

Advanced Vision Practical 1

Bob Fisher
School of Informatics

January 2014

Abstract

This describes the first assignment for assessment on Advanced Vision. The main goal is to reliably detect and track a set of marbles moving in a set of video data. The assignment is due: **4pm Thursday 13 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.

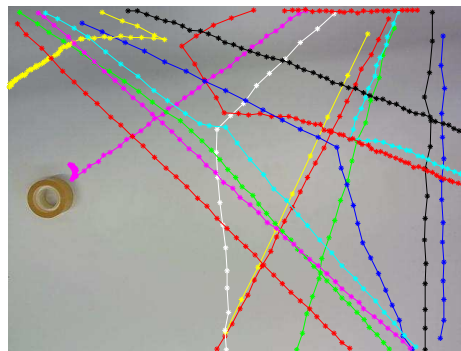
Task Background

At the URL:

<http://homepages.inf.ed.ac.uk/rbf/AV114DATA>

you will find a tar file `av1seq1.tar` of 71 consecutive images. The camera is essentially stationary with a plain background (except for the tape reel to stop the camera's autofocus algorithm from getting lost on the featureless background). The folder also contains a file (`gtSeq1.mat`) listing the 'ground-truth' position of the marble centres by marble and by frame. The `showgt.m` has a function that plots the true ball centres and trajectories, from the ground truth. See `showgt()` to understand the datastructures used for recording the ground truth.

The video shows a set of marbles of different colours rolling on the tabletop. There are up to 18 marbles in each frame, and there are 2 marbles each of 9 different colours or styles. A typical frame (#35) is here (left) and the ground truth trajectories (right). The balls enter from the top:



This assignment is about detecting and tracking the marbles. (Note: if you are colour-blind, please find a partner who is not.)

Your task for this assignment is to write a set of programs that can:

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

Each of these is described in more detail below.

Marble 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. Frame 1 can be used as the background. You should be able to easily detect the marbles using a background subtraction method. If necessary, you can mask out the tape roll, which may move due to collisions and camera jitter.

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

Label the marbles uniquely when they first appear and keep the same label throughout.

Represent each marble by its centre of mass.

Tracking

Determine the correspondence between the marbles in each image. Some marbles will not move much between consecutive frames but others will. There are up to 18 marbles in each image. There are several collisions.

The information that can most easily be exploited is:

1. The RGB colour distribution of each marble is nearly constant. You might represent each marble by a 2D histogram of the r/g components from normalised RGB values. Comparing histograms could use the Bhattacharyya distance,
2. Each marble will move in a roughly constant direction and speed.
3. There is a limited range over which most marbles will move.
4. All marbles appear and disappear at the edges of the image, not in the middle, so marbles in the middle will have to match something from the previous and next frames.
5. There are few collisions.
6. There are 9 marbles entering the view from each side.
7. There are 2 marbles of each colour, one entering from each side.

The Condensation tracker discussed in the lecture video is potentially a good way to solve the tracking problem, because it can keep different hypotheses. In this case, the state vector should be more complex, including information for each of the 18 marbles, such as its position and velocity. In this way, if there is some ambiguity about what detection to pair from the previous frame, then different state vectors can try different pairings. You'll also need the Condensation tracker to cope with collisions.

Estimated probabilities (or some equivalent numerical score) can be based on the Bhattacharyya distances and consistencies of estimated velocities.

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

Evaluation

The ground truth file shows the marble centres as found by hand, for each marble colour. Compare your estimated marble centre (use the centre of mass of the detected pixels) with the ground truth centre for each marble and each frame (approximately 18*71 comparisons in total). You can manually add a table that relates each of your detected tracks to the corresponding track in the ground truth (ie. it is not necessary to do track matching - fun but requires too much extra work for this assignment).

Report the number of marbles that were detected within 10 pixels of the ground truth centre.

Report the mean distance between the ground truth and estimated centres, for all marbles within the 10 pixel distance threshold.

For the full set of videos, report the percentage of the instances where your track correctly pairs the marbles between consecutive frames. I.e. suppose marble 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 marbles, 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. Also report the number of correct and erroneous pairings.

As part of your development, plot the centres of your marble detections on top of each frame. Include a few of these images in your report. The `showgt()` function can be adapted for this task. Pause between each frame.

Draw an image of all trajectories on top of an image frame, where each trajectory has a different colour (out of the set of 8 possible), as done by `showgt()`. The `showgt()` function can also be adapted for this task. Include this image in your report.

Your Report

Write a 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 described above.
- Show example images of each processing stage, including a few examples of detected marbles.
- Show the trajectories of the marbles.
- Show examples of successful and unsuccessful detections, and trajectory connection.
- Report the detection and tracking statistics.
- 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 13. 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 2-5 pm on Friday February 14, 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 foreground marbles for each frame overlaid over the original images using the cross and circle method described above. Use a 1 second pause between images.
2. The tracked trajectories of each marble so far overlaid on the first image (use a different coloured track for each marble).
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 some 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 supplied image data	20%
3. Clear Matlab code	20%
4. Discussion of result quality and causes of any failures	10%
5. Live demonstration performance on supplied video data	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.

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://www.inf.ed.ac.uk/teaching/plagiarism.html>

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.*