Computer Vision Plan – Team 19218 – By Jora

Purpose

This plan will define an algorithm to locate where the closes alliance / neutral sample is on the field and subsequently allow the robot to align and collect the sample from the ground.

Scope

This plan assumes a camera mounted so that it is able to get a bird's eye view of the field area being scanned for samples.

This plan also assumes that the camera is mounted in close vicinity and at the same height as the intake mechanism when attempting to scan for samples.

This plan assumes even lighting conditions when scanning for samples.

This plan assumes that the installed intake can collect a sample placed on the ground at any angle. The algorithm may be modified to compensate for an intake that doesn't achieve this, however that is out of scope for this document.

For the sake of this document we will assume that we are on the red alliance, thus only searching for red or yellow samples.

Algorithm

Overview

To achieve our goal, find the closes sample within our view port, we will use the EasyOpenCV library to filter and scan for samples on the ground. Using the library we will filter – depending on our alliance colour – Red or blue samples, yellow samples will always be scanned for.
As we will be working with pictures from the webcam, we will look at an image pixel by pixel using our code. A pixel can only be one colour at a time, so we will scan through the image until we find the closest pixel with the colour we are looking for.

Once this found we will relate where this pixel is relative to the intake mechanism and provide distances for the robot to travel in order to align itself over the sample and collect it.

Step 1 – Viewport Division

Divide the Camera view into 9 boxes. This will let us map where the samples are from their pixel locations to a relative physical location. We will number the columns and rows for easy reference, as seen below in figure 1.

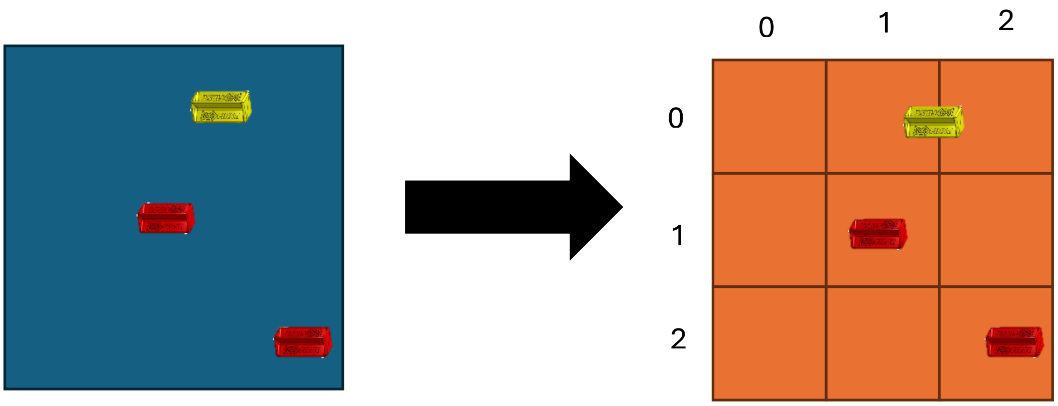

Figure 1 - Defining of Grid on Viewport

Step 2 – Pixel search

Starting from the pixel found to correlate to the center of the intake if it was lowered to the ground, we will perform a middle out search, where our search area increases from a center point (think of a growing rectangle). This will allow us to find the closes sample. Figure 2 shows a visualisation of this step, where the search area (white) grows in each loop *i,* in figure 2 we also assume the intake position is at the center of the image.

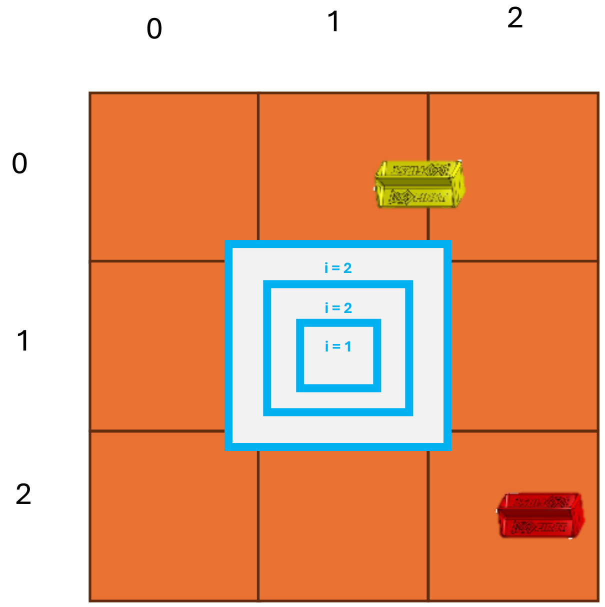

Figure 2 - Middle-Out Search area from center

Figure 3 shows a search area calibrated to an intake position that does not correlate to the center of the image. This is most likely what we will see in real life as the camera will have to be mounted at an offset to the intake. Figure 3 depicts an intake where the camera is mounted horizontally Infront of it when in the scanning position, and thus we need to start our scan from lower in the picture.

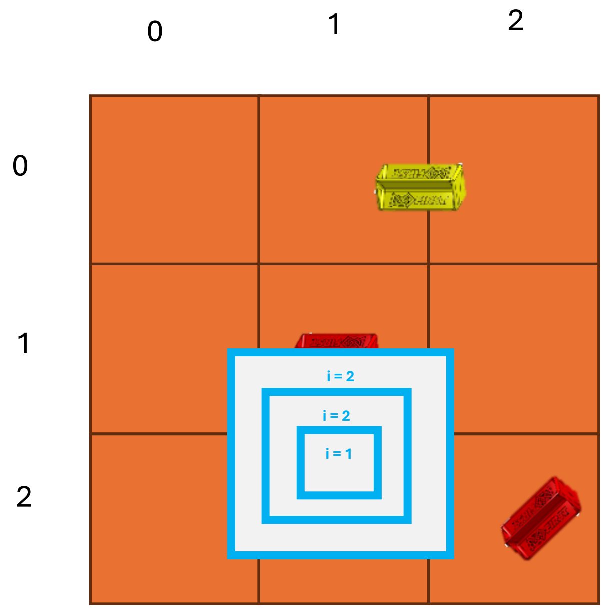

Figure 3 - Offset Middle-Out Search

Step 3 – Find Target Pixel

When we do find our sample using the method in step 2, it will always be the edge of the pixel. Thus, we must find the center of the pixel so that we may properly align the robot to the midpoint of the pixel. To do we *draw* 6 rectangles in intercardinal directions (North, North-East, East, South-East, South, South-West, West, North-West) and average the amount of desired colour in each rectangle. See Figure 4 for what this will look like when the first sample found is in row 1, column 1. The boxes we draw are in yellow, light green and dark green, and the first pixel found is blue.

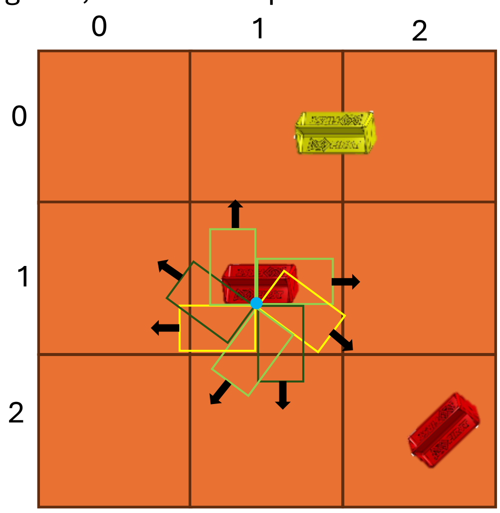

Figure 4 - Rectangles drawn on a sample

Once we determine which rectangle has the greatest amount of colour in it we divide it into two rectangles, and figure out which one of those has the greatest amount of colour. This will allow us to send the intake to the point of the sample where it is most likely to grab it. Figure 5 shows what this will look like.

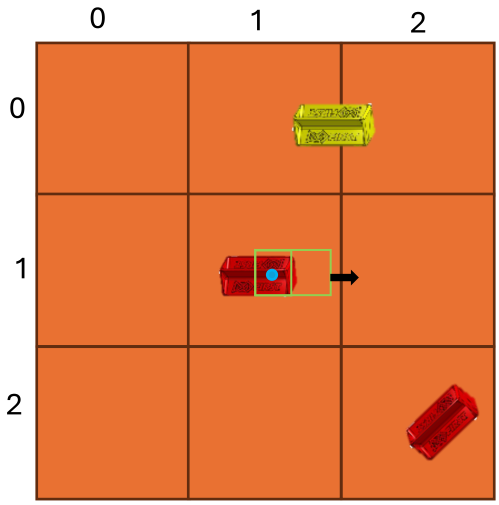

Figure 5 - Further subdivision of the search rectangle

We then can find the target pixel to send the robot to (blue).

Step 4 - Find Distance from Center of Intake to the Target Pixel.

We then need to find the distance from the intake position and send it the target position. This can be found using the coordinates of each pixel and the distance formula,

$$d = M \times \sqrt{(targetPixel_x - intakePixel_x)^2 + (targetPixel_y - intakePixel_y)^2}$$

where *d* is the distance, the intake needs to travel to get to the Sample and *M* is the modifier we find by trial and error. *TargetPixel* is pixel we found in step 4 and *IntakePixel* is the pixel we have found our intake to drop to initially when searching.

We can then use this distance to move our robot to the desired point.