Stereoscopic Observation System (SOS)

Project Proposal Draft

Team Members: Jeana Choi, Leilani Trautman, Ryan Mansilla

Overview

- *What will it do?*

We would like to use two cameras to examine the camera's surroundings, such as the distance and location of objects around the cameras. Then, our results will be displayed on a monitor via VGA. In particular, we will be tracking a brightly colored ball (for our expected goal a green one) and calculating its distance from the cameras.

- *How do we plan to implement this functionality?*

We will first assume that we know the size of the ball beforehand. We use a second camera to remove ambiguity of the distance of the object from the camera because essentially a camera maps a 3D world into a 2D screen. Then, we can use the two cameras to detect the edges of the ball and find the radius and center of the ball from the cameras' perspective. After this, we know the proportional difference between the actual ball and the ball displayed through the cameras' lens, and we can use triangulation to calculate the distance from the cameras to the ball. Another aspect we want to implement is to track the ball - we do this by rigidly attaching a laser to the mounted cameras. Once we know the location and velocity of the ball, we can use the servo to rotate the cameras (which are rigidly attached in parallel to something) and follow its trajectory → this is a stretch goal but for now we are trying to accurately point the laser to a stationary ball after calculating its distance. We will want to display the camera data on a VGA display, where we will show the distance between the ball and camera and also other features: for example, when the edges are detected on the ball, the entire ball on the VGA display can be highlighted. Another stretch goal idea is to have a user input of which ball to track (R, G, B) and have the cameras follow the ball once it gets detected.

Goals

- *Baseline*

The baseline would have 2 cameras that can detect the green ball, which only moves in one axis.

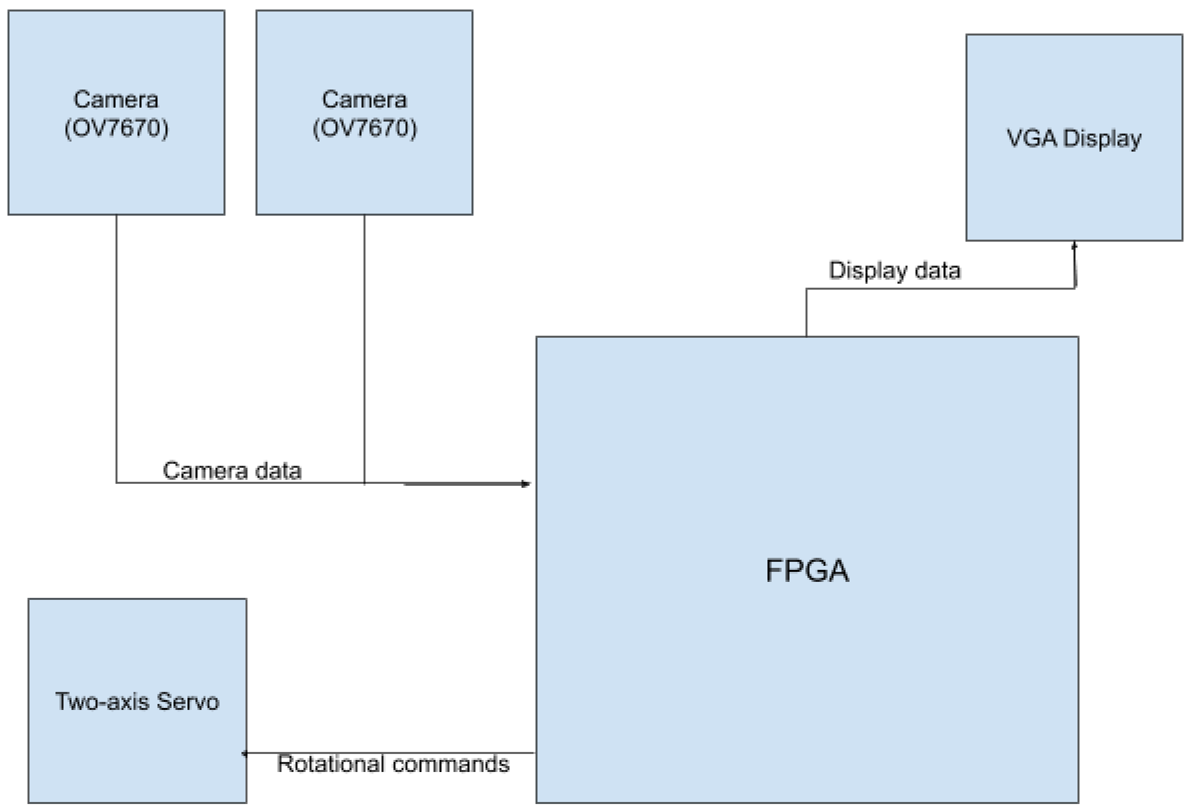
- *Expected*

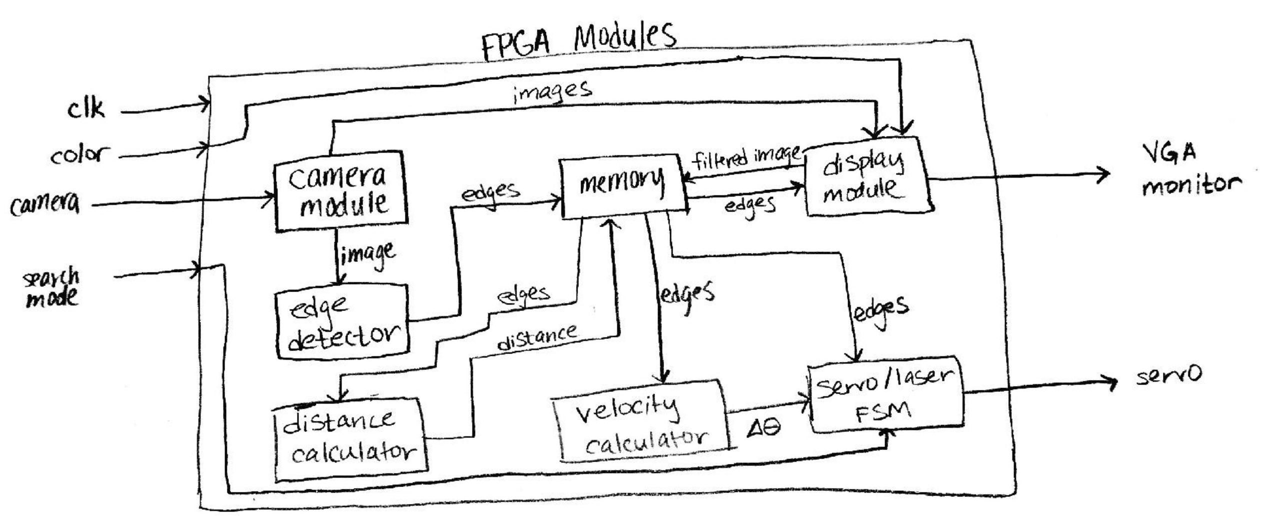
The expected goal is to have 2 cameras rotating on one servo + being able to measure the distance of the green ball. We also want the laser to be able to travel to the center of the stationary green ball.

- *Stretch*

We want one of our stretch goals to be able to detect balls in different colors (ex. User selects 'red' and the cameras detect a red ball). Another stretch goal is to be able to predict the path of the green ball based on its position and velocity. Our final stretch goal is to expand the system's capability by enabling 2-axis rotation of the cameras.

Block Diagram





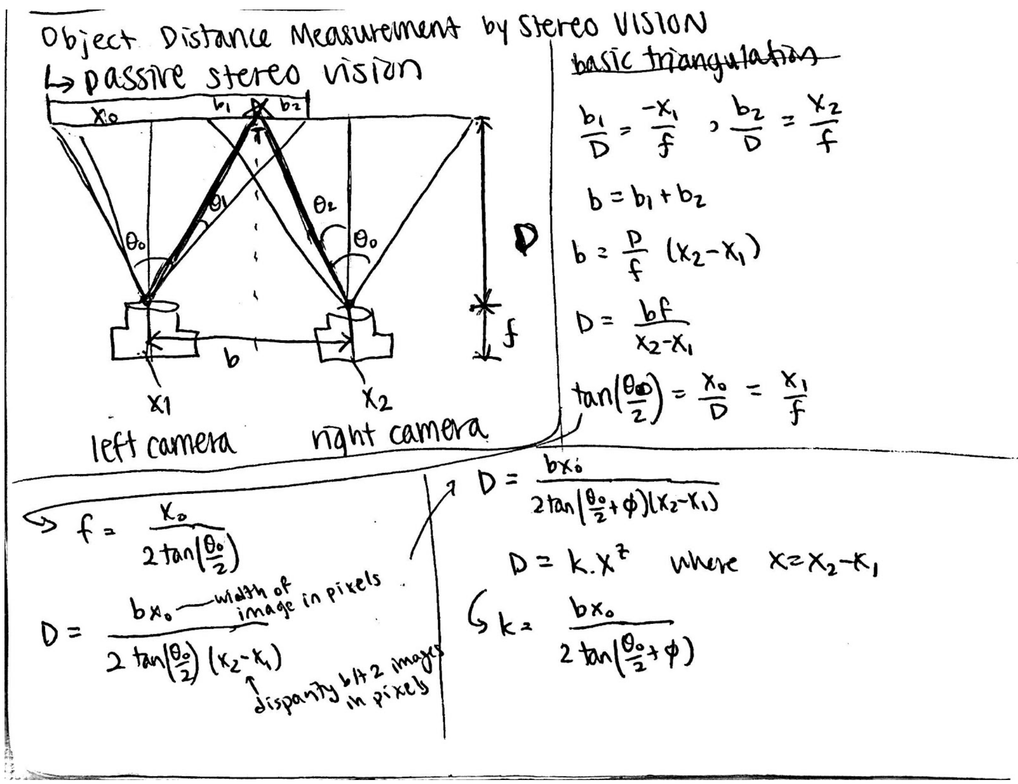
Modules:

<u>Top Level Module (Jeana)</u>

- *Inputs and outputs*
    - Inputs: image data, servo data
    - Outputs: VGA display information
- *Description of Module*
    - The top level module will integrate all of the other modules by taking in the image and servo data and feeding that into the appropriate modules, then taking the output values and feeding that data into the VGA display.
- *Complexity and level of performance (e.g., number and type of arithmetic operations, size of internal memories, required throughput)*
    - Essentially creating instances for all modules

<u>Calculate Distance Module (Jeana)</u>

- *Inputs and outputs*
    - Inputs: center point coordinates from both cameras, radius of perceived ball
    - Outputs: distance from camera to ball
- *Description of Module*
    - This module will take the locations of the target in perspective of both cameras (x1 and x2), the focal length (f), the distance between the two cameras (b), the viewing angle ($theta_0$), and we can calculate the distance D. Effectively we are doing triangulation to solve for the distance between the cameras and target.
- *Complexity and level of performance (e.g., number and type of arithmetic operations, size of internal memories, required throughput)*
    - Equations below taken from (1)



- *Describe how the module will be tested*
    - First, we can test the module by having a stationary ball and making sure the baseline is working (distance is calculated accurately). Then, we can move the ball in one axis and make sure that the distance stays the same. Lastly, we can move the ball in two axis and make sure the distance stays the same (if we stay in a plane parallel to the camera's lens).

<u>Edge Detection Module (Ryan)</u>

- *Inputs and outputs*
    - Inputs: Camera data, binary that indicates whether or not to search for edges, distance info, info on what color to look for
    - Outputs: coordinates that will go into servo module, radius of object
- *Complexity and level of performance (e.g., number and type of arithmetic operations, size of internal memories, required throughput)*
    - Sweep of image data left to right, in which comparisons are made, in terms of color. Then the two edges' coordinates will be found.
- *Describe how the module will be tested*
    - We will put an object in front of the camera, record its distance and coordinates, display it on the monitor, and then activate the module. We will then compare the object's actual edge and center coordinates and the one the module has chosen for the servo.

Servo Module (Leilani)

- *Inputs and outputs:*
    - Inputs:
        - Desired change in angle of the servo (based on laser tracking or sweeping camera to find target)
        - Sweep signal
    - Outputs:
        - Position of the servo
- *Description of Module:*
    - This module will have an FSM to figure out the state of the servo (sweeping, stationary) and set the location of the servo based on the desired position from other modules. It will output the position of the servo for other modules to use as an input. The servo will be controlled by PWM.
- *Complexity and level of performance (e.g., number and type of arithmetic operations, size of internal memories, required throughput)*
    - This module will mainly be concerned with communicating with the servo to set its position, so the complexity should be minimal.
- *Describe how the module will be tested*
    - The module will be tested by inputting different angle commands to the servo and visually confirming if the servo travels to the desired position.

Display Module (Ryan)

- *Inputs and outputs*
    - Inputs: camera images, distance data, “edge detected,” object to search
    - Outputs: camera image (as received and filtered), statistics (object under search, distance of object, velocity info), picture (to go to memory)
- *Complexity and level of performance (e.g., number and type of arithmetic operations, size of internal memories, required throughput)*
    - When just updating the live feed, the module will simply refresh. However, if filtering is required, then convolution will be done on the entire image data array, which will result in summations proportional to the number of pixels that are being filtered.
- *Describe how the module will be tested*
    - A sample object will be moved in front of the camera to confirm we have images that correspond to what we expect.

Velocity Calculator Module (Jeana)

- *Inputs and outputs*
    - Inputs: Center coordinates of the target in multiple snapshots
    - Outputs: average velocity of target
- *Description of Module*

        - This module is to calculate the velocity of the object we are tracking. We take the coordinates of multiple snapshots and take the average slope of the positions (aka take derivative of position to get velocity).
- *Complexity and level of performance (e.g., number and type of arithmetic operations, size of internal memories, required throughput)*
    - This requires multiple additions but the level of performance depends on how much data we want to work with (but should not be too complex overall).
- *Describe how the module will be tested*

This module can be tested by first having the target stationary and making sure that the velocity stays at zero. Then, we need to find a clever way to be able to measure the target moving at a constant speed (maybe control the movement with a servo attached to something rigidly attached to the target?)

<u>Camera Data Module (Leilani)</u>

- *Inputs and outputs*
    - Inputs: data from the cameras
    - Outputs: Camera images
- *Description of Module:*
    - This module will take the raw data from the cameras (8 bits of data per pixel for each camera) and construct an image array based on all the data received. There will be a separate instance of this module for each camera and it will be another module's job to correlate the two images.
- *Complexity and level of performance (e.g., number and type of arithmetic operations, size of internal memories, required throughput)*
    - This module should not be computationally complex as it will mainly be formatting the images and sending them to other modules.
- *Describe how the module will be tested*:
    - This module will be tested by connecting the camera to the FPGA then retrieving the data. To check if the data is correct, the image array will need to be converted into an image file that can be viewed and checked for correctness.

Stretch Goal Module:

<u>Laser Prediction Module (Jeana)</u>

- *Inputs and outputs*
    - Inputs: average velocity, last tracked position
    - Outputs: predicted trajectory of ball
- *Description of Module*
    - This module takes previous velocity and positions and predicts a linear path to where the ball will next be located.
- *Complexity and level of performance (e.g., number and type of arithmetic operations, size of internal memories, required throughput)*

- Depends on exactly what the inputs will become - if we have a bunch of previous velocities + positions, then we will have to compute the average velocity which is n arithmetic operations (if there are n velocities) and calculate the slope of the predicted line.
- *Describe how the module will be tested*
    - The module can be tested by seeing how the ball actually moves vs. the predicted trajectory.

List of Parts

Non-external components:

Part Name	**Quantity**	**Unit price**	**Total price**	**Notes**
OV7670 Camera	2			Apparently available in lab
2 Axis Servo	1	$18.95	$18.95	https://www.adafruit.com/product/1967?gclid=CjwKCAjwusrtBRBmEiwAGBPgEwhlhnJ-7cpVuLOvcuPDkLJSgt15tbxjBJVnRjXMQID8L4I26vN-jxoCAAkQAvD_BwE
Laser	1	$5.95	$5.95	https://www.adafruit.com/product/1054
Green sphere	1			

References:

1. *Mahammed, Manaf A., Amera I. Melhum, and Faris A. Kochery. "Object distance measurement by stereo vision." International Journal of Science and Applied Information Technology (IJSAIT) 2.2 (2013): 05-08.*