TECHFEST 2022-23

Find the Lag

Design Abstract

Team ID: HA-226107
Team Composition:

Sr. No.	Name	Email Id	Contact
1	Aditya Rajeev Harakare	adityaharakare1@gmail.com	+91-9881251626
2	Ruchir Chheda	ruchir.iitb@gmail.com	+91-8879797905

Introduction

Finding the lag between the control input and the action registered by the computer / server is a crucial aspect of any game, especially single player, real time games which are played on servers or competitive gaming. So the problem statement asks us to develop a system by which we can measure this gap. For most of the games this gap might involve registering control by our computer, sending it across to the server (if required) and receiving response and processing the visual output w.r.t the control / response. There might be a lag in the system because of the frame rate of display and other similar reasons but that should be insignificant given the setup of the experiment.

Approach

We can break down the problem into 2 possible subparts. We have to look at the video in terms of frames, we have to find the start and end frame of the measurement, count the number of frames and time can be calculated using FPS

1. It is to detect the blue light blip, which would be an easy task. We can isolate positions of blue blips ie (positions of left, right …) all the actions. Easiest approach would be to just check the appearance of blue blips() in the black / grey apparatus and mark the frame there. (Start frame)
2. It is to detect motion between images, we can do it using simple technique, averaging out between frames and looking for blurring. Let's say we have frames f1, f2, f3, f4, f5 … We get merged frames m1 = (f1 + f2 + f3) / 3, similarly m2 and m3. We compare the SSIM scores between them. Lets say SSIM between m1 and m2 is 0.98 which is pretty high, there must be no motion between them between m2 and m3 is 0.92, which is still pretty high but considering we are looking for motion ranging a few pixels the estimate

seems fair (we suspect that the threshold for considering a movement will be very high). This should give me the end frame of the measurement.

3. These measurements can solve our above problem in a crude way.

We can work on two aspects to bring an improvement -

1. It is in the direction of robustness -
    a. So while considering blue blips there are 3 possible positions that light up one by one, we can estimate the rough gap between the timings, and get 3 measurements and reinforce the start frame time.
    b. While considering motion blur, we can divide the game images into small patches (16 x 16 or 32 x 32), which will help in measuring pixel level measurements. And thresholding across 10-15 images will bring stability in the detection as against single outliers. This might prove to be crucial as we do not really know which motion is to be generated
2. We can look at different methods for motion detection in terms of blurring which one was mentioned above-
    a. We can look into motion detection based on movement of prominent objects, let's say we find important squares, circles, landmarks in the image look for the movement in the centers of these artifacts this can be a good estimate (this can be given by opencv)
    b. We can also look into measuring movement w.r.t signals that will remain stationary w.r.t the screen (timers, signs etc.).
3. We can try out ML in detection of movement but that might give very clear results as we are looking at pixel level movements and quick movements. Whereas we would rather try using ML in a situation where the essence of the action is of importance and not the exact position or timing as many a times the bounding boxes can be rather lose, they tend to point out that there is an object I have found, not really focussed on when and where exactly to the pixel level and milli-second precision. Though there are approaches like yolo v5 motion tracking that work towards a similar task and give a pretty good precision. We can possibly look at extensions in this direction also. This can work in cases of quick movements w.r.t surroundings, it will be detected smoothly and quickly (because we think that the fps constraint is tighter than the pixel movement constraint. Just to say it out in simple words the objects should move 5s if not 10s of pixels between frames. So we might not really require that high level of pixel precision).

So we will use OpenCV based approaches, make it robust, check for results and look for extensions in ML.