Advanced Vision Practical 1

Bob Fisher
School of Informatics

January 2016

Abstract

This describes the first assignment for assessment on Advanced Vision. The main goal is to reliably detect and track several interacting individuals, as observed in a video acquired from an overhead camera. The assignment is due: **4pm Thursday 11 February**. You must do this practical in teams of 2, and submit 1 PDF report only. There will also be an assessed live demonstration of your practical.

Task Background

At the URL:

`http://homepages.inf.ed.ac.uk/rbf/AVDATA/AV116DATA/`

you will find a file `frameset1.tar` of 210 consecutive images (frameX.jpg, X=110-319) plus a background image `bgframe.jpg`. The images show a sequence of images of 4 people dancing, as seen by an overhead camera. The camera is essentially stationary viewing a stationary background. The camera data is recorded at about 9 frames/second, but occasionally there are larger gaps between frames due to network loads, etc. Ignore the person who is standing at approximately pixel (row,column) = (150,223) who is 'calling' the dance.

The tar file also contains a file (`positions1.mat`) listing the 'ground-truth' position of the people. These are recorded in a data structure:

```
positions(4,NUMFRAMES,2)
```

where `NUMFRAMES=210` and `positions(p,f,:)` is the (row,column) position of person `p` seen in frame `f`.

Here is the background image (`bgframe.jpg`) and a typical frame with the people to detected:

The overall task for this assignment is to detect and track the people, even though they are interacting (and may collide). To do this you need to write a set of programs that can:

1. Detect the moving people,
2. Compute a trajectory for each person through all frames (taking account of interactions), and
3. Evaluate the correctness of the detections and tracking against the ground truth dataset.

Each of these is described in more detail below.

Person Detector

The image data is a set of RGB colour images and the camera is fixed. There is a constant largely featureless background image and the lighting is mostly constant. There are some shadows from objects in the room and the people also create new moving shadows. You should be able to easily detect the people using a background subtraction method. You might find using normalised RGB helps to remove the effect of the shadows, but remember that this normalisation also makes white, grey and black have the same normalised RGB values. So, you might need to use both the normalised and unnormalised images, and then apply some reasoning based on what you know about the people.

Use operations like open/close or dilate/erode to clean up the image. Use the largest connected components.

Label each person uniquely and try to keep the same numerical label throughout.

Represent each person by their centre of mass and draw a circle whose area is equal to the area of the detected person.

Note: the people will be interacting and so a detected region might consist of more than one person. Think about how to determine when this happens and what to do to separate the individual people.

Tracking

Determine the correspondence between the people in each image. Some people will not move much between consecutive frames, but others will.

The information that can most easily be exploited is:

1. The RGB colour distribution of each person is nearly constant. You might represent each person by a 2D histogram of the r/g components from normalised RGB values. Comparing histograms could use the Bhattacharyya distance,
2. There are 4 people always in the field of view.

The Condensation tracker discussed in the lecture video is potentially a good way to solve the tracking problem, because it can keep different hypotheses for each of the people, which might be very helpful when considering potentially ambiguous tracking correspondence hypotheses. While one could consider a combined state vector for all 4 people, it is simpler to have one set of state vectors (and thus one condensation tracker) for each person being tracked. The example presented in lecture can be adapted for each person. A new tracker can be started for each new detected person. The person's size might help resolve ambiguities.

You'll also need the Condensation tracker to cope with collisions, This will require an extension of the state transition diagram to include the collision events and estimate new transition probabilities.

On the other hand, a Kalman filter is probably not necessary for this problem.

Evaluation

The ground truth file shows the person centres as found by hand, for each person. Compare your estimated person centre (use the centre of mass of the detected pixels) with the ground truth centre for each person and each frame (4 people * 210 frames comparisons in total). Report the number of people that were detected within 10 pixels of any ground truth centre. Report the mean distance between the ground truth and estimated centres, for all people within the 10 pixel distance threshold. Report the number of false person detections.

As part of the assignment, plot your person detections on top of each frame by drawing a circle of the estimated radius ($\sqrt{Area/\pi}$) around the detected position. Draw the centre of the circle. Use a different colour for each trajectory. Include a few of these images in your report.

We also want to evaluate the quality of the tracker.

Compute the percentage of the instances where your tracker correctly pairs the people between consecutive frames. Ie. suppose person a is found at positions $\vec{p}_t$ and $\vec{p}_{t+1}$ in frames t and $t+1$. If your program detected these 2 people, then a correct trajectory pairing means that your program should have assigned them the same trajectory identifier. If they don't have the same identifier, then this is a tracking error. Report the number of correct and erroneous pairings.

Report the percentage of person detections that belong to trajectories with only 1 detection.

The final tracking evaluation is done by hand: Draw an image of all ground-truth trajectories on top of an image frame, where each trajectory has a different colour. Then, draw a similar image showing each of your trajectories on another image, again with different colours for each trajectory. Finally, report the number of tracked trajectories for each of the 4 people (these can be counted by hand from the 2 images). Include the images in your report.

Your Report

Each team writes a single report that describes:

- The algorithms that you used for each stage of the process.
- How well the algorithms performed on the supplied test data. Show the statistical results and images requested above.
- Show example images of each processing stage, including a few examples of detected people.
- Show examples of successful and unsuccessful detections, and trajectory connection.
- Discussion on performance: successes and failures, causes of failures and potential remedies.
- Your code. Do not include code that was downloaded from the AV or IVR web sites.

Other Comments

1. You can use the lecture example code from:
   `http://www.inf.ed.ac.uk/teaching/courses/av/MATLAB/`
2. Because there are a limited number of MATLAB Image Processing library licenses available, use alternative MATLAB functions from
   `http://www.inf.ed.ac.uk/teaching/courses/av/MATLAB/UTILITIES/`

Assignment Submission

Submit your report in PDF online by 4pm Thursday February 11. The online submission line is:

```
submit av 1 FILE
```

where `FILE` is the name of your PDF file.

Live Demonstration

There will also be a demonstration session assigned between 9:00-13:00 on Friday February 12, where you will have to demonstrate your code on a new dataset (taken also at the same time as the other images, including the same background). We'll email you about the location and schedule.

You will need your matlab program to show:

1. The detected people for each frame overlaid over the original images using the centre-point and circle method described above. Use a 1 second pause between images.
2. The current tracked trajectories of each person in the current frame, overlaid on the first image (use a different coloured track for each person).
3. The final set of trajectories overlaid on the first image.

Note that there is no ground truth file supplied for the live demonstration, so your program will not be able to calculate statistics.

The assignment is estimated to take 10 hours coding/test and 5 hours report writing per person, resulting in a 5 page report plus the code appendix. You must do this assignment in teams of 2. You must find your partner and email Bob Fisher (rbf@inf.ed.ac.uk) the name of your partner. A single, joint, report is to be submitted. Split the work so that each partner does most work independently (i.e. share the work rather than duplicate it).

The assignment will be marked as follows:

Issue	Percentage
1. Clear description of sensible algorithms used	30%
2. Performance on first video data set	20%
3. Clear Matlab code	20%
4. Discussion of result quality and causes of any failures	10%
5. Live demonstration performance on new video data set	20%

Publication of Solutions

We will not publish a solution set of code. You may make public your solution **but only 2 weeks after the submission date**. Making the solutions public before then will create suspicions about why you made them public.

Good Scholarly Practice: Please remember the University requirement as regards all assessed work for credit. Details about this can be found at:

http://www.ed.ac.uk/schools-departments/academic-services/students/undergraduate/discipline/academic-misconduct

and at:

http://web.inf.ed.ac.uk/infweb/admin/policies/academic-misconduct

Furthermore, you are required to take reasonable measures to protect your assessed work from unauthorised access. For example, if you put any such work on a public repository then you must set access permissions appropriately (generally permitting access only to yourself, or your group in the case of group practicals).

Plagiarism Avoidance Advice

You are expected to write the document in your own words. Short quotations (with proper, explicit attribution) are allowed, but the bulk of the submission should be your own work. Use proper citation style for all citations, whether traditional paper resources or web-based materials.

If you use small amounts of code from another student or the web, you must acknowledge the original source and make clear what portions of the code were yours and what were obtained elsewhere. You can ignore this condition for the AV lecture examples, which can be used freely.

The school has a robust policy on plagiarism that can be viewed here: http://web.inf.ed.ac.uk/infweb/admin/policies/guidelines-plagiarism.

The school uses various techniques to detect plagiarism, included automated tools and comparison against on-line repositories. *Remember: a weak assignment is not a ruined career (and may not reduce your final average more than 1%), but getting caught at plagiarism could ruin it.*

Late coursework policy

See: http://web.inf.ed.ac.uk/infweb/student-services/ito/admin/coursework-projects/late-coursework-extension-requests