**Latency Optimization Guide (rough draft)**

**Notes from our manuscript (currently under review)**

To estimate the effect of different setup components on the overall latency of our optimized system, we examined different hardware and software configurations.

To assess the latency due to using the digitizing tablet as a source of position information, we replaced it with a low-latency computer mouse, which reduced latency by about 16ms, i.e. 60-70% of the 25ms overall latency. Despite the tablet being responsible for most of the overall latency, however, we did not substitute it with a mouse, as the tablet provided higher position precision and, importantly, provided absolute position measurements that would be reliable even if the pen were lifted and placed on a different location on the tablet.

To assess the effect of the experiment environment, including execution time, we compared latency outside vs. within the experiment environment, using mouse input in both cases, finding that latency was merely 0.8ms lower outside Matlab, i.e. about 3% of the 25ms overall latency.

Finally, our screen’s input lag was 4.4ms or about 18% of the overall latency.

A notable finding during the process of optimizing our setup was the importance of how the software handled screen updates. We coded our experiment to use asynchronous screen updates, as we found that synchronous screen updates added an additional ~17ms (60-70% more) to the latency.

**Best Practices (stump - under construction)**

1. Measure and report latency! A tool for assessing latency from videos is provided in this folder.
2. To minimize latency, aim for
   1. Monitors with minimal input lag and high refresh rates. At 60 Hz, the refresh rate would add, on average, 8.3ms of latency (16.7ms/2); at 120 Hz, the refresh rate would add half that, 4.2ms. It may be worth investing in a good monitor like the ones used within the gaming community.
   2. If a projector needs to be used, keep in mind that projectors generally add large latencies (~100ms for older models), though more recent models can be better. We found that <https://www.projectorcentral.com/> very useful in comparing details, including projector lag. Note this lag is only one part of the total latency experienced: it would add to latencies from sensor input, computer processing, and graphics output especially when common double-buffering schemes are used for graphics.
   3. Use a tablet / input device with a low input lag. As we mention above, replacing the tablet with a computer mouse as the source of position input reduced latency from 25ms to about 9ms. However, this comes with other limitations: tablet offers higher precision and provides absolute position measurements. The latter is important if somebody were to lift the mouse and place it on a different location.
   4. It’s best to have the experiment code update the screen asynchronously. In our case, using asynchronous updates decreased latency from about 42ms to 25ms. If using MATLAB Psychtoolbox, synchronous/asynchronous updating is controlled through an argument in the Screen function:

Screen(‘[Flip](http://psychtoolbox.org/docs/Screen-Flip)’, windowPtr [, when] [, dontclear] [, dontsync] [, multiflip]);

Where dontsync is 0 by default, but can be set to 2 which tells the function to show the stimulus right away without any wait or synching. For example, in our code, Screen('Flip',win,0,0,2) updates the screen window “win” asynchronously. Detailed description of Screen-Flip can be found here:

<http://psychtoolbox.org/docs/Screen-Flip>.