Mansib Ahmed

CMPSC 497

10/17/2023

Lane Detection with OpenCV

Objective: To use Python’s OpenCV library and Hough lines to detect highway lanes

Materials:

1. Python 3.9.7 anaconda interpreter
2. OpenCV
3. Numpy
4. Jupyter Notebook
5. Youtube guide: <https://www.youtube.com/watch?v=eLTLtUVuuy4&t=3338s>

Test images:

A car driving on a highway

Description automatically generated

A road with trees and a blue sky

Description automatically generated

Code:

import cv2 as cv

import numpy as np

def canny(image):

    gray = cv.cvtColor(image, cv.COLOR\_RGB2GRAY)

    blur = cv.GaussianBlur(gray, (5,5), 0)

    canny = cv.Canny(blur, 50, 150)

    return canny

def display\_lines(image, lines):

    line\_image = np.zeros\_like(image)

    if lines is not None:

        for line in lines:

            x1, y1, x2, y2 = line.reshape(4)

            cv.line(line\_image, (x1,y1), (x2, y2), (0, 0, 255), 5)

    return line\_image

def region\_of\_interest(image):

    height = image.shape[0]

    polygons = np.array([

        [(200, height), (1100, height), (550, 250)]

    ])

    mask = np.zeros\_like(image)

    cv.fillPoly(mask, polygons, 255)

    masked\_image = cv.bitwise\_and(image, mask)

    return masked\_image

image = cv.imread("lane1.png")

lane\_image = np.copy(image)

canny\_image = canny(lane\_image)

cropped\_image = region\_of\_interest(canny\_image)

lines = cv.HoughLinesP(cropped\_image, 2, np.pi/180, 100, np.array([]), minLineLength=40, maxLineGap=5)

line\_image = display\_lines(lane\_image, lines)

combo\_image = cv.addWeighted(lane\_image, 0.8, line\_image, 1, 1)

cv.imshow("lanes", combo\_image)

cv.waitKey(0)

Conclusion:

Overall, this program successfully detected lane lines on an interstate highway. The only drawback is that it also detected lines other than the lanes such as railing on the side of the rode and even horizontal edge lines on cars. The angles, orientations, and width of the detected lines could be tweaked further to filter these unwanted lines.