Introduction to Vision and Robotics
Assessed Practical 1: Ball Ninja

January 24, 2014

1 Introduction

In this pratical, a robot has to detect, track and recognise objects, Balls (and other random objects) are thrown in the air. The goal of this assignment is to develop an automatic system that will detect the highest point each ball reaches in the given video while ignoring all other objects: If the ball reaches the highest point, the video should pause for 3 seconds and a cross needs to be drawn indicating the position of the ball. There will however also be other objects (random objects), which have to be ignored by the students program when reaching the highest point. (Every correctly detected spherical ball will be given 4 points, successful detection of the highest point will give 1 additional point, every other object incorrectly classified as ball will give -2 points).

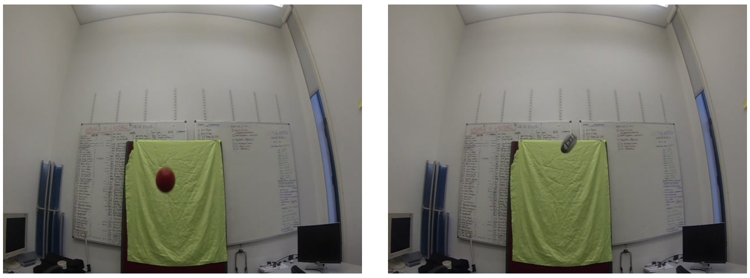
The first 25 frames of the videos will have no objects in them, allowing student to learn a background model. The student will get 4 videos to experiment with, however 2 novel videos will be given to the students on the day of the evaluation, to see how their software performs in solving the problem. The first 25 frames of these videos will have no objects in them (but the background might be different), afterwards ball and other objects are thrown in the air, however the balls can have different colors/sizes and other objects can be thrown in the air as shown in previous videos given to the students.

Two different examples of "images typical of the sequence" are shown in Figure 1:

2 Organisation

You will work in pairs and must submit a joint report *but* this reporst must be accompanied by a short explanation of how the work was shared and how you think the (joint) mark should be distributed. 50:50 is OK if you shared the assignment fairly. **You can choose the partner yourself, but you will also have to choose a different partner for the second practical assignment.**

You should be able to normally access the IVR lab (AT 3.01) anytime with your swipe card, where you can use Matlab at any time on the DICE machines. There are license number limits on availability so *never* stay connected to the programs when you are not using them.



(a) (b)

Figure 1: An example of a ball and other random objects (rugby ball does not count as ball) that need to be detected, tracked and recognised

3 Theory

In lecture, there was an example of background subtraction, but how can one get a background image? If the camera and illumination is largely stationary, as it is here, then you can synthesise a background image by median filtering the values observed over time at each pixel. That is, if $\{v_1, v_1, ..., v_n\}$ are the values observed of the red colour channel at pixel (r, c) at different times, then the value $\mathbf{median}(\{v_1, v_1, ..., v_n\})$ is an estimate of the background value of that colour channel at that pixel.

How can you decide if the object seen in one frame is the same as that seen in the next frame? There are two useful sources of information: 1) Position - assuming that the frame rate is high and the object velocity is low, then the target will not move much between consecutive frames. 2) Colour - the RGB value of the pixels of a target will not change much between consecutive images. How can you cope with the difference in illumination in different places in the image? Use normalised RGB instead of raw RGB values.

When you threshold the difference images, there might be many small regions detected due to image noise. How can they be removed? Look at the `bwmorph(fore,'erode',1)` command.

How might you recognise a spherical ball from other objects? Look at properties of the detected objects and corners/edges. Investigate moments, centre-of-mass, compactness, no sharp corners, etc.

How might you find the object regions? Look at the `bwlabel(foremm,4)` and `regionprops(labeled,['basic'])` commands. What to do if a object is not detected in a given frame? Use an algorithm that can robustly resume a track after N frames (e.g. $N = 3$).

4 The Task

The overall goal is to develop an algorithm that takes the sequence of images and detects and tracks each of the objects, recognising if the object is a ball or not at the highest point that the object reaches. We supply you with four sets of videos to test your methods on.

What you have to do is:

- Develop an algorithm that detects the object in each frame. In your report, present the criteria the defines a detection. If present, show examples of correct detections, missed detections and invalid detections (*i.e.* detections reported at locations where no object is present). Represent the performance of the detection methods in a clear way and discuss observed errors (missed detections and invalid detections).

- Develop an algorithm that correctly links together detections of the objects in consecutive frames, show the path formed by linking together the centres of the detected objects drawn on top of the estimated background image. Evaluate the performance of the tracking of objects (*i.e.* broken track and incorrect tracking). Explain the cause(s) of the invalid/broken tracks if present.

- Develop an algorithm that correctly finds the highest point of the object in the sequence, describe how the method works and how many times it is able to estimate the highest point correctly. (algorithms that do not take future frames into account get higher scores).

- Develop an algorithm that correctly recognises balls and is able to ignore other objects. Describe the features/clues you use to determine if the object is a ball or not. Report on each frame the recognition performance of correctly separating balls from other object. Explain the cause of possible incorrect recognitions.

You should produce algorithms that are robust to common variantions in images, like illumination, camera position/distortions (but you can assume that both remain mostly fixed during the capture of a sequence).

5 Files You Need

You will need these files (which can be downloaded from the IVR webpage):
Dataset 1: `GOPR0002.zip,GOPR0004.zip` to develop your algorithms.
Dataset 2: `GOPR0005.zip,GOPR0008.zip` to test and report results of your algorithm.
During a live demonstration session, you will have to demonstrate your program working on two new sequences that are similar to the sequences supplied here.

6 Writing the report

The report should be a concise description of what you did, why, and what happened. The entire report should be no more than 4000 words (excluding appendices). It should contain the following sections:

1. Introduction: an overview of the main ideas used in your approach.

2. Methods: Explain the vision techniques that you used. Then give a functional outline of how these ideas were implemented and the structure of your code. Explain how each part of it is meant to work. Where suitable, justify your decisions, e.g. why you used one method rather than another, what you tried that didn't work as expected, etc.

3. Results: You should provide some actual data using the set sequences on how well your algorithm performs, as described previously. Show an example of your results for each stage of the detection. Well documented failure will get more marks than unsupported claims of success (well-documented success would be even better!).

4. Discussion: Assess the success of your program with regard to the reported results, and explain any limitations, problems or improvements you would make.

5. Code: the new Matlab code that you developed for this assignment. Do not include code that you downloaded from the course web pages. Any other code that you downloaded should be recorded in the report, but does not need to be included in the appendix. (code is not include in word count)

Your final mark will be based on how well you explain your approach to the task and evaluate the capability of your Matlab program as well as the performance.

7 Live Demonstration

On 7th of March 2014, you will have to demonstrate your program working on two new sequences that are similar to the sequences supplied here. You will be allocated a demonstration time.

8 Submission

Your submission should be a single PDF file and should be submitted electronically by 6th of March 2014 (before 4pm). The command to use for on-line submission is:
`submit ivr cw1 FILENAME`
where FILENAME is the name of your PDF file.
This assignment is estimated to take 10-15 hours work. You must do this assignment in pairs and assign credit in your final report. The assignment will be marked as follows:

Issue	Percentage
Program Design, including comments	25%
Report Clarity	25%
Experimental Results (in Report)	25%
Live demonstration Results	25%

The live demonstration results will be marked based on a combination of target detection (5%), successful tracking (5%), highest point detection (5%), ball recognition (5%) and processing speed (5%).

9 Plagiarism

This assignment is expected to be in your own words and code. 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. Before submitting please acknowledge any additional sources of code that you use and ensure that your submission follows the university policy on plagiarism:
www.ed.ac.uk/schools-departments/academic-services/students/postgraduate-taught/discipline/academic-misconduct