

! This class has been made inactive. No posts will be allowed until an instructor reactivates the class.

note

111 views

HW3: Detection and Tracking

Your goal is to:

- Detect the face in the first frame of the movie
 - Using pre-trained Viola-Jones detector
- Track the face throughout the movie using:
 - CAMShift
 - Particle Filter
 - Face detector + Kalman Filter (always run the `kf.predict()`, and run `kf.correct()` when you get a new face detection)

Bonus (20pt): Face Detector + Optical Flow tracker (use the OF tracker whenever the face detector fails).

Due: Thu 10/19 9am

Skeleton and bootstrap code is provided.

[HW3DetectionTracking.zip](#)

We provide skeleton code to help detect faces, and build your tracker.

The code is bundled in the zip, see the file named `detection_tracking.py`.

To detect a face we have a convenience function:

```
x,y,w,h = detect_one_face(frame) # rectangle of the face, or (0,0,0,0) if no face found
```

With the tracker skeleton you can basically just plug in the right tracker to get your functionality.

This is a skeleton of a tracker:

```
def skeleton_tracker():
    # read video file
    v = cv2.VideoCapture("input.avi")

    output = open("output.txt", "w")

    # read first frame
    ret, frame = v.read()
    if ret == False:
        return

    # detect face in first frame
    c,r,w,h = detect_one_face(frame)

    # set the initial tracking window
    track_window = (c,r,w,h)

    # calculate the HSV histogram in the window
    # NOTE: you do not need this in the Kalman, Particle or OF trackers
    roi_hist = hsv_histogram_for_window(frame, (c,r,w,h)) # this is provided for you

    # initialize the tracker
    # e.g. kf = cv2.KalmanFilter(4,2,0)
    # or: particles = np.ones((n_particles, 2), int) * initial_pos
    # see further explanation below

    while(1):
        ret, frame = v.read() # read another frame
        if ret == False:
            break

        # perform the tracking
        # e.g. cv2.meanShift, cv2.CamShift, or kalman.predict(), kalman.correct()

        # use the tracking result to get the tracking point (pt):
        # if you track a rect (e.g. face detector) take the mid point,
        # if you track particles - take the weighted average
        # the Kalman filter already has the tracking point in the state vector

        # write the result to the output file
        output.write("%d,%d\n" % pt)

    output.close()
```

OpenCV functions/objects to look into:

```
cv2.CamShift
cv2.KalmanFilter
cv2.calcOpticalFlowPyrLK
cv2.calcBackProject
```

Kalman Filter

To save you the trouble of figuring it out, here's how to initialize the KF

```
state = np.array([c+w/2,r+h/2,0,0], dtype='float64') # initial position
kalman = cv2.KalmanFilter(4,2,0) # 4 state/hidden, 2 measurement, 0 control
kalman.transitionMatrix = np.array([[1., 0., .1, 0.], # a rudimentary constant speed model:
                                     [0., 1., 0., .1], # x_t+1 = x_t + v_t
                                     [0., 0., 1., 0.],
                                     [0., 0., 0., 1.]])
kalman.measurementMatrix = 1. * np.eye(2, 4) # you can tweak these to make the tracker
kalman.processNoiseCov = 1e-5 * np.eye(4, 4) # respond faster to change and be less smooth
kalman.measurementNoiseCov = 1e-3 * np.eye(2, 2)
kalman.errorCovPost = 1e-1 * np.eye(4, 4)
kalman.statePost = state
```

During your KF tracking you will need to do something like the following:

```
prediction = kalman.predict()

# ...
# obtain measurement

if measurement_valid: # e.g. face found
    # ...
    posterior = kalman.correct(measurement)

# use prediction or posterior as your tracking result
```

Particle Filter

Here's how I initialize the PF with 200 particles:

```
# a function that, given a particle position, will return the particle's "fitness"
def particleevaluator(back_proj, particle):
    return back_proj[particle[1],particle[0]]

# hist_bp: obtain using cv2.calcBackProject and the HSV histogram
# c,r,w,h: obtain using detect_one_face()
n_particles = 200

init_pos = np.array([c + w/2.0,r + h/2.0], int) # Initial position
particles = np.ones((n_particles, 2), int) * init_pos # Init particles to init position
f0 = particleevaluator(hist_bp, pos) * np.ones(n_particles) # Evaluate appearance model
weights = np.ones(n_particles) / n_particles # weights are uniform (at first)
```

During the PF tracking you will need to do something along the lines of

```
# Particle motion model: uniform step (TODO: find a better motion model)
np.add(particles, np.random.uniform(-stepsize, stepsize, particles.shape), out=particles, casting="unsafe")

# Clip out-of-bounds particles
particles = particles.clip(np.zeros(2), np.array((im_w,im_h))-1).astype(int)

f = particleevaluator(hist_bp, particles.T) # Evaluate particles
weights = np.float32(f.clip(1)) # Weight ~ histogram response
weights /= np.sum(weights) # Normalize w
pos = np.sum(particles.T * weights, axis=1).astype(int) # expected position: weighted average

if 1. / np.sum(weights**2) < n_particles / 2.: # If particle cloud degenerate:
    particles = particles[resample(weights),:] # Resample particles according to weights
# resample() function is provided for you
```

Submission details

For each frame write out the tracking results to a text file, in the format specified below.

```
0,x_0,y_0
1,x_1,y_1
2,x_2,y_2
...
256,x_n,y_n
```

Plain text file with frame_index and x,y coordinates of the middle of the face/head for each frame in the movie (no space between).

Note: you also need to write the initial first position, i.e. the first frame face detection (a total of 257 frames). Make sure you have the correct number of frames in your .txt file.

Write 3 (or 4) outputs for each condition:

- output_camshift.txt
- output_particle.txt
- output_kalman.txt
- Bonus: output_of.txt

Put your results in a zip with your id as the filename.
Results are checked automatically, so pay extra attention to file naming conventions as well as formats.

// ===== //

An example folder structure for HW3 is uploaded and available under "Resources": Bilbo_Baggins_111123456.zip
It is recommended to download that file, replace the original files with your files, change the name and submit it.

The grades for HW3 will be calculated based on the difference between your tracking results and ground truth results.
We will release the ground truth and the function computing your grades after the deadline of HW3. A visually correct tracking result should be sufficient for a good mark.

Please keep the interface of the program intact and we will test your program under VM (make sure to get rid of all unnecessary dependencies). If the generated results does not match the one in "Result" folder, you will receive 0 mark for that question.

It is critical to have the correct number of frames in your results.

#pin

hw3

Updated 2 months ago by Roy Shilkrot and Fan Wang

followup discussions for lingering questions and comments

☒ Resolved ☐ Unresolved



Anonymous 2 months ago How about doing it with C++?
Could I bother you to give me a framework?
Thank you very much.



Fan Wang 2 months ago We have only one c++ submission out of 100 last time. Please consider using python. It would take a few days to prepare a c++ script.



Anonymous 2 months ago Sorry, if the C++ is forbidden, could Professor give us a previously reminder before the class?



Roy Shilkrot 2 months ago C++ is still allowed. We simply don't have skeleton code for it.
The Py codes translate very literally to C++, as the OpenCV functions are virtually the same. The Numpy functions need translation, which in many cases is a simple OpenCV operator.
Fan said you should **consider** switching to Py, but you are welcome to submit in C++ all the same.

☒ Resolved ☐ Unresolved



Anonymous 2 months ago
Professor,
is it possible to postponed the due date to 10/19 Thr,
since it comes out at this moment which we have exams just around the corner.

I believe most of us don't really have time to deal with it these days.



Anonymous 2 months ago +1 Please !



Roy Shilkrot 2 months ago Submission deadline is duly extended to Thu 10/19.

☒ Resolved ☐ Unresolved



Anonymous 2 months ago
Professor,

Can we please have 1 class extension for this HW deadline? HW was posted Tuesday very late night and it is due on next Tuesday morning. You mentioned in class that we would have about 1 and half week for this assignment. But we don't have one full week also to complete the assignment. Plus many of us have midterms in this/next week in other subjects as well apart of CV midterm.

Can you please consider 1 class extension and shift deadline Thursday morning instead of Tuesday morning?

Thanks



Roy Shilkrot 2 months ago Submission deadline is duly extended to Thu 10/19.

☒ Resolved ☐ Unresolved



Xuan Li 2 months ago
Can we define four tracker functions with the same skeleton you provided and call them as follows:

```

if (question_number == 1):
    camshift_tracker(video, "output_camshift.txt")
elif (question_number == 2):
    particle_tracker(video, "output_particle.txt")
elif (question_number == 3):
    kalman_tracker(video, "output_kalman.txt")
elif (question_number == 4):
    of_tracker(video, "output_of.txt")

```



Fan Wang 2 months ago Sure, as long as you keep an uniform interface so I can run your program on VM

☒ Resolved ☐ Unresolved



Anonymous 2 months ago
I am using mac.

This command `cv2.VideoCapture("input.avi")` is not working.

I searched on net for some leads.

Some sources say to install perian and do

```

<code>brew uninstall opencv
brew install opencv --with-ffmpeg -v</code>

```

still this is not solving the issue.

Can someone please help me with this.

I have struggled a lot since the beginning - install virtualenv, then xfeatures2d was not working in previous assignment and now this. This is so annoying.



Anonymous 2 months ago Post continued...

I found a way to go around this problem...
After installing perian, I can convert the .avi to .mov
This .mov is working.

Can I use this converted .mov file ?
Will this have any problem or have different results?



Anonymous 2 months ago I am using VLC player. It plays all kinds of format in mac, even .avi.



Anonymous 2 months ago what does VLC has to do anything with `cv2.VideoCapture` ??

☒ Resolved ☐ Unresolved



Anonymous 2 months ago
Question about Optical flow output

Should we return the result from the optical flow only in case of measurement failure?
i.e return the measurement out output if its fine

Pseudocode:

```

if( invalid measurement ):
    pt = getOpticalFlowEstimate()
else
    pt = measurement

```

return pt



Anonymous 2 months ago Check @150

☒ Resolved ☐ Unresolved



Anonymous 2 months ago
"If the generated results does not match the one in "Result" folder, you will receive 0 mark for that question." In many places we're using random functions, which means for each run output values are different. So above rule doesn't hold.



Alex Scarlatos 2 months ago +1



Anonymous 2 months ago +1



Fan Wang 2 months ago In that case, a threshold will be applied to determine if they match.

☒ Resolved ☐ Unresolved



Timothy Zhang 2 months ago
Can we please have HW3 available for submission on Blackboard.



Fan Wang 2 months ago Working on it.

☒ Resolved ☐ Unresolved



Anurag Arora 2 months ago
Hi,

In the submission details above, it is mentioned that output should look like this:
0,x_0,y_0

But in the "example.txt" file in Bilbo_Baggins_111123456 folder, the output is in the below format:
0 56 23
1 12 42
3 ...
4 ...

So can you please confirm once regarding the correct output format?



Anonymous 2 months ago opps...



Anonymous 2 months ago Please confirm



Fan Wang 2 months ago Please use 0,x_0,y_0 format

☒ Resolved ☐ Unresolved



Anonymous 2 months ago
The assignment states that the OF is of 20 bonus points but Blackboard shows a total of 112 points. Is the bonus 12 points or 20 ?



Roy Shilkrot 2 months ago 20.

☒ Resolved ☐ Unresolved



Anonymous 2 months ago

If the generated results does not match the one in "Result" folder, you will receive 0 mark for that question.

Does this rule also hold for particle filter? Because the particle filter is random, it seems to be impossible to generate the same result.



Roy Shilkrot 2 months ago the PF is not random. it may use a random motion model, but still track the object very well.



Anonymous 2 months ago My particle filter is tracking the object properly. But on running it multiple times and plotting the values obtained I see some small difference between the values in different runs.



Roy Shilkrot 2 months ago small changes are likely to be insignificant.

☒ Resolved ☐ Unresolved



Anonymous 2 months ago Hi Prof,
regarding the last requirement of the submitted code generating exact output: There will be cases where it wont match exactly with the submitted output, owing to the randomness of how the samples were initially selected

would that be a problem?