CS 378 Computer Vision
Problem set 4
Out: Tuesday, Nov 25
Due: Thursday, Dec 4
With automatic extension until Tuesday Dec 9, 11:59 PM

Tracking in video

For this problem set, the goal is to track moving objects in video using a feature-based matching approach. The main implementation is divided into two stages. The first part (A) follows a single feature on some moving object, performing the matching step with normalized cross-correlation. The second part (B) generalizes this to track multiple points per frame, and to allow the introduction of new moving objects into the scene. Use the video data provided on the class website.

Part A: [100 points]

1. Use the provided `harris.m` function 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 describe the tradeoff of the patch and window size parameters.

Part B: [25 points]

1. 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.

    - To save computation, tracks that are stationary for some number of frames may be discarded from the updates, if we assume that all objects of interest will either initially be moving, or else will later enter the scene and immediately move. Only show moving object tracks in the final output display.

    - Detect when a good match cannot be found for a patch (e.g., due to the moving object being occluded, or leaving the scene) by setting a threshold on the correlation score. End a track trajectory in the output display if this threshold is not met (and remove the point from the list of tracked features).

2. Explain the correct and incorrect tracks found by the system: what worked, and where and why does it fail?

OPTIONAL: Extra credit [up to 25 points max]

If you choose to implement either of these, include in your submission a description of your approach, and also display and explain the output results you obtain.

1. 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).

2. 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.

Submission instructions: what to hand in

Create a tar file with

- Your documented Matlab code. Include your name at the top of each file in a comment.
- A single pdf file containing the following
    - Your name at the top
    - Your responses and image results for each question, numbered. Insert image figures in the appropriate places.
    - (optional): any results and descriptions for extra credit portions. For max credit explain what you implemented and also interpret the results.

1. Name the tar file with <username>_pset4.tar and submit via the turnin program: turnin --submit harshd pset4 <filename>

2. Also submit a hardcopy of the pdf file. (Be conscious of your figure layouts so as to conserve paper!)

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).

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/.