MovingObjectDetectionProject

December 30, 2023

[ ]:
```
# Import libraries
import cv2
import time
import imutils

# Initialize the camera
cam = cv2.VideoCapture(0)

# Allow the camera to initialize for 1 second
time.sleep(0)

# Initialize variables
firstframe = None  # Store the first frame
area = 500  # Threshold area for contour detection

# Infinite loop for capturing frames and detecting motion
while True:
    # Read a frame from the camera
    _, img = cam.read()

    # Set initial status text
    text = 'Normal'

    # Resize the frame to have a width of 500 pixels
    img = imutils.resize(img, width=500)

    # Convert the frame to grayscale
    grayimg = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    # Apply Gaussian blur to reduce noise
    gaussianimg = cv2.GaussianBlur(grayimg, (21, 21), 0)

    # If it's the first frame, store it and continue to the next iteration
    if firstframe is None:
        firstframe = gaussianimg
        continue
    # Calculate the absolute difference between the current frame and the first␣
↪frame
    imgdiff = cv2.absdiff(firstframe, gaussianimg)

    # Apply a binary threshold to the difference image
    threshimg = cv2.threshold(imgdiff, 25, 255, cv2.THRESH_BINARY)[1]

    # Dilate the thresholded image to fill gaps and smooth contours
    threshimg = cv2.dilate(threshimg, None, iterations=2)

    # Find contours in the thresholded image
    cnts = cv2.findContours(threshimg.copy(), cv2.RETR_EXTERNAL, cv2.
↪CHAIN_APPROX_SIMPLE)
    cnts = imutils.grab_contours(cnts)

    # Iterate over detected contours
    for c in cnts:
        # Skip small contours
        if cv2.contourArea(c) < area:
            continue

        # Get bounding rectangle coordinates
        (x, y, w, h) = cv2.boundingRect(c)

        # Draw a green rectangle around the detected object
        cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 2)

        # Update status text
        text = 'Moving Object Detected'

    # Print status text
    print(text)

    # Display the annotated frame
    cv2.putText(img, text, (10, 20), cv2.FONT_HERSHEY_COMPLEX, 0.5, (0, 0,␣
↪255), 2)
    cv2.imshow('camerafeed', img)

    # Wait for a key press and check if it's 'q' to break the loop
    key = cv2.waitKey(1) & 0xFF
    if key == ord('q'):
        break

# Release the camera and close OpenCV windows
cam.release()
cv2.destroyAllWindows()
```