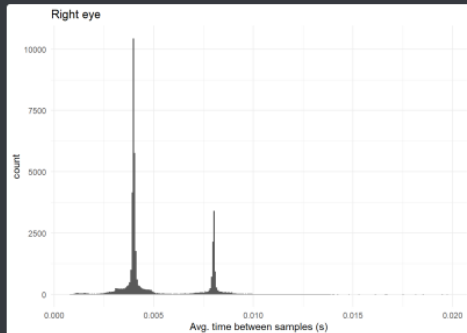




eioe Yesterday at 9:38 AM

Dear pupil community,
just for a quick checkback: I plotted the time between my pupil samples (per eye) for one of my experimental blocks (approx 8 minutes) and this is what I get:



The mean is very close to the 5ms that I was expecting (these data were recorded with the 200Hz VR add-on).

I just wanted to ask whether that is expected behavior or whether I should be suspicious of something in my hardware (or software) setup.
Thanks for feedback already!



papr Today at 11:49 AM

@eioe The pupil timestamps are taken from their respective eye video frames. On Windows, the eye video frame timestamps are calculated based on the time they are received in the application and a fixed offset that compensates the transmission delay from the camera to the application.

Unfortunately, this real transmission delay is not fixed and can only be approximated. What you are seeing is the variance in this transmission delay.



eioe Today at 1:04 PM

@papr thanks. I'm asking as I want to apply a saccade detection algorithm that expects a fixed sampling rate. If the variance in the recorded timestamps is actually driven by transmission times (not by recording time), would it be a legit assumption that the samples are actually equidistantly spread out over time originally?



papr Today at 1:07 PM

@eioe Yes. On operating systems where can measure the time of exposure (macOS/Linux), we see a much smaller variance in the timestamp differences.



eioe Today at 1:11 PM

@papr interesting. So for velocity calculations would you recommend using the pupil timestamps (in my case unfortunately recorded on Windows) or assuming a fixed delay of 5ms between two samples of the same eye?



papr Today at 1:13 PM

I would assume a fixed 5ms duration between samples, but rely on the pupil timestamps to detect dropped frames, e.g. if the time difference between pupil timestamps >10ms



eioe Today at 2:04 PM

@papr thanks! so the 2nd/right peak at ~8ms in the histogram above (~25% of the data) does not consist of actually delayed frames due to dropped frames but of problems with the time estimation? Actually dropped frames are then the single points beyond 10ms (in my case <1%)?
That's very good to know.