Lab 2 – Camera: line following

Components: Intel RealSense Depth Camera D435i

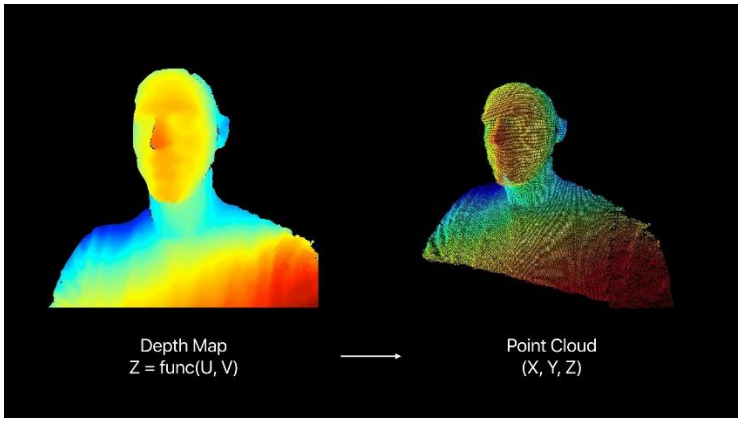
Software: OpenCV

Concepts: HSV thresholding, color masking, contour drawing,

Background

Camera
For our line following lab, we will be using the Intel RealSense Depth Camera D435i.

This camera can sense depth which makes it useful for autonomous navigation and will be used to capture images for our line following. The camera can generate depth images as well as 3D point clouds like those shown below. It also contains an IMU (inertial measurement unit) which can detect speed and tilt.



OpenCV

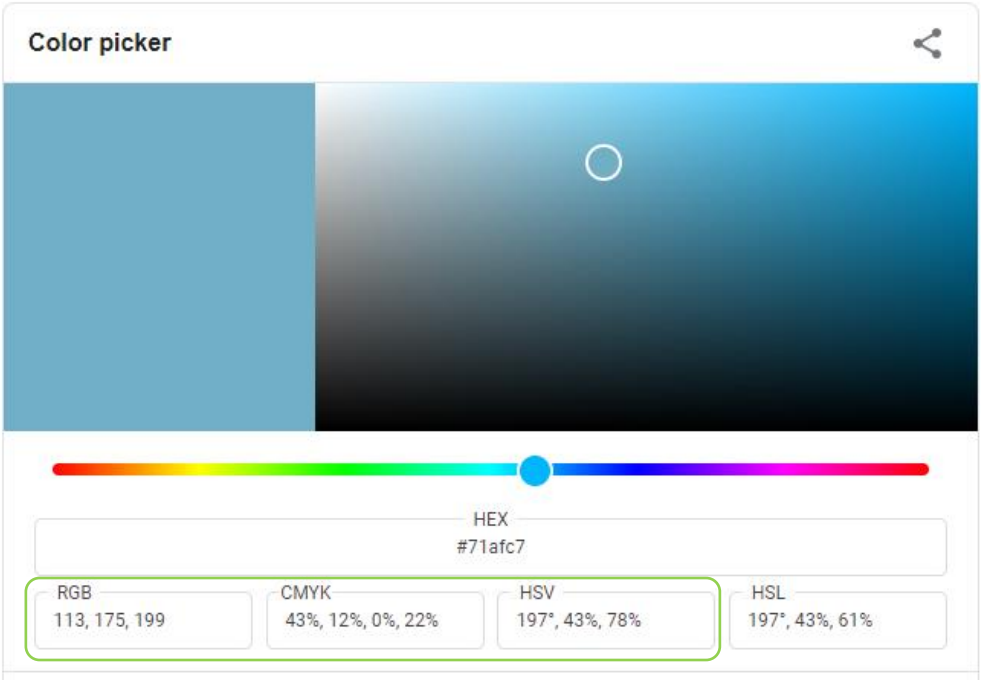
To understand what OpenCV is, we can gain some intuition by understanding the name. OpenCV stands for Open Source Computer Vision. Open source indicates that all the original code from OpenCV is freely available and may be modified for personal use and redistributed. **Computer Vision** is an area of study which focuses on helping computers understand the content of images and videos. OpenCV is an open source computer vision library which gives us the necessary tools for line following.

Concepts

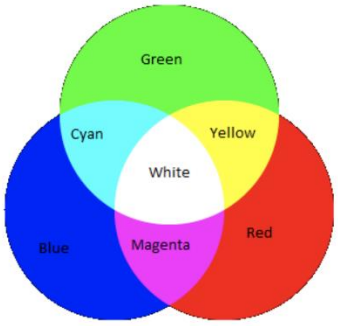
A key component of line following is **object identification**. Our camera will return images to us, but we need to be able to extract the important information. In this case, we want to identify the line as an object which requires **HSV thresholding**, **color masking**, and **contour drawing**.

HSV Thresholding

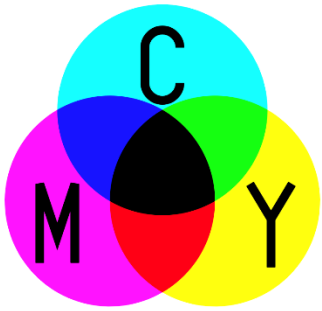
There are different standards of identifying colors using **color spaces**. Color spaces cover a subset of visible color and make it easy to reproduce colors properly. Some very common color spaces are **RGB**, **CMYK**, and **HSV**. We do not use RGB or CMYK in this project, but they can help us understand how color spaces are used. Here you can use an online color picker which displays RGB, CMYK, HSV, and other color space values.



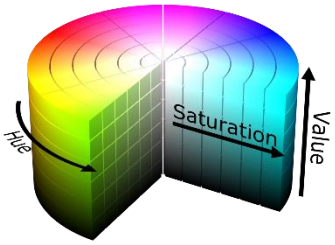
color picker with RGB, CMYK, and HSV values



In RGB color space we indicate the amount of red, green, and blue in a color. Combining red, green, and blue we can create all the colors in the space. When we have high levels of each color, we create white. When we have low levels of each color, we create black. This is how your computer screen displays colors.



In CMYK color space we indicate the amount of cyan, magenta, yellow, and black (key) in a color. Cyan, magenta, yellow, and black combine to make all colors in the space. Combining these colors has the opposite effect of combining RGB colors. High levels of each color create black, and low levels leave white. CMYK is commonly used in printing



With HSV color space we indicate the **hue**, **saturation**, and **value** in a color. Unlike RGB and CMYK, we do not combine different colors. Hue determines the general color. Saturation determines how **vibrant** our color is from muted to very colorful. Value determines how **light or dark** our color will be. HSV is more intuitive to most and is commonly used for digital color pickers.

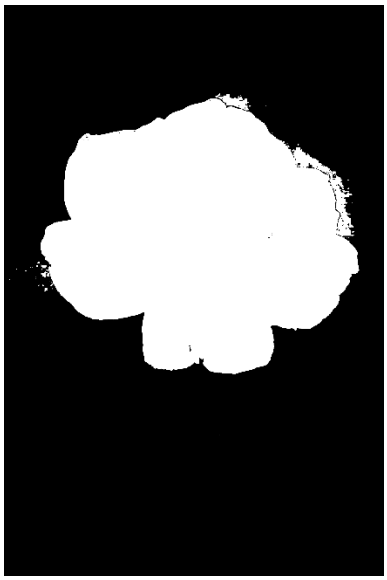
If we look back at the color picker, it uses HSV! There is a slider for hue and a 2D, saturation/value space to drag our color selector.

We will use a range of HSV colors as a threshold for masking. By identifying which range of colors to allow we will be able to extract the correct part of an image.

Color Masking

Masking blacks out all parts of the image which we want hidden and whites out all parts of the image we want to be seen. **Color masking** uses the original image and the black and white mask to create a color mask. White areas of the mask will have color from the original image and black areas will remain black.

original Image

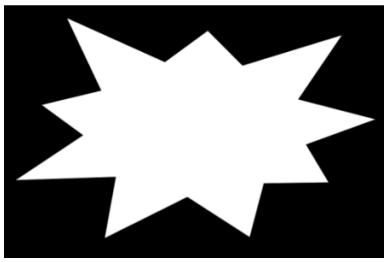


masked image

color masked image

Masking does not always have to take into account the original image. We can use any black and white mask we want to create a color mask.

original Image



mask

color masked image

Contour drawing

A contour is a curve joining all continuous points along a boundary. OpenCV determines the boundaries of an object by creating contours around areas of same intensity.

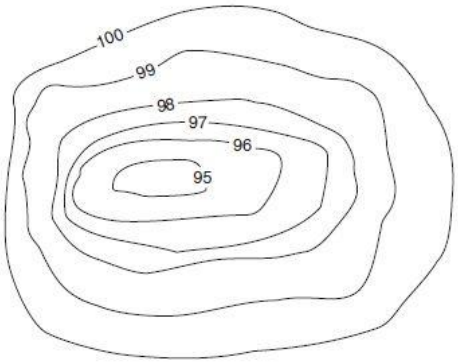
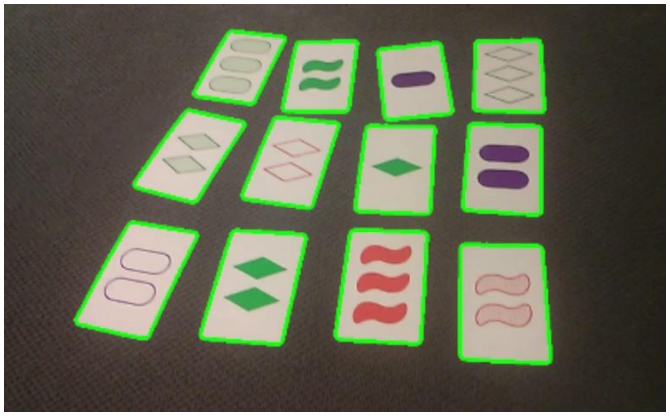


Fig. 17.1. Contours



Lab Guide

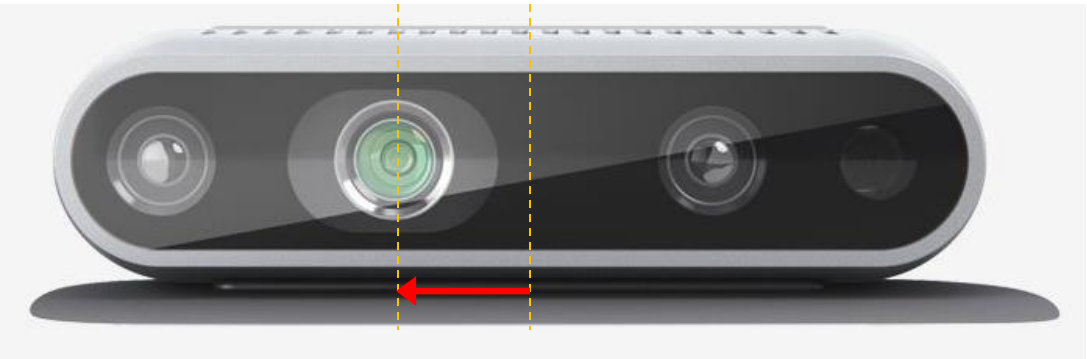
For our main goal students will be tasked with following a line using outlined concepts: HSV thresholding, color masking, and contouring.

Two parts: Identification and Following

Identification

Following – Camera Offsetting, updating speed and angle, adjust parameters

Before following an identified object, it is important to account for the camera offset. The RealSense camera is not perfectly centered with the center of the car. This can be adjusted by centering the car in front of an object or line and readjusting the center by setting `SCREEN_CENTER` to the correct point on the x-axis.



From our contour we can follow the line. However, if you have a long line will your car always know what to follow? Instead of using the whole image, it is important to use the most useful part of the image, the bottom. The bottom of the image is where the line is closest to the car. Restricting the contour to the closest part of the line we can adjust where we drive based on where the center of the cropped line is.