Stereo Vision for FRC Robots with Webcams
Gary Lewis, Team 1311 Mentor
Draft, version 0.1

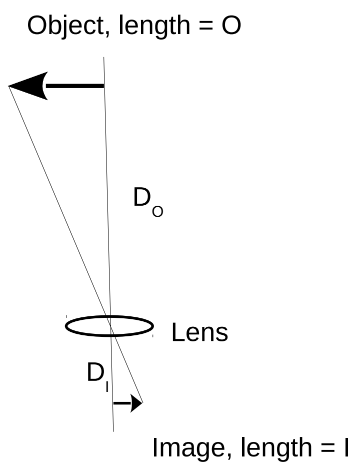
Introduction

These are some rough notes relative to using webcams for stereo vision. In general, webcams are not good for stereo vision because of difficulties with synchronizing the cameras and the fact that they use rolling shutters.

These notes are made in the hope that webcams will be sufficient to get the disparity for a single object, and thus be able to provide distance information for that object.

Background for Calculations

The typical equations for stereo calculations use the lens distance formula below.



Distances and sizes are related by similar triangles.

$$\frac{D_O}{O} = \frac{D_I}{I}$$

Note that the typical calculations use the focal length (f) of the camera for D_I. This is not correct in general, but they are essentially the same for fixed-focus cameras (focus at infinity). Also, if you look in an optics textbook, that equation will usually have a negative sign because the distances are in opposite directions.

We need some specific, useable calculations, so we will start with the equation for disparity from the OpenCV documentation. This is derived by using the equation above on each of the cameras and combining them.

$$disparity = x - x' = \frac{Bf}{Z}$$

B is distance between cameras
Z is distance to object

x is offset at image plane in left camera
x' is offset at image plane in right camera

All variables in this equation are distances (inches, cm, etc.)

Note that the distance Z is the perpendicular distance to the object (perpendicular to the line of the cameras), which may not be what you want for distance. Dealing with that is described later.

Practical Results

We want to get the object location from the pixel offset in the two cameras. In the following, we will used R and L subscripts to denote variables associated with the left and right cameras. The disparity equation is then:

$$x_L - x_R = \frac{Bf}{Z}$$

If we use:

p to represent the pixel offset in a camera for a particular object point
s to represent the pixels per inch of the cameras

Then:

$x = p/s$

And the disparity equation becomes:

$$\frac{p_L - p_R}{s} = \frac{Bf}{Z}$$

Or:

$$p_L - p_R = \frac{Bfs}{Z}$$

In terms of the object distance:

$$\boldsymbol{Z = \frac{Bfs}{p_L - p_R}}$$

With some caution, we can use this to calculate distances.

1. The product *fs* (“focal length” times pixels per inch) can be determined by a calibration procedure, and should always be the same for a particular camera.
2. Any program will have to check for zero pixel offset to avoid dividing by zero. If things are properly calibrated, this would only occur when the object is at infinity (or, in practice, a large distance.)

We can use the original lens equation to also calculate the offset (*O*) of the object from the cameras.

$$O_L = \frac{x_L Z}{f} = \frac{p_L Z}{fs} \quad \text{and} \quad O_R = \frac{x_R Z}{f} = \frac{p_R Z}{fs}$$

The offset from the centerline of the cameras is then:

$$O = \frac{p_R Z}{fs} - \frac{B}{2} \quad \text{or} \quad O = \frac{p_R Z}{fs} + \frac{B}{2}$$

With these we can use Z and O to calculate the actual distance and angle to the object.

$$distance = \sqrt{Z^2 + O^2}$$

$$angle = \arctan \frac{O}{Z}$$

Calibration and Precautions

Turn off any autofocus! This will mess up the f parameter. (Logitech C930e has autofocus.)

Check for the divide by zero condition mentioned above, and handle it.

Calibration will be needed for two things:

1. To determine the product *fs*. This should be inherent to the camera and not change. In principle "*s*" would change if the camera is not perpendicular, but this should be negligible for small angles.
2. To determine the "center pixel" for each camera (zero point for p). This will be highly dependent on camera angle (even small ones), and will need to be done regularly.