limhpone / computervision-final-prep

Code Issues Pull requests Actions Projects Wiki Security In

computervision-final-prep / lab / Lab 09 (Tracking)-20251128 / 9.2_single_object_tracking.py

limhpone lab 9-obj tracking 267755b · 2 hours ago

91 lines (70 loc) · 2.89 KB

Code Blame Raw

```python
import cv2
import sys

(major_ver, minor_ver, subminor_ver) = (cv2.__version__).split('.')

# For GOTURN, you need to set up the model files.
# Download the following files: https://github.com/Mogball/goturn-files

if __name__ == '__main__' :

    # Set up tracker.
    # Instead of GOTURN, you can also use

    tracker_types = ['BOOSTING', 'MIL','KCF', 'GOTURN']
    tracker_type = tracker_types[3]

    if int(minor_ver) < 3:
        tracker = cv2.Tracker_create(tracker_type)
    else:
        if tracker_type == 'BOOSTING':
            tracker = cv2.TrackerBoosting_create()
        elif tracker_type == 'MIL':
            tracker = cv2.TrackerMIL_create()
        elif tracker_type == 'KCF':
            tracker = cv2.TrackerKCF_create()
        elif tracker_type == 'GOTURN':
            tracker = cv2.TrackerGOTURN_create()

    # Read video
    # video = cv2.VideoCapture("videos/video.mp4")
    video = cv2.VideoCapture(0)                    # for using CAM

    # Exit if video not opened.
    if not video.isOpened():
        print("Could not open video")
        sys.exit()

    # Read first frame.
    ok, frame = video.read()
    if not ok:
        print ('Cannot read video file')
        sys.exit()

    # Define an initial bounding box - if you already know the object location, you can set i
    # bbox = (287, 23, 86, 320)  # Example: (x, y, width, height)

    # Select ROI manually
    bbox = cv2.selectROI(frame, False)

    # Initialize tracker with first frame and bounding box
    ok = tracker.init(frame, bbox)

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

        # Start timer
        timer = cv2.getTickCount()

        # Update tracker
        ok, bbox = tracker.update(frame)

        # Calculate Frames per second (FPS)
        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_SIMPLE

        # Display tracker type on frame
        cv2.putText(frame, tracker_type + " Tracker", (100,20), cv2.FONT_HERSHEY_SIMPLEX, 0.7

        # Display FPS on frame
        cv2.putText(frame, "FPS : " + str(int(fps)), (100,50), cv2.FONT_HERSHEY_SIMPLEX, 0.75

        # Display result
        cv2.imshow("Tracking", frame)

        # Exit if ESC pressed
        if cv2.waitKey(1) & 0xFF == ord('q'): # if press SPACE bar
            break
    video.release()
    cv2.destroyAllWindows()
```