Mansib Ahmed

CMPSC 497

10/19/2023

Red Object Tracking with Jetbot

Objective: To design and test red color detection algorithm with Jetbot

Materials:

1. Jetbot
2. Python 3.9.7 anaconda interpreter
3. OpenCV
4. Jupyter Notebook

Test images:  
A screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Code:

*# A program that uses the camera to detect a red object and draw a bounding box around it.*

import cv2

import numpy as np

from IPython.display import display, Image

import ipywidgets as widgets

import threading

from jetbot import robot

*# This is the camera location that needs to be input into the VideoCapture arugments in order to properly locate the camera and access it.*

camera\_pipeline = "nvarguscamerasrc sensor-id = 0 ! video/x-raw(memory: NVMM), width = 640, height = 480, format = (string)NV12, framerate = (fraction)30/1 ! nvvidconv ! video/x-raw, width = (int)640, height = (int)480, format = (string)BGRx ! videoconvert ! appsink"

*# Initialize the video capture object*

cap = cv2.VideoCapture(camera\_pipeline, cv2.CAP\_GSTREAMER)

*# The main part of the program where the camera rapidly takes images in a loop and displays them.*

def view(button):

    display\_handle = display(None, display\_id=True)  *# Displays the image box.*

    while True:

*# Read the current frame from the camera*

        ret, frame = cap.read()

*# Convert the frame from BGR to RGB color space*

        rgb = cv2.cvtColor(frame, cv2.COLOR\_BGR2RGB)

*# Define the lower and upper bounds for the red color in RGB*

        lower\_red = np.array([112, 0, 0])

        upper\_red = np.array([255, 80, 80])

*# Create a mask for the red color*

        mask = cv2.inRange(rgb, lower\_red, upper\_red)

*# Apply morphological operations to remove noise*

        kernel = cv2.getStructuringElement(cv2.MORPH\_ELLIPSE, (5, 5))

        mask = cv2.morphologyEx(mask, cv2.MORPH\_OPEN, kernel)

        mask = cv2.morphologyEx(mask, cv2.MORPH\_CLOSE, kernel)

*# Find contours of the red objects in the mask*

        contours, \_ = cv2.findContours(

            mask, cv2.RETR\_EXTERNAL, cv2.CHAIN\_APPROX\_SIMPLE)

*# if there are no red objects, spin the robot to the right*

        if len(contours) == 0:

            robot.right(0.4)

        else:

*# Draw bounding boxes around the largest red objects*

            largest\_contour = max(contours, key=cv2.contourArea)

            x, y, w, h = cv2.boundingRect(largest\_contour)

            cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)

*# Converts the information to jpeg to it can be displayed properly.*

            \_, frame = cv2.imencode('.jpeg', frame)

*# Updates the image box to display the updated frame. A conversion to bytes here is needed.*

            display\_handle.update(Image(data=frame.tobytes()))

*# Checks to see if the button was pressed which will stop the camera from functioning.*

        if stopButton.value == True:

            camera.release()

            display\_handle.update(None)

*# This is a simple button that follows most GUI library standards. This button is important because it allows the program to have*

*# a way to stop the video so the camera is still not running after the program has stopped.*

stopButton = widgets.ToggleButton(

    value=False,

    description='Stop',

    disabled=False,

    button\_style='danger',

    tooltip='Description',

    icon='square'

)

*# Displays the stopButton.*

display(stopButton)

*# Connects the stopButton to view.*

thread = threading.Thread(target=view, args=(stopButton,))

thread.start()

Conclusion:

Overall, this program was successful in making the robot spin continuously and stopping when it detected the largest red object. Although there was a significant amount of lag and fps drop in the video capture, the issue was gone when everyone else disconnected from the router. Therefore, the issue could’ve been caused by the limitations of the router’s bandwidth. In terms of test cases, we were short on the number of red objects in the classroom, so we had to resort to pull images from the internet or through MS paint.