Computer Vision Targeting System Development

Overview:

The computer vision targeting system allows the robot to fire with maximum accuracy and efficiency without the need for human intervention.

Step: 1	Date: 01/21/2023	Engineer: Jeremy Eastham
Witness: Mariana Sanchez	Date Reviewed: 02/22/2023	

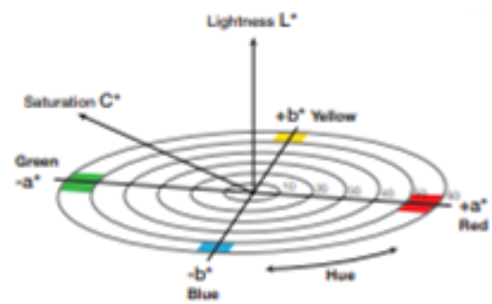
Objective:

- Research preprocessing algorithms for the Raspberry Pi vision system

Remarks:

This problem is relatively simple, but certain aspects are also immensely complex. Given a video feed on top of the robot pointed in the direction that it is targeting, locate the angle of the goal relative to the robot. This angle can be used to dynamically adjust the robot's aim in real time. The team decided to use OpenCV-Python for this task due to its performance and usability.

Video data is not complex, but there is a lot of it. This data needs to be processed efficiently, so the first problem is narrowing down the amount of data that we have to analyze. If this is done intelligently, this allows the algorithm to perform faster without sacrificing accuracy.



Video data is optimized for display on a screen. The RGB color space does not correlate very well with the way that humans perceive color and light. The vision algorithm needs to be resilient in varying lighting conditions, so the first preprocessing step will be to manipulate the color of the image into a more useful space. The L*a*b* color space was chosen due to its lightness component mapping well to the amount of ambient light. The a* and b* components encode color in such a way that mimics human perception. This color space is used frequently in photo editing software due to this useful property. This means that we can perform color isolation for field elements using thresholding in the L*a*b* color space.

It is also possible to crop the image coming from the camera based on the previously determined angle of the goal. However, if the robot is turning or the camera is rotating, we need to ensure that the goal will not be cropped out of the video.

Image Credit: X-Rite (Pantone)

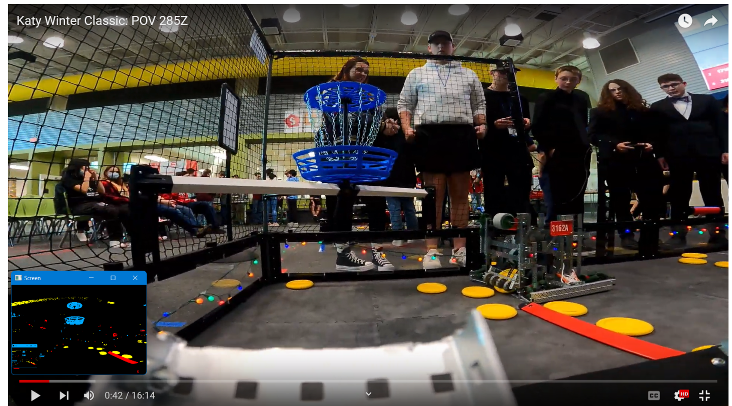
Accomplishments:

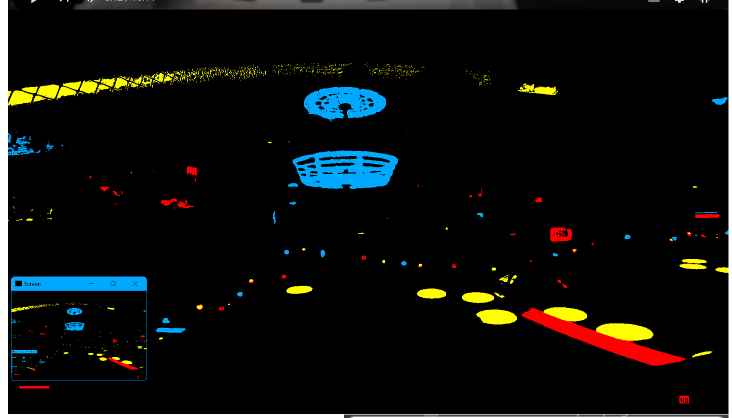
- Designed an algorithm to preprocess video data

Step: 2	Date: 02/01/2023	Engineer: Jeremy Eastham
Witness: Mariana Sanchez	Date Reviewed: 02/22/2023	

Objective:

- Implement a testing program and experiment with preprocessing algorithms

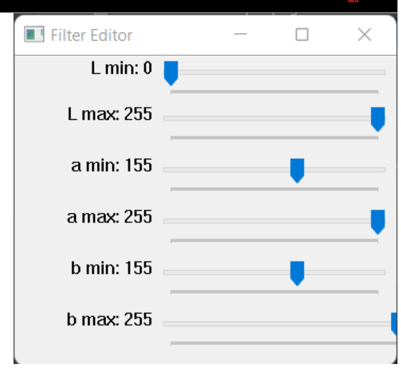




Remarks:

It is much easier to develop vision code without having to test on an actual robot. The vision program was built from the ground up with flexibility in mind. It allows processing any video source including webcams, video files, and screen captures. Initial testing was done using video from a high quality camera mounted on a robot at the Katy Winter Classic.

The team needed a way to visualize the image processing algorithm in real time. To do this, an output window was added to the program that displays the processed image (as seen in the bottom left of the first image). Another window was added that can be used to adjust the variables of a filter while simultaneously viewing the output, making it unnecessary to relaunch the program every time and allowing for faster iteration.



The first objective was tuning color isolation filters to identify field elements. First, each frame of the video is converted into the L*a*b* color space, then thresholding is applied to the image. The pixels that satisfy this threshold are put into a mask that is visualized as a solid color in the output window. The next step will be goal isolation.

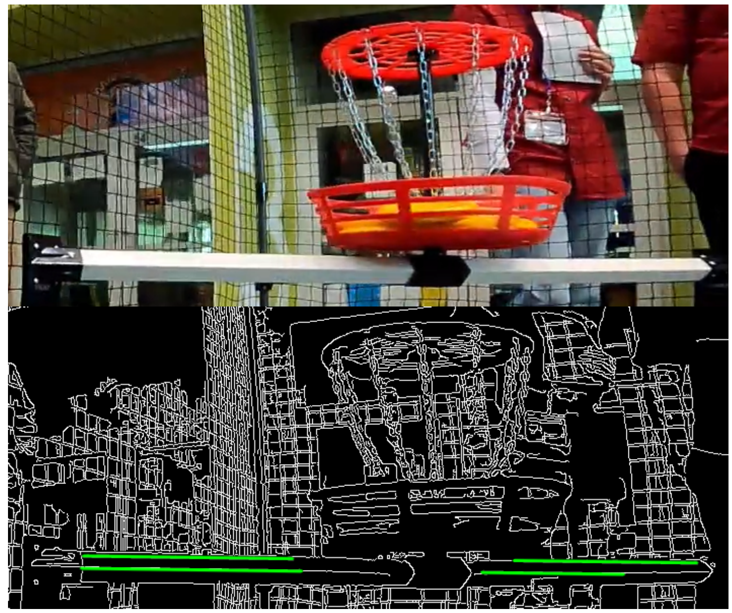
Accomplishments:

- Created a vision program that can run without a robot, visualize output, and tune filter parameters in real time
- Isolated the colors of field elements to be used for further processing

Step: 3	Date: 02/11/2023	Engineer: Jeremy Eastham
Witness: Avi Punjabi	Date Reviewed: 02/22/2023	

Objective:

- Implement a goal identification algorithm



Remarks:

Now that important colors have been isolated, the goal needs to be detected in order to determine the aim angle. One of the ways to do this is by using Canny Edge Detection. As seen in the second image, Canny Edge Detection is very effective at finding all of the edges in the image. Now that the edges are isolated, the long silver bar under the goal can be detected. The longest, flattest lines can be identified with a Hough Transform. Combined with the color data that has already been isolated, the goal can be precisely located.

Accomplishments:

- Implemented a goal detection algorithm

Engineer: Jeremy Eastham Witness: Mariana Sanchez Date Reviewed: 02/26/2023