computervision-final-prep / lab / Lab 09 (Tracking)-20251128 / 9.3_multi_object_tracking.py

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

91 lines (73 loc) · 2.87 KB

Code Blame Raw

```python
# import the necessary packages
import time
import cv2
import sys

# initialize a dictionary that maps strings to their corresponding
# OpenCV object tracker implementations
print(cv2.__version__)

# choose a tracker type you want to use for each ROI you add
tracker_type = "KCF"  # or "MIL"

# pick legacy factories when available
if hasattr(cv2, "legacy"):
    FACTORIES = {
        "KCF":   cv2.legacy.TrackerKCF_create,
        "MIL":   cv2.legacy.TrackerMIL_create,
    }
    trackers = cv2.legacy.MultiTracker_create()
else:
    FACTORIES = {
        "KCF":   getattr(cv2, "TrackerKCF_create", None),
        "MIL":   getattr(cv2, "TrackerMIL_create", None),
    }
    trackers = cv2.MultiTracker_create()

tracker_factory = FACTORIES.get(tracker_type)

if tracker_factory is None:
    print("Tracker", tracker_type, "not available in this build")
    sys.exit(1)

```

```python
# initialize OpenCV's special multi-object tracker
trackers = cv2.legacy.MultiTracker_create()

# if a video path was not supplied, grab the reference to the web cam
video = cv2.VideoCapture('video.mp4')
if not video.isOpened():
    print("Could not open video")
    sys.exit()

# loop over frames from the video stream
while True:
    # grab the current frame, then handle if we are using a
    # VideoStream or VideoCapture object
    ok, frame = video.read()

    # check to see if we have reached the end of the stream
    if ok is None or frame is None:
        break

    # resize the frame (so we can process it faster)
    frame = cv2.resize(frame, (720, 640))

    # grab the updated bounding box coordinates (if any) for each
    # object that is being tracked
    success, boxes = trackers.update(frame)

    # loop over the bounding boxes and draw then on the frame
    if success:
        for b in boxes:
            x, y, w, h = map(int, b)
            cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 0, 0), 2)
    else:
        cv2.putText(frame, "Tracking failure detected", (100, 80),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.75, (0, 0, 255), 2)

    cv2.imshow("Frame", frame)
    key = cv2.waitKey(1) & 0xFF

    # if the 's' key is selected, we are going to "select" a bounding box to track
    if key == ord("s"):
        # select the bounding box of the object we want to track
        # (make sure you press ENTER or SPACE after selecting the ROI)
        box = cv2.selectROI("Frame", frame, fromCenter=False,
                            showCrosshair=True)

        # create a new object tracker for the bounding box and add it
        # to our multi-object tracker
        tr = tracker_factory()
        trackers.add(tr, frame, box)
```

```
82
83          # if the `q` key was pressed, break from the loop
84          elif key == ord("q"):
85              break
86
87      # if we are using a webcam, release the pointer
88      video.release()
89
90      # close all windows
91      cv2.destroyAllWindows()
```