Motion Detection & Tracking

Hello, today I’m going to implement using Visual Code and python some motion detection and tracking techniques and algorithm on videos.

I started by making a simple main window where I can find every single option that I will be implementing during this task, here is a brief look of it:

图示

AI 生成的内容可能不正确。

## Background subtraction

First of all i implement a background substraction. Here is the full example code.

def background\_subtractor\_simple(video\_path, window\_name):

**Video initializations**

    cv.namedWindow(window\_name, cv.WINDOW\_NORMAL)

    cv.resizeWindow(window\_name, 800, 600)

    cap = cv.VideoCapture(video\_path)

    if not cap.isOpened():

        messagebox.showerror("Error", f"Could not open video:\n{video\_path}")

        return

    ret, prev\_frame = cap.read()

    if not ret:

        messagebox.showerror("Error", "Could not read first frame")

        return

    prev\_gray = cv.cvtColor(prev\_frame, cv.COLOR\_BGR2GRAY)

    while True:

        ret, frame = cap.read()

        if not ret:

            break

**Apply filters**

        gray = cv.cvtColor(frame, cv.COLOR\_BGR2GRAY)

        diff = cv.absdiff(prev\_gray, gray)

        \_, thresh = cv.threshold(diff, 30, 255, cv.THRESH\_BINARY)

        if cv.getWindowProperty(window\_name, cv.WND\_PROP\_VISIBLE) < 1:

            break

        cv.imshow(window\_name, thresh)

        prev\_gray = gray

        key = cv.waitKey(30) & 0xFF

        if key == ord('q'):

            break

    cap.release()

    cv.destroyWindow(window\_name)

By pressing the option 1 button, it will pop up a video selector, and the technique will be applied to the video selected. The result of the code above is as follows:

电脑萤幕截图

AI 生成的内容可能不正确。

Selecting the video

手机屏幕的截图

AI 生成的内容可能不正确。

Results

## Background subtraction with MOG2 and KNN

We will try different techniques for the background subtraction, this time with MOG2 and KNN, here is the code

#subtractor1 = cv.createBackgroundSubtractorMOG2()

    subtractor2 = cv.createBackgroundSubtractorKNN()

    while True:

        ret, frame = cap.read()

        if not ret:

            break

        #fg\_mask = subtractor1.apply(frame)

        fg\_mask = subtractor2.apply(frame)

        #cv.putText(fg\_mask, "Algorithm: MOG2(Can be changed in the code)", (10, 30),

        #           cv.FONT\_HERSHEY\_SIMPLEX, 1, (255, 255, 255), 2)

        cv.putText(fg\_mask, "Algorithm: KNN(Can be changed in the code)", (10, 30),

                   cv.FONT\_HERSHEY\_SIMPLEX, 1, (255, 255, 255), 2)

        if cv.getWindowProperty(window\_name, cv.WND\_PROP\_VISIBLE) < 1:

            break

        cv.imshow(window\_name, fg\_mask)

As you can see the yellow part is where I applied the mask, cause opencv already has a function for us. Here are the results

文本

AI 生成的内容可能不正确。

手机屏幕截图

AI 生成的内容可能不正确。

## MeanShift/CAMshift

I tried to implement a window where you can select the values, but in python with visual code it is inconvenient, and I ended up with a function that took more than half of the total code:

图形用户界面, 应用程序

AI 生成的内容可能不正确。文本

AI 生成的内容可能不正确。

The function went from line 150 to line 346

文本

AI 生成的内容可能不正确。

So i just decided to use some default values, here is the code and the results:

def meanshift\_tracking(video\_path, window\_name):

    … Rest of the code …

Select the object you want to track

    r = cv.selectROI("Select Object for MeanShift", frame, fromCenter=False, showCrosshair=True)

    cv.destroyWindow("Select Object for MeanShift")

    x, y, w, h = r

    track\_window = (x, y, w, h)

Parameters for the algorithm

    roi = frame[y:y+h, x:x+w]

    hsv\_roi = cv.cvtColor(roi, cv.COLOR\_BGR2HSV)

    mask = cv.inRange(hsv\_roi, np.array((0., 60., 32.)), np.array((180., 255., 255.)))

    roi\_hist = cv.calcHist([hsv\_roi], [0], mask, [180], [0, 180])

    cv.normalize(roi\_hist, roi\_hist, 0, 255, cv.NORM\_MINMAX)

    term\_crit = (cv.TERM\_CRITERIA\_EPS | cv.TERM\_CRITERIA\_COUNT, 10, 1)

    cv.namedWindow(window\_name, cv.WINDOW\_NORMAL)

    cv.resizeWindow(window\_name, 800, 600)

    while True:

        ret, frame = cap.read()

        if not ret:

            break

        hsv = cv.cvtColor(frame, cv.COLOR\_BGR2HSV)

        dst = cv.calcBackProject([hsv], [0], roi\_hist, [0, 180], 1)

        ret, track\_window = cv.meanShift(dst, track\_window, term\_crit)

        x, y, w, h = track\_window

        result = cv.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)

        … Rest of the code …

    cap.release()

    cv.destroyWindow(window\_name)

路旁的草地上

AI 生成的内容可能不正确。

路旁的草地上

AI 生成的内容可能不正确。

路上有辆绿色的车

AI 生成的内容可能不正确。

In this sample video, the algorithm will follow every black color car that appears in the video

The same way i implemented CAMShift by changing one line and the parameters:

def camshift\_tracking(video\_path, window\_name):

… Rest of the code …

    while True:

        ret, frame = cap.read()

        if not ret:

            break

        hsv = cv.cvtColor(frame, cv.COLOR\_BGR2HSV)

        dst = cv.calcBackProject([hsv], [0], roi\_hist, [0, 180], 1)

        # Apply CAMShift to get the rotated rectangle

        ret, track\_window = cv.CamShift(dst, track\_window, term\_crit)

        … Rest of the code …

    cap.release()

    cv.destroyWindow(window\_name)

Results:

路上有辆绿色的车

AI 生成的内容可能不正确。