Advanced Vision Practical 1

Bob Fisher
School of Informatics

January 2017

Abstract

This describes the first assignment for assessment on Advanced Vision. The main goal is to count the number of frames where I am sitting at my desk. The assignment is due: **4pm Thursday 16 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 on Friday morning, February 17.

Task Background

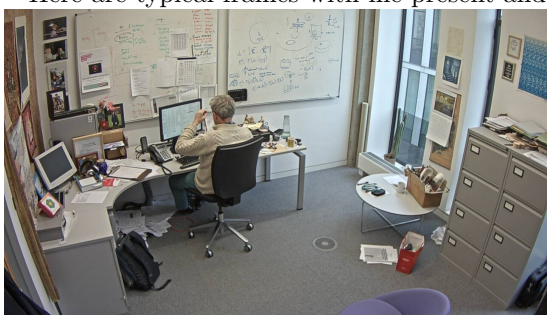
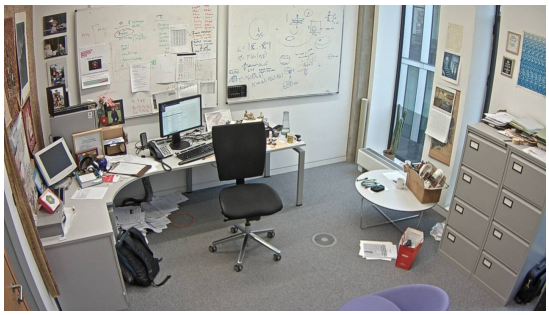
The data for this practical can be found at 2 URLs:

`http://homepages.inf.ed.ac.uk/rbf/OFFICEDATA/TAR/day_2.tar`
`http://homepages.inf.ed.ac.uk/rbf/OFFICEDATA/AVI/day_2.avi`

The tar file has 6719 consecutive images; the other is a 5 minute video. The avi and tar files are about 0.7 GB each. The image file names are of the form:
`inspacecam163_2016_02_19_12_51_42.jpg`, for year, month, day, hour, minute, second.

The images captured show me working in my office. The camera is essentially stationary viewing a stationary background. The camera data is recorded at about 1 frame/second, but occasionally there are larger gaps between frames due to network loads, etc.

Here are typical frames with me present and absent:

The overall task for this assignment is to detect whether I am in the office, and, if so, where I am. To do this you need to write a set of programs that can detect me against a largely stationary background. Complications include my changing position, some items are moved, changing lighting conditions in the office, movement on the computer monitor or outside the window, and the colour of my clothing (which is similar to other objects in the room).

Person Detector

The image data is a set of RGB colour images and the camera is fixed. There is a largely constant (except for the monitor and window) background image and the lighting is mostly constant, at

least over short periods of time. There are some shadows from objects in the room and these will change with the lighting.

You should be able to easily detect the people using a background subtraction or correlation based method. You might find using normalised RGB helps to remove the effect of the shadows, and lighting 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. Finding edges (e.g. using Canny) might also help as their background position is not affected by the lighting.

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

Record the centre of mass of the detected person (if there is one).

You can use general functions from the course matlab, the web or OpenCV, etc, but cannot use any pre-packaged person detector or change analysis programs.

Statistics

Once the detection results are complete, your program should compute some statistics:

1. The number of frames when I am working at the desk
2. The number of frames with no one in the office
3. The number of frames with me at the filing cabinet
4. The number of times I entered and exited the office, and what the time was when I did this (*i.e.* the frame number)

Evaluation

No ground truth is supplied for evaluation of this assignment. Therefore, you need to make some manual tests yourself.

Select 100 frames (either randomly or uniformly) spaced through the video. Estimate by hand the approximate centre of mass of the person.

Compute, over the 100 frames, the false detection percentage, the missed detection percentage, and the average distance (in pixels) between the manual and automatic detection of the person.

Report these percentages in your report, with example images of success and failure, as well as the frame counts and other numbers described in the previous section.

Once you have detected the person in each image, create a new image where the detections are represented by red dots (at the centre of mass of the detected person) and detections in consecutive frames are connected by a red line. Overlay this red line on top of the original image.

Note - it will take several hours of compute time to process all frames. Work on a subset with a variety of conditions to validate your algorithms.

If you want to explore background subtraction, the frame without me shown above is `inspacecam163_2016_02_19_13_05_01.jpg`.

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 successful detections (e.g. show the candidate change regions, and then the centre of mass of the final detection overlaid on the original image.)

- Show some examples of any unsuccessful detections.
- Discussion on performance: successes and failures, causes of failures and potential remedies.
- As an appendix, add the code that your team wrote. Do not include code that was downloaded from the AV or IVR or other 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 16. The online submission line on the School's DICE system 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 17, where you will have to demonstrate your code. We'll email you about the location and schedule.

You will need your matlab program to show:

1. The person detection for a set of 10 frames (randomly selected by the markers from 1..6719) with the centre of mass overlaid over the original image.
2. The image of the red line connecting all of the detections as discussed above. This is precomputed.

The assignment is estimated to take 10 hours coding/test and 5 hours report writing per person, resulting in a 5-10 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 the video data set	20%
3. Clear Matlab code	20%
4. Discussion of result quality and causes of any failures	10%
5. Live demonstration performance	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