CS 378 Computer Vision
Problem set 5
Out: Tuesday, Nov 24
Due: Friday, Dec 4, 11:59 PM
With automatic extension until Tuesday Dec 8, 11:59 PM

For this assignment, you may choose between answering the short answer questions (Section I) *or* doing the programming problem (Section II). (100 points = full credit.) The extra credit portions (Section III) expand on the programming problem.

Section I: Short answer questions: Recognition and Motion [100 points]

1. Briefly explain two distinct limitations of the sliding window-based approach to object detection.

2. In a *visual vocabulary tree* (or a pyramid of bins in the descriptor space), how does the appearance of the local image patches assigned to a given child node (bin) relate to the appearance of the local image patches assigned to that node's (bin's) parent?

3. How do the objective functions for the SVM and Boosting classification algorithms differ?

4. You are asked to write an algorithm to automatically place logos (each of which is contained in a small rectangular image) into another arbitrary larger image. The position where a logo is inserted should be such that it is far from the "interesting" parts of the image. That is, it should fall far from regions of high contrast. Design an algorithm that exploits the *distance transform* in order to do this task. Write pseudo-code to describe the steps.

5. What *constraints* allow a dense solution for optical flow using spatial and temporal image derivatives? Briefly explain.

6. A linear dynamics model is defined according to two key distributions: the systems dynamics model, and the measurement model. The dynamics affect the distribution over the current state given the previous state, as given by $\mathbf{x}_t \sim N(\mathbf{D}\mathbf{x}_{t-1}; \Sigma_d)$. The measurement model affects the distribution over the measurement given the current state, as given by $\mathbf{y}_t \sim N(\mathbf{M}\mathbf{x}_t; \Sigma_m)$. Suppose we are modeling a situation where the state is completely observable, though we are still influenced by noise. How should this be encoded through the above variable(s)?

7. Suppose we are using a Kalman filter and linear dynamics model to track a local feature point. Explain how the *variance* of the Gaussian distribution representing the predicted state can be used to guide the search for the feature in the next frame of video.

Section II: Track a single object [100 points]

Tracking in video

For this program, the goal is to track moving objects in video using a feature-based matching approach. For the main assignment, the program should follow a single feature on some moving object, performing the matching step with normalized cross-correlation.

For extra credit, you can optionally generalize this to track multiple points per frame, and to allow the introduction of new moving objects into the scene, or add a Kalman filter (see below).

Use the video data provided on the class website. There are two sequences, each with 50 frames. Choose one sequence to show results for Section I.

1. Use the `harrisDetector.m` function from Pset4 to perform corner detection in the first frame. Manually select a detected corner on one of the objects that will move within the video.

2. Extract a patch around the single selected interest point, and use normalized cross-correlation (`>>help normxcorr2`) to track the point in the subsequent frames. In each frame, limit the search to a search window surrounding the previously detected position, and continually replace the template (patch) with the most recent detection.

3. Show the tracking result by drawing (plotting) the detected position across the frames.

4. Test and briefly describe the tradeoff of the patch and window size parameters.

Section III: [OPTIONAL] Extra credit: [30 points max]

1. [20 points] Consider the most salient corner points (high *v* values exceeding a threshold) as potential points to track per frame. Search for them in subsequent frames as above. Display the tracked points with different colors on the output to help distinguish them visually. In any frame after the initial frame, if a new corner appears that is not currently in a tracked region, start a new track and add it to the list of currently tracked points. Explain the correct and incorrect tracks found by the system: what worked, and where and why does it fail?

2. [10 points] Perform background subtraction, and show the foreground blobs over the entire sequence. Use feature(s) within each blob to assign a motion vector to each tracked object (blob). Briefly explain how results compare to your sparse feature-point tracker.

3. [20 points] Implement a Kalman filter (e.g., with a constant velocity assumption), and show cases where it can improve your tracking estimates. You may need to introduce some synthetic image noise to make the matching step yield poorer measurements.

Tips:

While debugging, process only a short portion of the video.

Read the demo code provided in Matlab's help for `normxcorr2` carefully to see how the indices are given.

While debugging, it may be useful to display the patch region and search window regions. The function `rectangle` can be used to draw a box on top of the image at the given position and size.

See `>>help avifile` to convert a sequence of frames shown in a figure into an avi movie file, in order to save your results as a movie (optional).

Acknowledgement

The video used for this problem is courtesy of Prof. H.-H. Nagel's group at Karlsruhe University: http://i21www.ira.uka.de/image_sequences/.

Submission instructions: what to hand in

<u>Electronically:</u>

- Your well-documented Matlab code .m files.
- A pdf file containing:
    - Your name and CS login ID at the top.
    - Your answers for Section I **OR** your image results and accompanying explanations for the program in Section II.
    - (optional): any results and descriptions for extra credit portions in Section III.

Submit all the above with one call to turnin:

```
>> turnin --submit jaechul pset5 pset5.pdf codeFileXYZ.m
codeFileABC.m etc.
```

<u>Hardcopy:</u> Print out the pdf file, and either bring it to class on Thurs 12/3, or drop off at Jaechul's office in CSA by Tues 12/8. Do not print out code. As usual, the hardcopy must be the same as what is submitted electronically.