Signal, Image and Video Processing Project

Gabriele Padovani, Giorgio Scarton, Andrea Cristiano

February 2, 2022

1 Initial idea

The project consists in finding and tracking the position of a basketball shot from the free throw line. We chose videos of people shooting a basketball, trying to select ones with a still camera and a clean background. As it turns out, not many videos online have these properties.
It is possible to change the video analyzed with the parameter video_number of the function *execute()*.

2 Process

Our first idea was to use a Yolo classifier to identify the position of the ball and track it accurately. However, we soon realized we could get away with a much simpler approach, using a Blob Detector, which selects groups of pixels with similar properties. Experimenting with the parameters, we were mostly successful in identifying the basketball.
In the instances where the Simple Blob Detector fails, we still made possible selection of the ball by toggling the property select_area of the function *execute()*.
Once the first instance of the ball is found, we made use of a Motion Tracker to trace the object frame by frame. Since many different object tracker were available, we allowed switching between them using the parameter tracker_type of the function execute(). In any case as can be clearly seen by checking results, which are sorted in results folder, the best tracker available is CSRT. We fine tuned parameters for this tracker only.

All the points found during the shot are used to calculate the trajectory of the ball, using linear interpolation. Since most of the times the ball hits the rim, it's trajectory will change. To keep of it as smoothly as possible, we applied a couple of correction involving both x and y axis. The curve is first monotonized on the x axis and then on the y axis between the maximum point and the extreme ones. This way we are able to handle both the cases in which the ball goes from left to right and in the opposite direction. For example, if at time t, the ball has x value 100 and is going towards the right, we remove any points at time later than t that have value smaller than 100. This adjustment is only necessary during the estimation of the trajectory. An example of pre and post application of the correction is below:

To determine whether the shot ended in a score or a miss, we were planning to find the basket position. However, we found extremely difficult pin-pointing this object, as in all videos is always still, and the colorful environment made it tough to use color recognition.
To still be able to give an approximation of the probability of success, without resorting to a machine learning approach, we calculated the variance between the points from the ideal trajectory (the one

Figure 1: Algorithm for removal of unordered points

we estimated) and the points from the actual movement of the ball.
First of all, we calculated the estimated ball direction as a vector, by subtracting the last point (extrapolated maximum) by the last position tracked. This vector also needs to be normalized, by doing this, we are able to multiply any distance by this direction vector, and obtain the new estimated ball position.

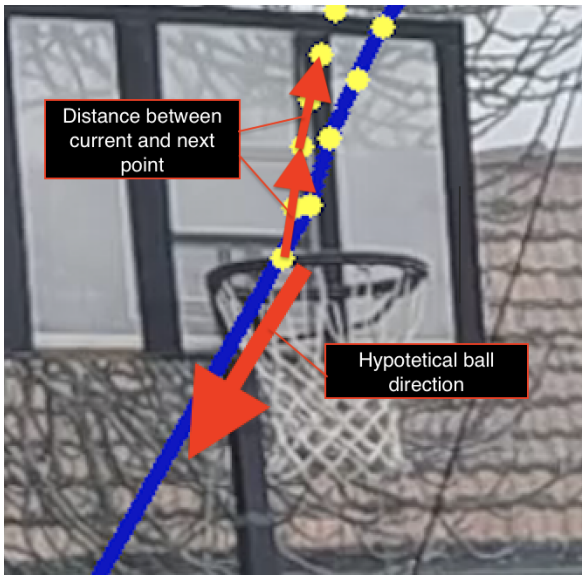

Figure 2: Algorithm for distance calculation

Then, for all the points tracked after the ball hit the rim, we calculated the distance between the position given by the tracker, and the one estimated using the previously calculated direction. A missed shot will have a larger variance between tracked and estimated position, while an airball will have no difference, as the ball never leaves the trajectory. This way we can differentiate, using two threshold, the outcome of the shot, without having to resort to machine learning, or having to identify the basket.

A short recap of the results achieved in our project can be found in the next lines while some statistics and the final result image for each video and for each tracker can be found in results directory divided by the source video name (e.g. in directory /results/ft1/ can be found outcomes of each tracker for video track ft1).

3 Video list and some comments

1. Ball tracked, outcome classified correctly
2. Ball is not found right away, the blob detector has problems identifying the ball, once it finds it, the motion tracker has problems maintaining sight, probably because of the bottom view of the rim.
3. Ball tracked, outcome classified correctly
4. Ball tracked, outcome classified correctly
5. Ball not tracked right away, the blob detector seems to have problems, probably because of the different coloring of the ball passing across a background full of objects. As soon as the ball goes on top the white roof section, the blob detector is able to identify it.
6. Ball tracked, outcome classified incorrectly, probably becouse the video is too short
7. Ball tracked, outcome classified correctly
8. Ball is tracked as soon as it passes on top of the white background. The blob detector is probably not able to find it in front of the tree. Outcome classified correctly
9. Ball tracked, outcome classified correctly