Vision Team Report : Team ShoeStalker
Computational Robotics

Written by

Amanda Sutherland, Claire Diehl, and Jasper Chen

November 10, 2014

Olin College of Engineering

Contents

List of Figures

1 Project Goal

The goal of this project was to program the Neato to follow shoes using the Neato's camera. Our MVP was to use keypoints, and to extend it to a combination of keypoints and histogram if we had the time.

2 End Result

At the end of the project period we ended up with a script which used keypoints and histogram to follow a particular shoe based on a captured image.

3 Problem Solving

We originally based much of this code on the object detection that can be found in the computational robotics github. We started by going through the code to find the parts that were relevant to us and rewrite them to make it a bit more readable for ourselves. From here we needed to write the code that would actually get the image from the robot. There was a bit of struggle with this so we wrote all the pseudo code for the robot moving and the actual data processing. After figuring this out we [progressed through the pseudo code, integrating it. After the Neato followed shoes with keypoints we added histogram tracking as well.
The full end product can be found here:

https://github.com/AmandaSutherland/ShoeStalker.git

4 Design Decision

A major design decision that we made in implementing the goal was to include both the histogram and keypoint detection.
Initially we started only with keypoint detection but after testing found, in some situations, keypoints have significant trouble detecting the shoes. We added histogram code to increase the accuracy of finding shoes. A major issue with this is the decisions that the robot makes considering both data sets. If the found areas are averaged, if one detection method gets very far off (as seen in the left most image of Figure 1), the robots responses would be totally wrong. Instead we gave each set of data a value based on how much they resemble the reference image. If neither data set was close enough the robot enters lost shoe mode.

Figure 1: Finding a shoe. From left to right - the histogram live capture, the keypoints displayed on the live capture with the reference image in the top right, and the window tracking the bounding box from keypoints.

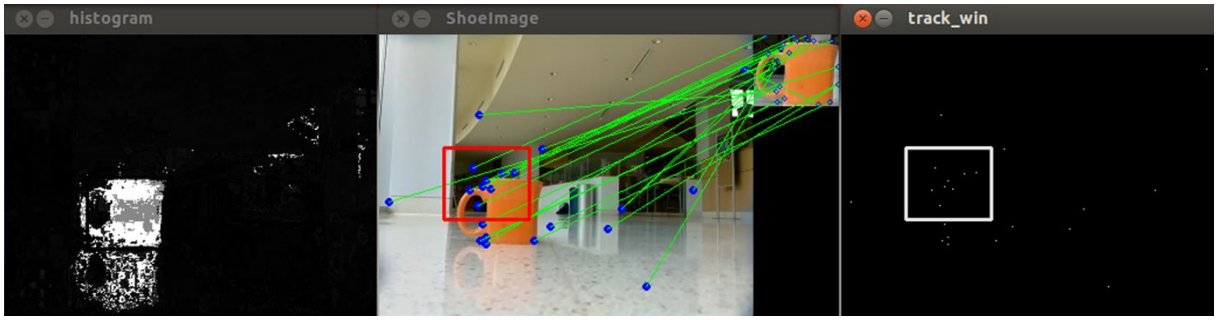

Figure 2: Finding the orange mug. From left to right - the histogram live capture, the keypoints displayed on the live capture with the reference image in the top right, and the window tracking the bounding box from keypoints.

5 Code Structure

All of our code is in one file. When our code is run, the code prompts the user to take a picture of the shoe of interest. This is the image that the code uses to compare to our video stream. Our code has four main parts: image selection and processing, detecting the shoe, moving towards the shoe and lost shoe. The code takes in the image supplied by the Neato and streams it to a window. This is where the user selects the shoe. While the Neato has the shoe in sight, by default, the Neato follows it using keypoint detection. It switches to histogram detection if there aren't enough matching keypoints per total keypoints in the bounding box. The histogram detection behavior compares the picture of the shoe to the image in the bounding box. The position of the bounding boxes for both behaviors provide the direction for the Neato. If the location provided by keypoints or histograms is not considered good enough the code enters lost shoe mode. In this mode the Neato spins until it refinds the shoe.

6 Challenges

One particular challenge we encountered was capturing an image. Issues with CvBridge prevented us from having an image to work with, and therefore we were not able to test code we wrote until at least a week and a half into the project. This meant we left all of our debugging until later, which meant a large portion of the work was left for the second half. Debugging the large quantity of code written also took longer than was optimal.

Another challenge was refinding the shoe once the Neato had lost it. We added a behavior to find the shoe, but further work would be nessicary to make this a more reasonable action(see future improvements section).

7 Future Improvements

A significant improvement to our code would be in how we refind a lost shoe. Currently we look for the shoe near where it was lost, but as the shoe is almost never close to that place we cannot refind it. If we had the time to change this, our technique of spinning to find the shoe would be much more effective.

Another significant improvement would be to work on cutting out the noise from the surroundings. Often the keypoint section got confused by things such as the carpet or background objects, which could be fixed with a better histogram section. If the image fed to the histogram were blurred slightly, it would be more capable of finding areas which match the colors of the shoe.

8 Lessons Learned

This project would have been much easier to make progress on if it had been done in testable chunks. For example, not having an image was such a significant issue we could not make real progress with anything else. If we had set up the code in smaller scripts or set it up such that we could use the webcam for image capturing we would have been able to work in parallel.