Traffic Barrel Detection

Rohan Ondkar
College of Computer, Mathematical, and Natural Sciences
University of Maryland, College Park
Maryland, United States
rondkar@terpmail.umd.edu

Naveen Harish
College of Computer, Mathematical, and Natural Sciences
University of Maryland, College Park
Maryland, United States
nharish@terpmail.umd.edu

Rahul Shah
A. James Clark School of Engineering
University of Maryland, College Park
Maryland, United States
rshah135@terpmail.umd.edu

Judy Song
College of Computer, Mathematical, and Natural Sciences
University of Maryland, College Park
Maryland, United States
jsong902@terpmail.umd.edu

Paulius Vilimas
College of Computer, Mathematical, and Natural Sciences
University of Maryland, College Park
Maryland, United States
pvilimas@terpmail.umd.edu

***Abstract*—This project presents an approach to detect traffic barrels using colors from a camera.**

I. Problem Statement

In this project, among two visually different types of traffic barrels, we aim to detect only the kind with white stripes. The algorithm is tested on a video that includes barrels with and without white stripes. This concept can be extrapolated for other projects involving computer vision, such as searching for specific targets to travel to.

II. The Approach

Our approach was to split the video frame-by-frame implicitly, while searching for the correct color combinations in each frame and outline only those. Because there are multiple barrels in the video, some of which do not have the white stripes, we had to search each tracked barrel for the white stripes and only outline those. We bounded the color orange to be tracked, as that was the more prominent color of the barrel. In order to achieve this, we first provided a range of oranges, whites, and grays for color detection. Afterwards, we converted the frames into gray-scale images, then finally inverted binary images for an easier outline process (as seen in figure 1.) With our code, OpenCV was able to stitch each frame implicitly and combine that into a resulting video with all the outlined shapes, so we did not have to worry about creating images for each frame.

III. Failure Cases

There was an issue with Rohan and Rahul's OpenCV's in which they would not work even though they installed and imported the OpenCV. It also was unable to work on Naveen's computer, but the solution towards this problem was to run the program as python3 and not just python. One of our main issues through this program was also just completing the program. None of us have used OpenCV and thus learning what everything did was a process in and of itself. The main issue was that we weren't able to distinguish between barrels that were orange and white and those that were only orange. We attempted to separate the white and the orange from the background. However, because of the nature of the colors in HSV format, we failed to exclusively only include those colors. For example, the green pixels that were extremely close to yellow were being outlined as well. Another issue we encountered was drawing a complete rectangle around the traffic barrel with white stripes. Since it was difficult to detect white, only the orange stripes were outlined.

IV. Sample Images

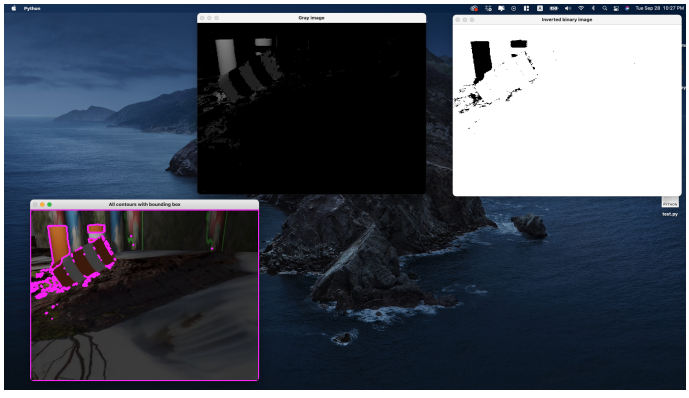

Fig. 1. A picture of the camera

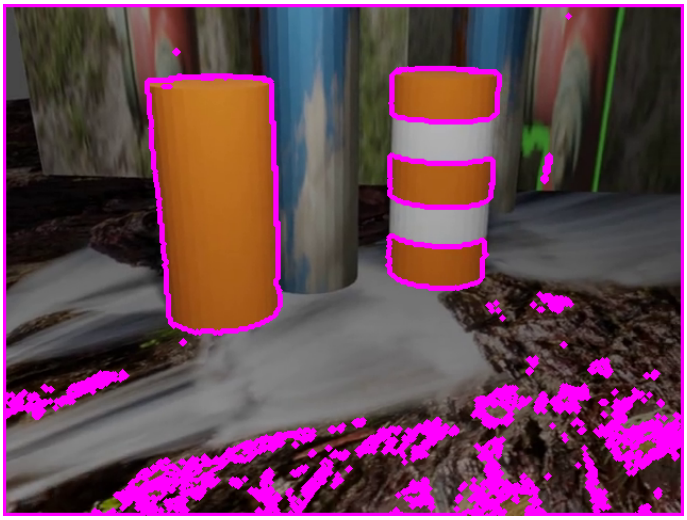

Fig. 2. A picture of the camera

V. Work Distribution

Rahul and Rohan worked on Latex and formatting getting everything on the paper. We got the program to function on Judy and Naveen's computer first, so we decided to just stick with them being the main programming force for this project. We all contributed by looking up articles and videos to teach us the commands as well as getting our ideas down for the project. We also added Paul later in the week for this assignment and spent the day getting him used to OpenCV. Naveen caught Paul up to speed on what our approach to the assignment is. He then helped out with the coding portion of the project and we went forwards with our attempt to complete the assignment.

References

[1] automaticaddison, A. (2021, February 28). How to detect and draw contours in images using opencv. Automatic Addison. Retrieved September 29, 2021, from https://automaticaddison.com/how-to-detect-and-draw-contours-in-images-using-opencv/.

[2] Nimadorostkar. (2020, March 6). Color-detector/main.py at master · NIMADOROSTKAR/Color-detector. GitHub. Retrieved September 29, 2021, from https://github.com/nimadorostkar/Color-Detector/blob/master/main.py.