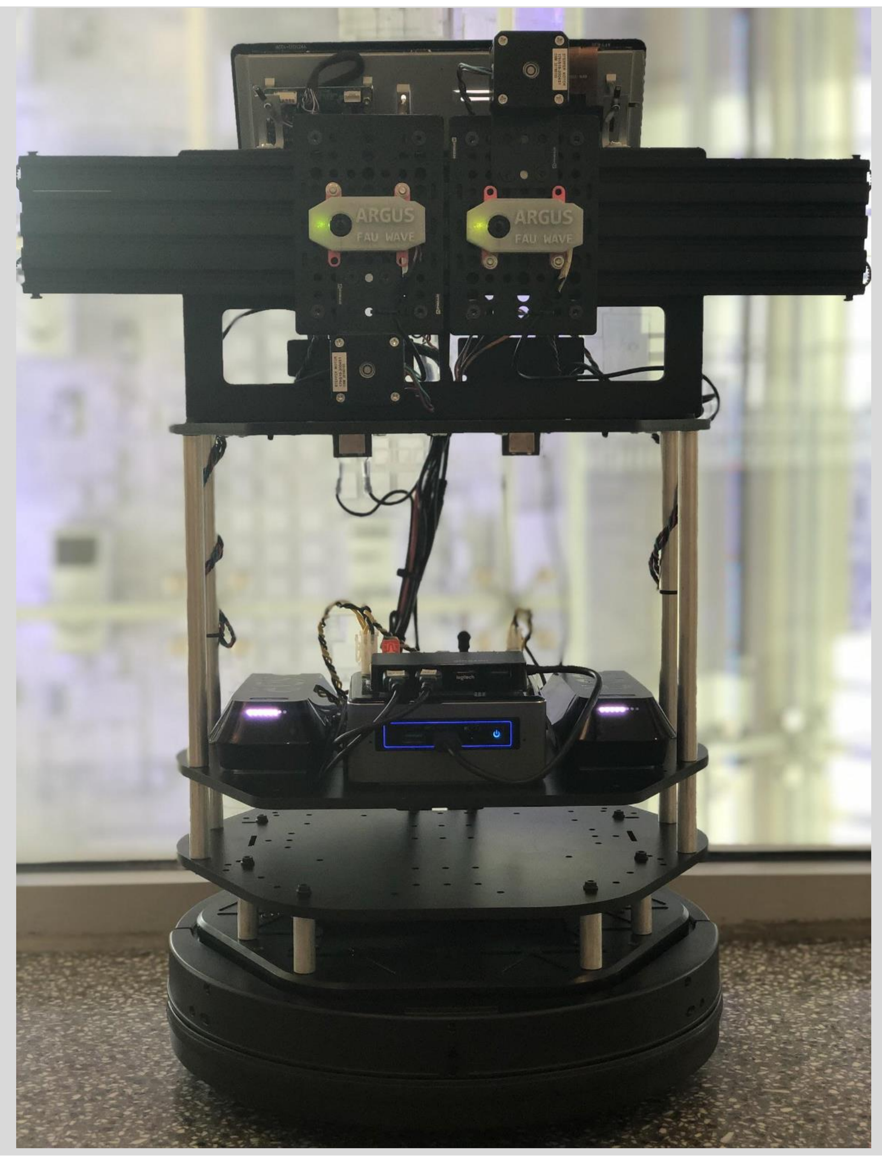
Computer vision is a growing field that has applications in many industries. Stereoscopic vision is a robust, yet error prone option for these applications. This method attempts to extract 3D information from a pair of images. Traditionally, 2 cameras are mounted rigidly and the baseline (distance between cameras) is kept constant. This creates certain blind spots and inaccuracies.

The depth error grows quadratically with distance from the camera, but it can be minimized by increasing the system baseline. Conversely, the minimum depth range grows as the baseline increases. This project attempts to take both of these factors into account.

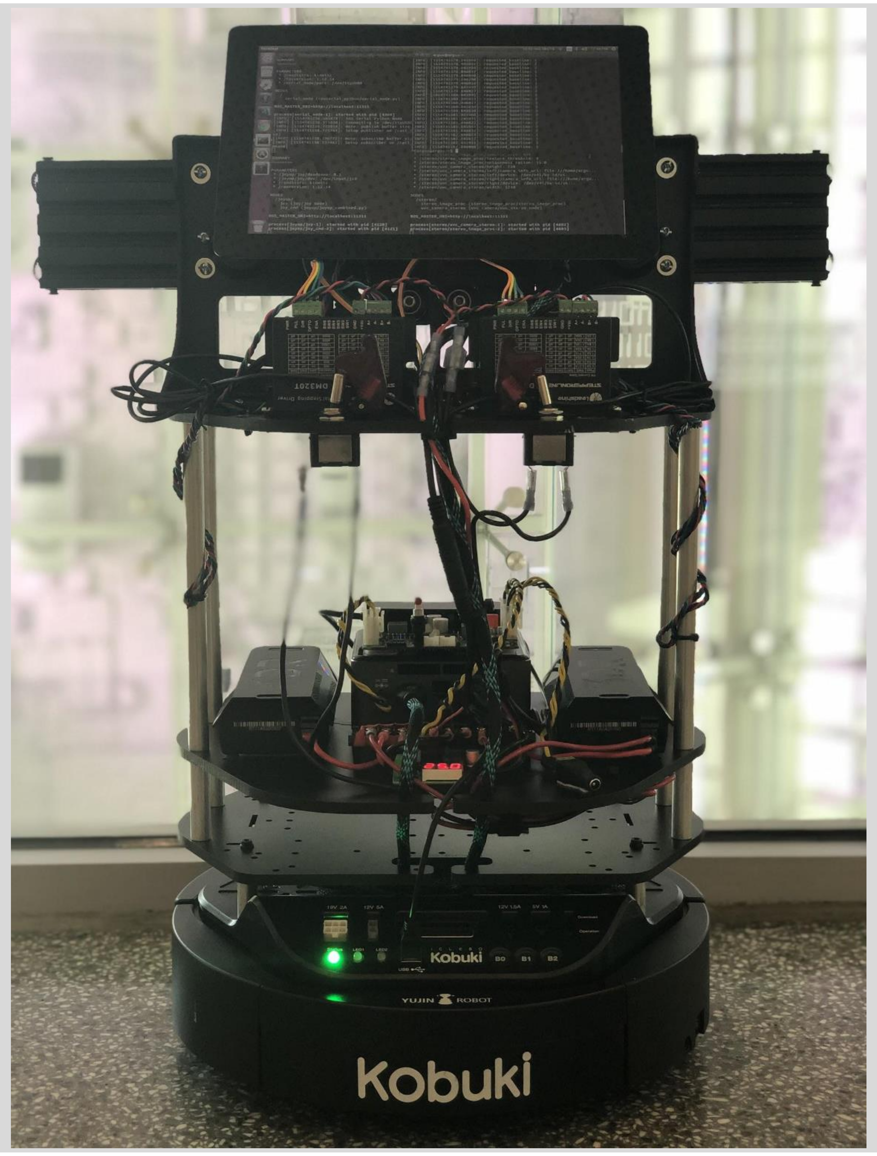
System Overview

The project consists of an active stereoscopic camera, and an unmanned ground vehicle. An Intel NUC is the primary computer. An Arduino Nano provides an interface to control positioning of the Camera System. Two 3DR Solo batteries and various voltage regulators power all components. An LCD provides access to the system graphical interface.

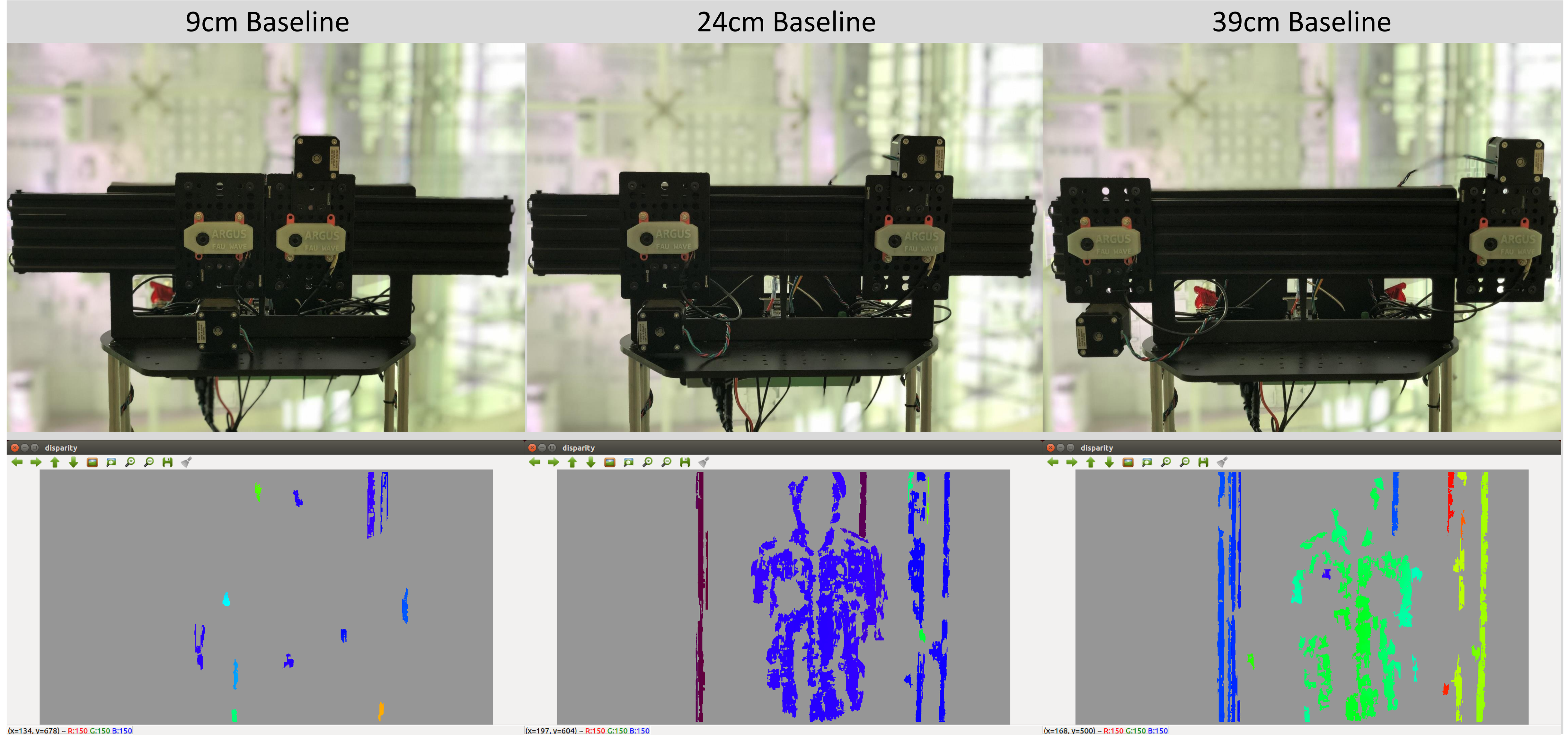
The goal of this project was to test a unique approach to stereoscopic vision, and to develop an autonomous unmanned ground vehicle (UGV) as a test platform.



Vehicle Front - Camera System, NUC, Batteries



Vehicle Back - LCD, Stepper drivers, Power Distribution



Comparison of disparity images of an object 2m away

Implementation - Camera System

Two Logitech C270 webcams were mounted on a moving rail that varied the baseline of the system. Images were taken at each baseline setting (9cm, 24cm, 39cm) and the depth information of the scene was extracted. The implementation of vision is based on *stereo_image_proc* ROS package. The camera positioning interface is a fully custom package.

Implementation - UGV

An iClebo Kobuki mobile base was used to develop the vehicle. This platform provided a pre-built mechanical solution to implement controls algorithms upon. Autonomous path planning is based on SLAM (Simultaneous Localization and Mapping) using the ROS packages *gmapping* and *navfn*, and an Xbox Kinect. The UGV creates a 2D occupancy grid where it localizes itself using various sensor inputs. The vehicle then moves to user provided coordinates by planning paths real-time, and avoiding obstacles. This avoidance is based on the *dwa_local_planner* ROS package.

Results

Each baseline was determined to outperform the other two in certain ranges:

- 9cm baseline: <1m
- 24cm baseline: 1m - 3m
- 39cm baseline: >3m