Running Head: B210 RT MEASUREMENT UNDER FLASH

Adobe Flash as a Medium for Online Experimentation: A Test of RT Measurement Capabilities

Stian Reimers and Neil Stewart
University of Warwick, United Kingdom.

Production number: B210

New address for correspondence:

Department of Psychology

University College London

26 Bedford Way

London WC1H 0AP

England

Tel: +44 207 679 5317

Fax: +44 207 436 4276

s.reimers@ucl.ac.uk

Reimers, S., & Stewart, N. (2007). Adobe flash as a medium for online experimentation: A test of RT measurement capabilities. *Behavior Research Methods, 39,* 365-370.

Abstract

Adobe Flash can be used to run complex psychological experiments over the web. We examine the reliability of using Flash to measure reaction times (RTs), using a simple binary choice task, implemented both in Flash and using a Linux-based system known to record RTs with millisecond accuracy. Twenty-four participants were tested in the laboratory using both implementations; they also completed the Flash version on their own computer. RTs from Flash on users' own computers were ~20ms slower than those using Flash in the lab, which were in turn ~10ms slower than baseline. RT standard deviations were similar in all conditions, suggesting that although Flash may overestimate RTs slightly, it does not appear to add significant noise to the data recorded.

Adobe Flash as a Medium for Online Experimentation: A Test of RT Measurement Capabilities

Adobe Flash, formerly known as Macromedia Flash, and colloquially called Flash, is an authoring application that allows programmers to create content for display over the web. Using vector graphics and a combination of simple frame-based animation and a programming language called Actionscript, Flash is most often used to create high-impact moving images to make websites attention-grabbing. Google indexes several tens of millions of Flash movies (the term "movie" is traditionally used to describe Flash programs, even though, as we shall see, it only captures one of Flash's capabilities), and Flash is particularly used for creating animations, games, splash screens, and, increasingly, advertisements. As such, Flash has become one of the key ways of supplying interactive content to web users.

In the past, Flash was predominantly used for animations. However, the development of Actionscript has meant that in principle Flash can be used to implement psychological experiments of almost any design, subject obviously to the constraints of web-based experimentation. Psychologists have begun to use Flash for running surveys and experiments over the web, often replicating results obtained in the laboratory (e.g., Reimers, in press; Reimers & Maylor, 2005, 2006).

Although there are many different ways of running experiments over the web – the dominant method being the use of surveys involving HTML forms – a key advantage of Flash is that it allows the measurement of accurate timings, which could make it suitable for reaction time- (RT-) based research.

There has been limited investigation of Flash's suitability for psychological experimentation. Schmidt (2001) compared presentation duration accuracy using several web animation methods. He found that Flash performed adequately on faster systems but deteriorated very significantly on slower machines. As the slower machines ran at 60-75MHz, and the faster machines only ran at 400-500MHz, meaning both are obsolete now, it is unclear how modern versions of Flash Player on more powerful computers would perform on a similar test.

We know of no studies in which Flash's accuracy in measuring response times has been quantified. As Flash runs within a browser, and has no control over interrupts from other programs, there are reasons to suspect that Flash would not record RTs as accurately as some other experiment implementations. Our aim is to investigate the degree of accuracy with which RTs are measured, relative to a baseline system which has demonstrated millisecond accuracy. We also aim to compare the performance of Flash in controlled laboratory computers and on computers of participants' choosing outside the laboratory, over which we have no control.

Introduction to Flash Player

There are two components involved in setting up and running Flash-based experiments: an authoring program, which the psychologist/programmer uses to develop experiments and export in .swf (Shockwave Flash) format, and the application player, which participants use on their own computer to run .swf files. Confusingly, both of these components are often referred to simply as 'Flash'. Here, we adopt the convention of calling the authoring program

'Flash' and the application player 'Flash Player'. Additionally, although we only discuss Adobe's proprietary authoring tool and application player, other software for creating and playing Flash files exists, including the authoring program "SwishMax", and the open source Flash player "Gnash".

Flash Player is generally run as a web browser plug-in, and is installed on the majority of internet enabled computers: In August 2006, Adobe claimed that Flash Player was installed in 97.3% of internet enabled desktop computers (Adobe, n.d.). Flash Player has been developed for several different platforms including Windows, Linux, MacOS, as well as for operating systems used in handheld devices, including Palm OS and Symbian.

Like Java, Flash Player runs as a sandboxed virtual machine. Data can be stored on the client machine and accessed later, in a similar manner to the use of cookies, but otherwise, access to the client machine's hard drive is not permitted. Data can be transmitted from Flash Player across the web to a remote server, but only to the domain from which the program was downloaded. In other words, a Flash experiment accessed at http://www.sitea.ac.uk/Experiment1.swf could pass data to sitea.ac.uk/cgi-bin/datahandler.pl, but not siteb.ac.uk/cgi-bin/datahandler.pl. Data can be sent using the standard GET or POST methods, as with HTML forms.

The player's initialization is fast and automatic, meaning in principle that users can be unaware that the plug-in has started running. However, in April 2006, an update to Internet Explorer changed the way that plug-ins initialized, which meant that users had to click on an embedded Flash movie to interact with it (although Javascript-based techniques have recently been

developed to circumvent this problem). Flash Player is also small in size, and does not have odious system requirements to run: recommended minimum system requirements for PCs and Macs are a 500 MHz processor (Pentium II or G3), and 128Mb of memory.

Introduction to the authoring program Flash

The authoring program Adobe Flash uses a combination of frame-based animation and Actionscript code associated with frames and objects. Frames work as in a traditional movie. The speed of the movie is set globally in frames per second, and the contents of each frame are displayed in sequence, at the specified frame rate. Flash contains an interface for drawing items within a frame, using standard graphics tools such as paintbrush, pencil, line, and circle. These items can be defined as objects, which can then be addressed by Actionscript code, and can have their attributes altered across frames.

Animation involves copying and changing objects across frames. For example, one could have a red square object at coordinates (50, 50) in Frame 1, copy it to Frame 100, and move it to coordinates (450, 450). By selecting 'create motion tween', the position of the square in frames 2 to 99 is interpolated. Thus, at a playback speed of 50 fps, the square would move smoothly across the screen in two seconds when Frames 1 to 100 were played. Similar animation techniques can be used on a number of attributes an object may have, such as magnification, position, rotation, transparency, brightness, and skew.

Although animation techniques can allow one to create visually

pleasing movies, it is the Actionscript code in which full experiments can be generated. Actionscript can be associated with frames and objects. Frame-based Actionscript is executed unconditionally when a frame is entered, and can include a `stop();` command, which prevents the movie from advancing to the next frame until a `play();` or similar command is given. Object-based code uses conditional statements that refer to user input. Often, the conditional statements refer to manipulation of the object itself: For example `on(press){x+=1;}` would increment the value of x each time the participant clicked on the object in question. However, the conditional statements can also refer to keyboard input: For example `on(keyPress "<Space>") {x+=1;}` would increment the value of x each time the participant pressed the space bar.

The combination of frame-based animation and Actionscript code within a frame or object can be confusing to a programmer. However, it does allow for the integration of eye-catching, user-friendly animations – which can be used to lead a participant through an example of the task they are about to perform – and the flexibility to implement complex experimental designs.

Suitability of Flash for Psychological Testing

Clearly, all methods for testing across the web have advantages and disadvantages. The similarity in architecture between Flash and Java (client-side, sandboxed, virtual machine) means that the two share many of the same advantages and disadvantages. Briefly, advantages over, say, JavaScript are more precise control over stimulus appearance and duration, and more accurate timing. Advantages over specialist packages such as Authorware

include having a more ubiquitous plug-in, so most participants are not forced to visit a third-party website and download a plug-in before they can participate. Disadvantages include the need to learn a relatively complex programming language, and possible differences in performance across platforms.

To distinguish between Flash and Java, the main advantages of Flash are the ubiquity of the plug-in, the initialization time (Java can take several seconds to start up), and the familiarity web users have with Flash-based content. An additional compelling advantage is the graphical user interface and frame-based structure of Flash, which makes it easy for a relatively inexperienced programmer to create ergonomic content, particularly using pre-prepared components like drop-down menus, text boxes, and buttons. One advantage of Java is that it is a more established, and rigorous, programming language, although Flash is improving in that direction. Also, Java code can be adapted easily for use in other domains, for example, non-networked computers and handheld devices. Again, there have been recent developments in Flash, inasmuch as basic versions of Flash Player are beginning to be found on mobile devices. We have recently implemented simple experiments on cellphones using both Flash and Java, which demonstrates the potential of both languages to run on other devices.

Flash has enough potential advantages as a testing medium for it to be worthy of consideration. Its disadvantage is that, as a new medium for psychological experimentation, it has not been as thoroughly tested. In particular, one key factor that appears largely untested is the accuracy with

which Flash can measure RTs. There are few options for web experimenters wishing to measure RTs with sub-100 ms accuracy. Authorware and Java are perhaps the only current alternatives that allow relatively accurate measurement of reaction times over the web.

In this experiment, we consider two sources of error that can distort RT measurements: random and systematic error. Random error can come from quantizing errors, and variability in timings of stimulus to response. A combination of random and systematic error can come from the delays in sampling the keyboard, interrupts from other programs meaning key or mouse presses are not immediately detected, and delays between calling a procedure to, say, put an image on the screen, and its actual appearance. Systematic error may cause recorded RTs to be substantially different in mean, but not standard deviation, from those recorded perfectly (e.g., they may all be shifted to be 50 ms slower). Conversely, random error should cause recorded RTs to be substantially different in standard deviation, but not mean, from those recorded perfectly. The two types of error have different implications for interpretation of noisy data. Systematic error makes it difficult to compare means from experiments using different measurement apparatus. However, it is rare to find direct numerical comparisons between different experiments – experiments generally compare the effects of an experimental manipulation between or within subjects, using a single measurement procedure. Although random error does not affect the means of any results, the increase in variance may lead to an experiment losing power to detect small differences between group means. That said, the addition of random error of the order of tens of

milliseconds often makes very little difference to tests of differences between groups (Ulrich & Giray, 1989). Additionally, if the amount of error is relatively small, it can be compensated for by increasing the number of trials (so the estimate of participant means is more accurate) or the number of participants (so that the estimate of group means is more accurate), or both. However, there are practical limitations on the extent to which number of trials or number of participants can be increased, so it is worth investigating the amount, and type, of error that Flash adds to any RTs, to evaluate whether Flash is viable for use in online experiments.

The Present Study

The present study was designed to allow us to compare RTs as measured by Flash with those measured by a system known to be reliable, under experimental conditions. Our reliable Baseline condition, which is known to measure RTs with millisecond accuracy, was programmed in C, and used the same set-up described in Stewart (2006a, 2006b). We compared this with two Flash conditions. In the Lab Flash condition, the setting and procedure was identical to that of the Baseline condition. Participants were seated at the same computers, and were given the same instructions. In the Non-lab Flash condition, participants were asked to find a web-connected computer and take the Flash experiment in their own time. The task we used was choice reaction time (CRT). We implemented an experiment measuring CRT, once in Flash and one in C, keeping the appearance as similar as possible across implementations.

Method

Participants

Participants were 24 undergraduate psychology students at the University of Warwick. All participated for course credit.

Design

There were three within-subjects conditions: Baseline, Lab Flash, and Non-lab Flash. Order of conditions was counterbalanced, subject to the constraint that Baseline and Lab conditions occurred consecutively (to avoid having participants make two visits to the lab). Thus there were four orders: Baseline, Lab, Non-lab; Lab, Baseline, Non-lab; Non-lab, Lab, Baseline, and Non-lab, Baseline, Lab. In all conditions, participants classified rectangles appearing on the screen as red or green, by pressing a key on a buttonbox (Baseline) or keyboard (Lab Flash and Non-lab Flash). The experiment comprised 30 such binary classifications. All trials were independent: on each trial there was an equal (.5) chance of the rectangle being red or green. Response key mapping was counterbalanced: half of participants pressed the left key if the rectangle was green, and the other half pressed if it was red. Response key side was the same for a given participant in all three conditions

Implementation

Flash (Lab and Non-lab conditions). A 750 x 500 pixel flash movie was embedded in a grey webpage. The first frame of the movie requested participants' ID number for counterbalancing purposes. The second contained instructions. The third contained the Actionscript code that ran the experiment. An invisible movie clip was embedded in the movie frame, containing an `onClipEvent(keyDown)` event handler. The procedure for measuring

reaction time was as follows. In line 1, a call to make either the red or green rectangle objects visible was made, setting `MovieClip._visible = true`. In line 2, the number of milliseconds elapsed since the movie started was recorded, using the `getTimer()` function. (It made no difference in RT measurements whether the timer was polled before or after the command to make the rectangle visible.) In line 3, the flag `allowresponse = true` was set, meaning the event handler would accept keypress responses. When a key was pressed, the event handler checked that `allowresponse == true`. If so, it recorded the number of milliseconds elapsed since the movie started and subtracted the value at the previous poll, when the target was made visible, giving the RT. It then reset `allowresponse = false`, meaning further keypresses would be ignored, and initiated the display of feedback. At the end of the experiment, a string of reaction times and accuracy was sent to a perl script on a remote server using the POST method; the perl script saved the data to a text file.

C (Baseline condition). A C program using the X Window System to display stimuli (Stewart, 2006b) and a parallel port button box (Stewart, 2006a) was used. The program was scheduled as SCHED_FIFO using the `sched_setscheduler()` system call to prevent interruption by other tasks running on the system. The memory used by the program was locked using a `mlockall()` system call to prevent the memory being swapped to disk. Timings were measured using the `gettimeofday()` system call, which provides microsecond accuracy. Finney (2001) discusses these techniques in detail.

RTs were measured from the end of the vertical retrace upon which the stimulus was drawn to the first detection of a close of a button on the parallel port button box. The error in this measurement was determined by repeatedly sampling the parallel port, and measuring the difference between the last time upon which the button was not pressed and the first time at which it was pressed. In all cases, the error was less than 1 microsecond.

Procedure

The system used for the lab experiments (Baseline and Lab Flash) was a dual processor 1.4 MHz AMD Athlon with 256 MB of RAM with a PCI NVidia GeForce 2 MX with 32 MB of video RAM. We used the Debian Sarge GNU distribution with a 2.4.27-2-k7-smp Linux kernel, XFree86 4.3.0 and the NVidia 1.0-5336 driver4 (and not the nv driver that comes with XFree86). The CRT monitor was a Sony CPD-G220 Color Trinitron running with a refresh rate of 85 Hz. The C code was executed from the command line; the Flash program was run by typing the URL of the testing page into a Firefox web browser. For the Non-lab condition, participants used their own machines, or student machines at the University.

The experiment began with instructions presented in a full-screen, borderless gray window. Each trial began with a “+” prompt in the center of the screen. After a random, uniformly distributed interval of between 1500 ms and 3000 ms[1], a red or green rectangle 200 x 100 pixels was displayed in the center of the screen until the participant responded. The rectangle was immediately replaced with either the word "correct" or "wrong" as feedback. There was then a 2000 ms ITI, during which the screen was blank, before the

next trial began. At the end of the experiment, data were written either to a remote server (Lab Flash and Non-lab Flash) or the local hard disk (Baseline).

Results

One participant was assigned the wrong participant number, and so their data were discarded. To allow appropriate counterbalancing, the participant that immediately followed the excluded participant was also excluded. This left 22 participants' data in the analysis.

There are two main aims of the analysis. The first is to compare group means and standard deviations, to investigate the extent to which the statistics typically reported in results sections are affected by the location and implementation of the experiment. The second is to look more qualitatively at the data, to investigate whether RTs in all conditions follow the same shaped distribution, or whether in Flash they are quantized, or skewed in ways that could affect interpretation of results.

The group means and standard deviations, along with error rates, are given in Table 1. Only RTs of less than 1 second are included in the analysis (100% of Baseline trials, 99.5% of Lab Flash trials and 99.2% of Non-lab Flash trials). One Baseline trial of 0 ms and one Non-lab Flash trial of 13 ms are excluded. All other RTs were in the range 203-956 ms.

To test the significance of any differences in RT and SD, we constructed a general linear model with between-subjects variables of in-lab counterbalancing (Baseline then Flash; Flash then Baseline) and lab Flash/nonlab Flash counterbalancing (Lab Flash then Non-lab Flash; Non-lab Flash then Lab Flash); and within-subject variables of condition (Baseline,

Lab Flash, Non-lab Flash). With participants' RT means as the dependent measure, there was a main effect of condition, $F(2, 36) = 4.07$, $p = .026$, and no other main effects or interactions. A similar effect was found with medians as the dependent measure, $F(2, 36) = 6.18$, $p = .005$. With participants' RT SDs as the dependent measure, there were no significant effects or interactions (all p's > .1), and with interquartile range as the dependent measure, the only significant effect was in the lab Flash/nonlab Flash counterbalancing variable, $F(1, 18) = 6.36$, $p = .02$. Finally, using range as the dependent variable, there was a significant effect only in the interaction between the two counterbalancing variables, $F(1, 18) = 4.93$, $p = .04$.

Figure 1 about here

Estimates of the average and the spread of RTs in the different conditions suggest that RTs recorded with Flash are between 10 and 40 ms longer than those recorded in the Baseline condition. The second aim now is to look qualitatively at the individual RTs, to investigate whether there are any effects of quantizing or any large differences in the distribution of RTs. To do this, we constructed a cumulative frequency distribution (CFD) chart for the three conditions (Figure 1). As we are largely interested in any mechanical differences in the different conditions, we included every datapoint from all 22 participants in the analysis (including the short and long RTs excluded from

the previous analysis, but still excluding the single 0 ms RT from the baseline condition). Any quantizing would be detectable as step-like patterns in the CFD, and any differences in the shape of the RT distributions would be detectable inasmuch as the CFDs would deviate from x-axis-shifted copies of each other.

There is no evidence of quantizing, suggesting that in all three conditions RTs were not constrained to a limited set of values (although this does not rule out quantizing elsewhere within the system). There is, arguably, a small difference in the distribution of RTs between the Lab Flash condition and the other two conditions as shown in apparent gradient differences of the CFD. This is clearly a small effect, and although further research into the effect is merited, a bias of this magnitude does not appear to undermine the viability of Flash in general, particularly as the more realistic Non-lab Flash condition does not appear to be different in shape from the Baseline. However, it suggests that the distribution of RTs may be slightly different in the Lab Flash condition from the two other conditions.

Discussion

The comparison of RTs measured using a millisecond-accurate baseline, Flash run on laboratory computers, and Flash run on web-connected computers outside the lab, suggests that there are small differences in the measured times across the conditions. RT means for Flash administered in the lab were around 10 ms longer than Baseline, and RTs for self-administered Flash outside the lab were 30-40 ms longer than Baseline. There were no significant differences in the RT SDs across the three conditions.

We set out to investigate whether testing using Flash adds random and/or systematic error to RT data. Our conclusion, based on the results above, is that Flash adds a small amount of systematic error to RTs, in that RTs measured in the lab are longer than Baseline. Small errors in RT measurement are largely unavoidable: Given a screen retrace takes 10-20 ms, any RTs based on display of visual stimuli are likely to have a certain amount of systematic error added to them, unless the issue of screen refreshes is explicitly dealt with (as it was in the Baseline condition)[2].

Flash does not appear to add significant random error to RT measurements. Ignoring conspicuously long RTs (> 1 second), standard deviation, interquartile range, and full range were the same in all three conditions. Unlike the Baseline condition, there were a small number of RTs in the Flash condition which were greater than 1 second (although only ~0.5% of responses). Some of those in the Non-lab Flash condition may be due to distractions, but others may be to be due to the interruption of other active applications on the computer. Whatever the cause, these data suggest that searching for and eliminating outliers is particularly important when looking at Flash data.

Use of Flash in the lab and over the web

A final finding of interest is that RTs in the Non-lab Flash condition are, on average, longer than those in the Lab Flash condition, even though both used Flash. There is a confound here between physical and psychological differences between the conditions. In the Lab condition, Flash was run under Linux, and no other user applications were active (apart from the web browser

in which the Flash was displayed). In the Non-lab condition, it is likely that the majority of participants used networked Windows machines. It is also likely that they had other applications open as they completed the experiment (most received an email with a link to the testing site, meaning that their mail tool was probably also active). Similarly, there are clear psychological differences between participating in a laboratory (where an experimenter is present, the time of participation is fixed, the room is quiet and contains no distractions) and at one's own computer. We wanted to make the conditions as similar to those encountered by web-surfers arriving at a site, by avoiding instructions to close other applications, and not telling people to minimise other distractions. Thus, it is increases our confidence in Flash that there was no more random error in the Non-lab Flash condition than Baseline, and that mean RTs were only 30-40 ms longer than baseline. Of course, this was not a perfect simulation of the conditions under which web participants generally complete an experiment. Our participants had met, or were going to meet, the experimenter; they knew they had to complete the task for course credit, and half had completed the task already. The range of computers our participants used was probably smaller than those used by participants in web-based research. Thus, we do not want to generalize too strongly from these results to all potential web-based Flash studies.

Similarly, software and hardware evolve at a rapid rate. Since we drafted this manuscript, a new version of Actionscript (3.0) has been released, which has changed the way in which experiments can be coded. Similarly, new versions of Flash Player have been released (8 and 9), which have been

designed to improve performance. It is therefore likely that RT accuracy will change slightly as the authoring program and application player develop. However, the results presented here suggest that, in principle, Adobe Flash is a viable method for running large-scale RT-based experiments.

References

Finney, S. A. (2001). Real-time data collection in Linux: A case study. *Behavior Research Methods, Instruments, and Computers, 33,* 167-173.

Adobe (n.d.). *Flash Player Statistics.* Retrieved August 21, 2006 from http://www.adobe.com/software/player_census/flashplayer/index.html

Reimers, S. (in press). The BBC Internet study: General Methodology. *Archives of Sexual Behavior.*

Reimers, S., & Maylor, E. A. (2005). Task switching across the life span: Effects of age on general and specific switch costs. *Developmental Psychology, 41,* 661-671.

Reimers, S., & Maylor, E. A. (2006). Gender effects on reaction time variability and trial-to-trial performance: Reply to Deary and Der (2005). *Aging, Neuropsychology, and Cognition, 13,* 479-489.

Schmidt, W. C. (2001). Presentation accuracy of Web animation methods. *Behavior Research Methods, Instruments, and Computers, 33*, 187-200.

Stewart, N. (2006a). A PC parallel port button box provides millisecond response time accuracy under Linux. *Behavior Research Methods, 38,* 170-173.

Stewart, N. (2006b). Millisecond accuracy video display using OpenGL under Linux. *Behavior Research Methods, 38,* 142-145.

Ulrich, R., & Giray, M. (1989). Measuring reaction times: How

accurate must a clock be? Good news for bad clocks! *British Journal of Mathematical and Statistical Psychology, 42*, 1-12.

Footnotes

[1]In the Flash implementation, the movie frame rate was set to the default 12 frames per second. This had the unforeseen effect that the check as to whether the interval duration had been reached was also only made 12 times per second, even though the experiment was contained within a single frame. Thus, in Flash, the intervals participants experienced were quantized, and on average 42ms longer than in the Baseline condition. It is unlikely that this made any difference to performance, given the small (i.e., 2%) difference in mean delays, the sub-100 ms quantizing, and the noise added to the quantized intervals by the monitor refresh rate. More importantly, RT measurement is unaffected by frame rate, so was not quantized.

[2]Although a screen refresh rate of 50 Hz would be expected to add 10 ms systematic error to RTs (assuming it added uniformly distributed noise with a range of 20 ms), the random error it contributed would be negligible. This is because the new SD would approximate to the square root of the sum of the squares of the participant's SD and the noise SD. In a simulation, we found that the effect of a 50 Hz refresh rate was to add just ~0.2 ms to RT SDs.

Author Note

Stian Reimers and Neil Stewart, Department of Psychology, University of Warwick, Coventry, UK.

We are grateful to Petko Kusev for his assistance in running the experiment

This research and Stian Reimers were supported by a grant from HSBC Bank. Correspondence concerning this article should be addressed to Stian Reimers, who is now at Department of Psychology, University College London, WC1H 0AP, UK. E-mail may be sent to s.reimers@ucl.ac.uk.

Table 1

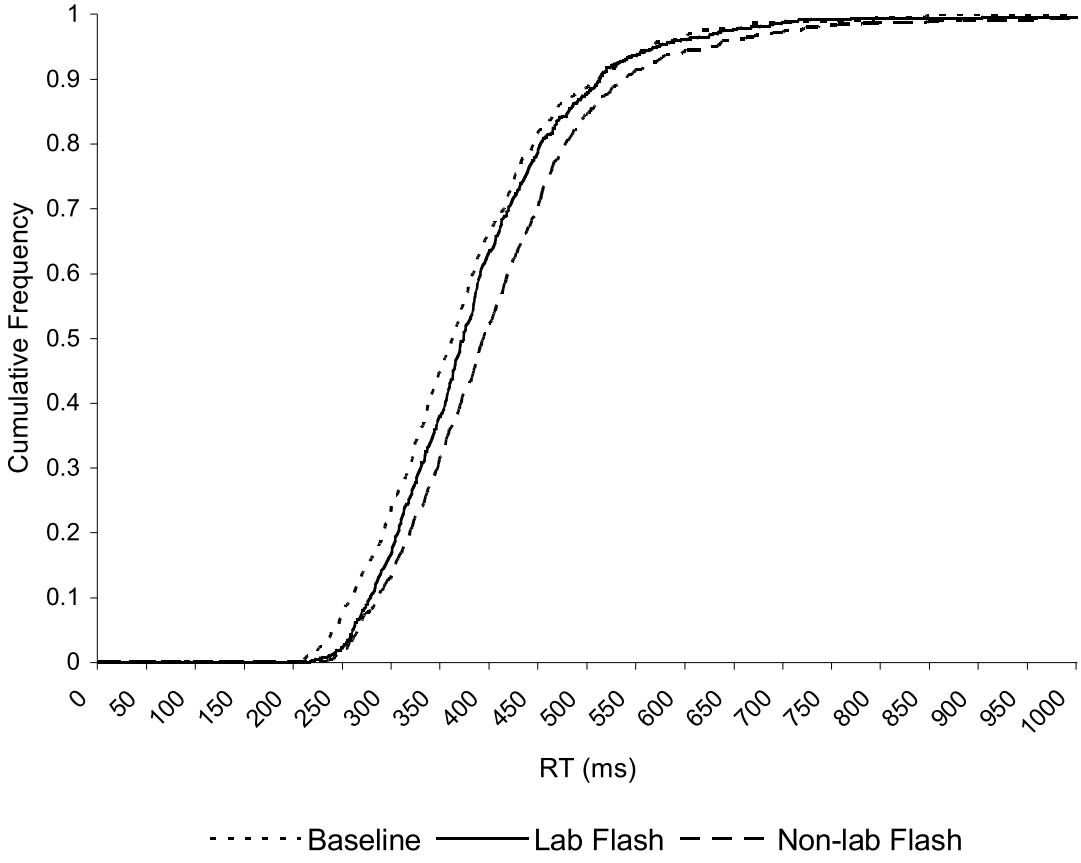
Differences in Average and Variability of RT Across Testing Condition (All in ms), and Error Rate.

Condition	Mean of participant RT means	Mean of participant RT SDs	Mean of participant RT medians	Mean interquartile range	Mean range	Error rate
Baseline	376.5	89.2	363.6	101.0	380.8	4.0%
Lab	386.6	83.0	376.8	96.2	369.0	4.4%
Non-lab	407.9	87.4	402.5	101.8	379.5	5.6%

Figure Caption

Figure 1. Cumulative frequency of trials as a function of RT (ms) for millisecond-accurate baseline, laboratory-administered Flash, and Flash run on participants' own computers.

Figure 1


Cumulative Frequency
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
50
100
150
200
250
300
350
400
450
500
550
600
650
700
750
800
850
900
950
1000
RT (ms)
Baseline
Lab Flash
Non-lab Flash