



Homework11 Object tracking

胡成成 2101210578

Question

三选一：

1. 在实时输入视频中跟踪物体；
2. 在输入视频文件中跟踪物体；
3. 将物体检测和物体跟踪结合起来，实现不需要初始标注物体的跟踪系统。

Answer

- 选择作业二
- 代码：

```
import cv2
import sys
import numpy as np

minor_ver = 4
boxes = [];
windowName = "Condensation Tracking"
current_mouse_position = np.ones(2, dtype=np.int32);
selected = False

def on_mouse(event, x, y, flags, params):

    global boxes;
    global selection_in_progress;
    global selected;

    current_mouse_position[0] = x;
    current_mouse_position[1] = y;

    if event == cv2.EVENT_LBUTTONDOWN:
        boxes = [];
        sbox = [x, y];
        selection_in_progress = True;
        boxes.append(sbox);

    elif event == cv2.EVENT_LBUTTONUP:
        ebox = [x, y];
        selection_in_progress = False;
        selected = True
        boxes.append(ebox);

#
def center(points):
    x = np.float32((points[0][0] + points[1][0] + points[2][0] + points[3][0]) / 4.0)
    y = np.float32((points[0][1] + points[1][1] + points[2][1] + points[3][1]) / 4.0)
    return np.array([np.float32(x), np.float32(y)], np.float32)
#

def nothing(x):
```

```

pass

#
def drawCross(img, center, color, d):
    #On error change cv2.CV_AA for cv2.LINE_AA
    # (for different versions of OpenCV)
    cv2.line(img, (center[0] - d, center[1] - d), \
              (center[0] + d, center[1] + d), color, 2, cv2.LINE_AA, 0)
    cv2.line(img, (center[0] + d, center[1] - d), \
              (center[0] - d, center[1] + d), color, 2, cv2.LINE_AA, 0)

#

if __name__ == '__main__':

    cv2.namedWindow(windowName, cv2.WINDOW_NORMAL);
    # Set up Callback.
    # Instead of MIL, you can also use
    cv2.setMouseCallback(windowName, on_mouse, 0);
    cropped = False;
    tracker_types = ['MIL', 'KCF', 'GOTURN', 'CSRT']
    # 0,1,2,3 tested OK
    # set up tracer
    for i in range(minor_ver):
        tracker_type = tracker_types[i]
        if tracker_type == 'MIL':
            print("Creating MIL tracker")
            tracker = cv2.TrackerMIL_create()
        if tracker_type == 'KCF':
            print("Creating KCF tracker")
            tracker = cv2.TrackerKCF_create()
        if tracker_type == 'GOTURN':
            print("Creating GOTURN tracker")
            tracker = cv2.TrackerGOTURN_create()
        if tracker_type == "CSRT":
            print("Creating CSRT tracker")
            tracker = cv2.TrackerCSRT_create()

    # Read video
    # video = cv2.VideoCapture("../video/surv.mp4")
    video = cv2.VideoCapture("./way.mp4")
    if not video.isOpened():
        print("Could not open video")
        sys.exit()
    while(1):
        # Read first frame.
        ok, frame = video.read()
        if not ok:
            print('Cannot read video file')
            sys.exit()
        # get bbox
        #cv2.imshow(windowName,frame)
        if(selected == True):
            bbox = (boxes[0][0],boxes[0][1],current_mouse_position[0]-boxes[0][0],current_mouse_position[1]-boxes[0][1])
            top_left = (boxes[0][0], boxes[0][1]);
            bottom_right = (current_mouse_position[0], current_mouse_position[1]);
            print(top_left,bottom_right)
            cv2.rectangle(frame,top_left, bottom_right, (0,255,0), 2);
            cv2.waitKey(30)
            break;
        cv2.imshow(windowName,frame)
        cv2.waitKey(30)
        # break
    #initialization
    ok, frame = video.read()
    #bbox = (276, 23, 86, 320)
    ok = tracker.init(frame, bbox)
    print("First frame initialization completed")

    while True:
        # Read a new frame
        ok, frame = video.read()
        if not ok:

```

```

        print("EOF reached")
        break
    # Update tracker
    # Start timer
    timer = cv2.getTickCount()
    ok, bbox = tracker.update(frame)
    fps = cv2.getTickFrequency() / (cv2.getTickCount() - timer);
    # Draw bounding box
    if ok:
        # Tracking success
        p1 = (int(bbox[0]), int(bbox[1]))
        p2 = (int(bbox[0] + bbox[2]), int(bbox[1] + bbox[3]))
        cv2.rectangle(frame, p1, p2, (255,0,0), 2, 1)
    else:
        # Tracking failure
        cv2.putText(frame, "Tracking failure detected", (100,80), cv2.FONT_HERSHEY_SIMPLEX, 0.75, (0,0,255),2)
    # Display tracker type on frame
    cv2.putText(frame, tracker_type + " Tracker", (100,20), cv2.FONT_HERSHEY_SIMPLEX, 0.75, (50,170,50),2);
    # Display FPS on frame
    cv2.putText(frame, "FPS : " + str(int(fps)), (100,50), cv2.FONT_HERSHEY_SIMPLEX, 0.75, (50,170,50), 2);
    # Display result
    cv2.imshow("tracking", frame)
    # Press Q on keyboard to exit
    if cv2.waitKey(25) & 0xFF == ord('q'):
        break
cv2.waitKey(0)
tracker.release()
video.release()
cv2.destroyAllWindows()

print("Loop left")
#video.release()
# Closes all the frames
cv2.destroyAllWindows()

```

- 运行结果：





- 总结：目标跟踪的直线运行物体的效果很明显，但是在拐弯的时候会产生一定的偏差。