Webcam-based online eye-tracking for behavioral research

Xiaozhi Yang[1], Ian Krajbich[1,2]

[1]Department of Psychology, The Ohio State University

[2]Department of Economics, The Ohio State University

Abstract:

Experiments are increasingly moving online (especially during the COVID epidemic). This poses a major challenge for researchers who rely on in-lab process-tracing techniques such as eye-tracking. Researchers in computer science have developed a web-based eye-tracking application (WebGazer) (Papoutsaki et al., 2016) but it has yet to see use in behavioural research. This is likely due to the extensive calibration and validation procedure (~50% of the study time) and low/inconsistent temporal resolution (Semmelmann & Weigelt, 2018), as well as the challenge of integrating it into standard experimental software. Here, we incorporate WebGazer with the most widely used JavaScript library among behavioral researchers (jsPsych) and adjust the procedure and code to reduce calibration/validation and dramatically improve the temporal resolution (from 100-1000 ms to 20-30 ms or better). We test our WebGazer/jsPsych combination with a decision-making study on Amazon MTurk. We find no degradation in spatial or temporal resolution over the course of the ~30-minute experiment. We replicate previous in-lab findings on the relationship between gaze dwell time and value-based choice. In summary, we provide an open-source, accessible, software template and tutorial for web-based eye-tracking in behavioral research that is sufficient to replicate in-lab studies with just a modest number of participants (N=35), and that is orders of magnitude faster than in-lab data collection. Moreover, we highlight that web-based eye-tracking is a useful tool for all behavioral researchers, as it can be used to ensure that study participants are humans and not machines.

Introduction

How people allocate attention is a crucial aspect of human behavior. It dictates the degree to which different information is weighted in guiding behavior. Attention is sometimes measured indirectly by inferring it from choice data or response times (RT). But increasingly, attention has been measured more directly using eye-tracking. Eye-tracking makes use of the eye-mind hypothesis: people generally look at the information that they are thinking about (Just & Carpenter 1984) (though this is not necessary).

The use of eye-tracking has become an important tool in decision science, and behavioral science more generally, as it provides a detailed representation of the decision process (Mormann et al., 2020). It has been used to understand the accumulation of evidence in sequential sampling models of choice (Krajbich, 2019), context effects in multi-attribute choice (Noguchi & Stewart 2014), strategic sophistication in games (Polonio et al., 2015), selfish vs. pro-social tendencies in altruistic choice (Teoh et al., 2020), simplification strategies in many-attribute choice (Fellows, 2006), etc.

A challenge to the continued growth of eye-tracking research, and physiological research more generally, is the shift of behavioral research from brick-and-mortar labs to the internet (Goodman & Paolacci, 2017). This shift has been accelerated dramatically due to labs being closed or severely restricted due to the COVID-19 pandemic. While online data collection has many advantages (e.g. speed, affordability), it has, so far, not been used to collect eye-tracking data in behavioral research.

However, there is reason for hope. Eye-tracking is, in principle, fairly simple. A (video) camera is pointed at the subject's face while a computer algorithm analyzes the resulting images to determine where the subject is looking. Thus, to do eye-tracking you need a good camera and

clever/powerful computing. As humans spend more of their time socializing and working online, front-facing cameras on computers, tablets, and phones are continually improving. Moreover, eye-tracking has garnered a lot of interest in the domain of human-computer interaction (HCI). For example, gaze-aware games can improve the gaming experience by providing timely effects at the gazed location (Majaranta et al., 2019). Consequently, researchers in computer science have been working to improve the algorithms to determine gaze location (e.g., WebGazer, Papoutsaki et al., 2016; Smartphone eye-tracking, Valliappan et al. 2020; TurkerGaze, Xu et al., 2015).

Here, we capitalize on these recent efforts and advances to bring eye-tracking to online behavioral research. We start with WebGazer, a JavaScript toolbox that was developed to monitor peoples' eye movements while on the internet (Papoutsaki et al., 2016). Until now, it has not been used in behavioral research, except in one methods article demonstrating some very basic gaze properties (Semmelmann & Weigelt, 2018). The methods in that article suggest that an extensive calibration and validation procedure is necessary (~50% of the study time) and that temporal resolution is low and inconsistent. This has likely dissuaded many researchers from using WebGazer.

Next, we integrate WebGazer into a user-friendly, open-source psychology toolbox called jsPsych. This addresses potential concerns about the difficulty of incorporating WebGazer into existing behavioral paradigms. Using our toolbox, researchers without programming expertise should still be able to add eye-tracking to their experiments.

Finally, we develop procedures for smoothly running eye-tracking experiments online and ensuring good data quality. It is important to ensure that subjects have adequate equipment and are set up to maximize the quality of the eye-tracking data. It is also important that subjects

understand that they are not being recorded and so there are no privacy violations as the images and video do not leave the subject's computer.

To illustrate the features of our WebGazer toolbox, we describe a simple decision-making experiment (Krajbich et al., 2010) conducted on Amazon Mechanical Turk (MTurk). Notably, this experiment took just a couple of days to run, in contrast to standard eye-tracking experiments which typically take several weeks to run. In the supplementary material we provide a template experiment and all of our experimental materials for others to use.

We also note that online eye-tracking is a useful tool for all online researchers, as it can be used to ensure that study subjects are humans and not computer algorithms, i.e. bots (Buchanan & Scofield, 2018; Buhrmester et al., 2011). We hope that this work will facilitate the continued growth of both eye-tracking and online behavioral research.

Results

Basic description of the experiment

To illustrate the use of our online eye-tracking toolbox, we replicated a simple decision-making experiment (Krajbich et al, 2010). In this food-choice experiment, subjects first rated how much they would like to eat 70 different snack foods, then in 100 trials they decided which of two snack foods they would prefer to eat (Fig. 1b).

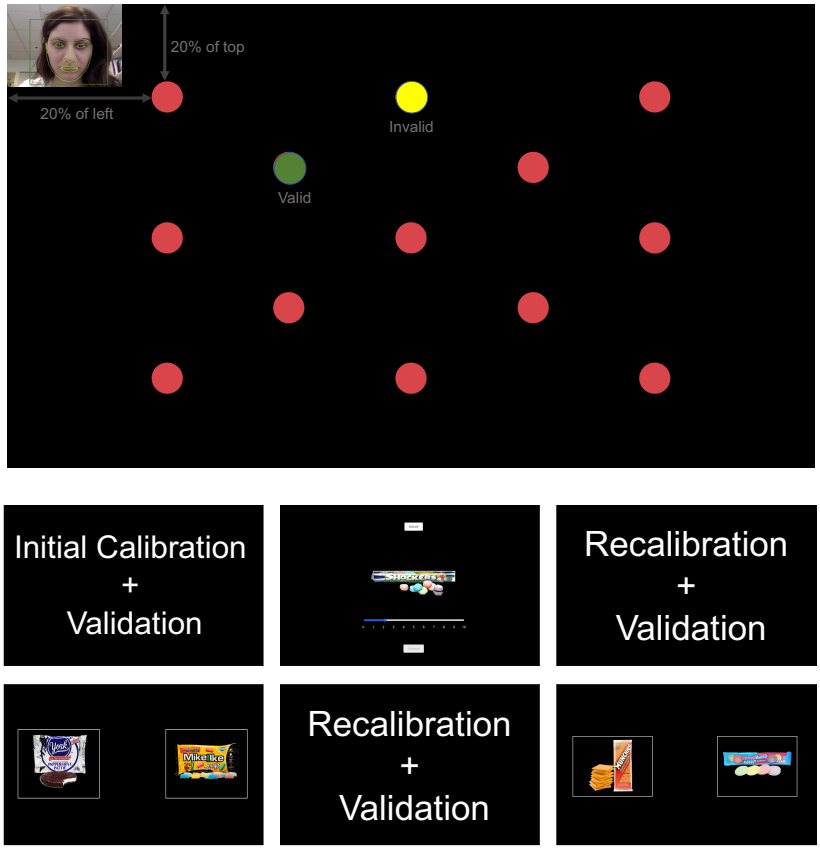


Figure 1. Experiment Design. a). Visualization of the calibration + validation process. Subjects would only see one dot at a time. During calibration only, the subject's face was present at the top left corner of the screen, along with a green box for positioning. During validation only, the dots would change color to indicate a valid or invalid measure. b). Overview of the experiment. There was an initial calibration + validation phase to screen out problematic subjects. Next, subjects rated how much they liked 70 different food items. Then there was another calibration + validation. This was followed by 100 binary-choice trials where subjects chose which food they preferred; there was a recalibration + validation halfway through these trials.

Basic setup and data quality

To begin, it is worth briefly describing a standard eye-tracking procedure in the brick-and-mortar lab. Typically, the eye-tracking camera is situated either below or above the computer screen, between the screen and the subject. The subject is seated, often with their head immobilized in a chinrest, though not always. Either way, subjects are instructed to try to keep their heads still during the experiment. Before the experiment begins, subjects go through a calibration procedure in which they stare at a sequence of dots that appear at different locations on the screen (Fig. 1a). A subsequent validation procedure has the subject look at another sequence of dots, to establish how well the eye-tracker's estimate of the gaze location aligns with where the subject is supposed to be looking (i.e. the dots). During the experiment, validation can be repeated (to varying degrees) to ensure that the eye-tracker is still accurate.

With WebGazer we used a similar procedure, with some qualifications. First, before signing up for the experiment, we required subjects to be using a laptop with a webcam, and to be using an appropriate web browser (see Methods). We also asked them to close any applications that might produce popups. We had no control over the subject's environment and we could not immobilize their head, but we did provide them with a number of suggestions for how to optimize performance, including keeping their heads still, avoiding sitting near windows, keeping light sources above or in front of them rather than behind them, etc. (see Methods). Subjects had three chances to pass the calibration and validation procedure, otherwise the experiment was terminated, and they received a minimal "showup" fee (see Methods).

During the experiment, we incorporated a small number of validation points into the inter-trial intervals, rather than periodically having a full procedure with many validation points. This step is not generally a requirement, but it allowed us to evaluate data quality over time. We

did recalibrate halfway through the choice task. The time interval between the calibration at the beginning of the choice task and the second calibration was 9.00 minutes on average (SD = 4.43 mins).

Prior work has documented the spatial resolution of WebGazer (Semmelmann & Weigelt, 2018). They established that, shortly after calibration and validation, online precision is comparable to, but slightly worse than that in the lab (online: 18% of screen size, 207px offset; in-lab: 15% of screen size, 172px offset). However, an unresolved issue is whether that spatial resolution persists as time goes on.

To assess spatial resolution over time, we examined the hit ratio for validation dots as the experiment went on. For each measurement, we calculated the Euclidean distance (in pixels) between the recorded gaze location and the center of the validation dot. If this distance was below a critical threshold (see Methods), we labeled the measurement a hit, otherwise we labeled it a miss. The hit ratio is simply the proportion of hits out of all the validation measurements (see Methods). Aside from an initial drop shortly after each calibration/validation, the hit ratio remained quite steady over time (Fig. 2a; mean hit ratio as a function of trial number: $\hat{\beta} = -0.00057$, $p = 0.023$). Table S3 shows the mean/median hit ratios for every intertrial validation.

A second, potentially more serious issue is temporal resolution over time. Eye-tracking setups often come with dedicated computer hardware due to the required computations. With online eye-tracking, there is no second computer and we have little control over subjects' hardware. If the computations overwhelm the subjects' hardware, the temporal resolution may suffer dramatically.

To assess temporal resolution over time, we examined the average time interval between gaze estimates made by WebGazer as the experiment went on. As we feared, an earlier pilot

experiment revealed that the time interval between estimates increases dramatically over time, from 95ms (SD = 13ms) in the first ten trials, to 680 ms (SD = 64ms) by the halfway point (13.20 min (SD = 3.55 min)). This decreased back to 99ms (SD = 12ms) after recalibration but then increased to 972ms (SD = 107ms) by the end of the experiment. This kind of time resolution is unacceptable for most behavioral work.

With some modifications to the WebGazer code (see Methods) we were able to reduce computational demands. As a result, the time interval between estimates in our main experiment remained steady at 24.95ms on average (SD = 12.46ms) throughout the experiment (Fig. 2b). This time resolution is comparable to many in-lab eye-trackers currently on the market and in scientific use (Carter & Luke, 2020).

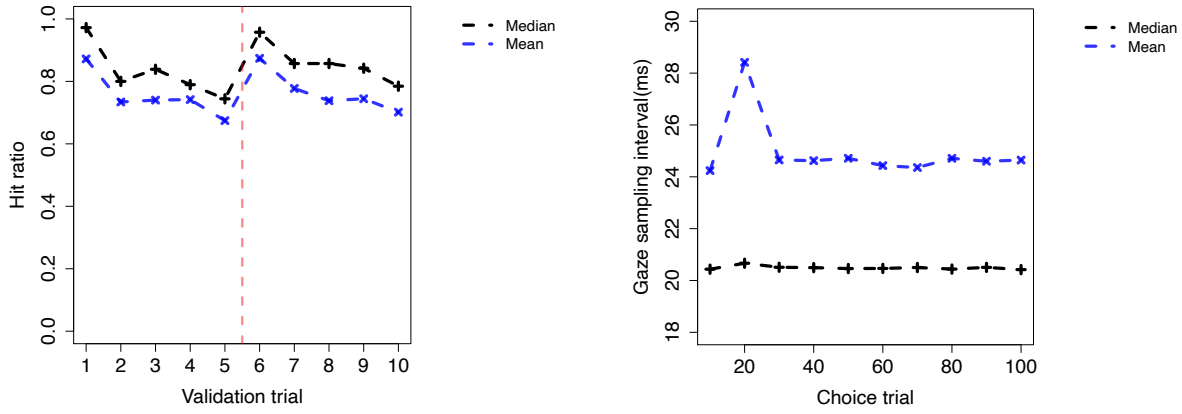


Figure 2. Spatial accuracy (a) and temporal resolution (b) over time. (a) The hit ratio, namely the proportion of successful intertrial validation points, as a function of number of validations completed. 10 intertrial validation trials were included per participant. The vertical orange line represents the recalibration halfway through the experiment. (b) The gaze sampling interval, namely the delay between gaze measurements, as a function of the number of choice trials completed. The black and blue curves indicate the median and mean values, respectively.

JsPsych integration

JsPsych is a popular toolbox for conducting web-based psychology experiments (De Leeuw, 2015). It is built on JavaScript, including a library of commands for behavioral experiments, but it also allows for integration of JavaScript-based libraries such as WebGazer. Therefore, we adapted a jsPsych plugin for eye-calibration and eye-validation. In this way, when an experimenter wishes to use Webgazer, they can simply load the plugin, specify the desired parameters, and then initialize the eye-tracking process. The eye-tracking data are recorded synchronously with other response measures, so no extra steps are needed to align the data.

In the supplementary material we provide an example initialization of WebGazer (Figure S4). The timeline includes the eye-tracking instructions and calibration setup. This is all that is required to begin eye-tracking online.

Privacy and bots

Given that WebGazer uses subjects' webcams to monitor their gaze location, privacy concerns naturally arise. It is therefore important to note, and to highlight for subjects, that the webcam images are processed on their own computers. The video/images never leave the subjects' computers. What leaves their computer is the output of the WebGazer algorithm, namely horizontal (x) and vertical (y) coordinates of where WebGazer thinks the subject is looking at a given point in time.

On the other end of the spectrum, WebGazer does give us some information about our subjects. Namely, it requires that subjects have a laptop, a webcam, a compatible browser, and most importantly, that they are human. A common issue with online studies is ensuring that subjects are human and not computer "bots". Researchers have developed ways to filter out bot data after the fact (Dupuis et al., 2019) or to use extra items to screen out bots during the study

(Buchanan & Scofield, 2018). The problem with the former approach is that it requires assumptions about how these bots will respond. Savvy Mturk users can program bots that violate those assumptions. The latter approach is more similar to ours, but it typically requires participants to exert extra effort that is irrelevant to the task, and these extra measures may also be defeated by savvy programmers.

WebGazer provides a simple way to ensure that subjects are human beings, without any additional questions or statistical tests. While it is surely not impenetrable, faking eye-tracking data would be no small feat.

Analysis of the dataset

To verify the quality of online eye-tracking, we sought to replicate the central analyses from the original experiment that we replicated (Krajbich et al, 2010). This experiment was originally run with an eye-tracker with comparable time resolution of 20ms. In that version, subjects first rated 70 snack foods, then in 100 trials decided which of two snack foods they would prefer to eat. Our online replication of that experiment was identical except for the particular stimuli, the number of trials, and the fact that the decisions were hypothetical.

In the original experiment, accuracy rates for rating differences of {1, 2, 3, 4 ,5} were {0.65, 0.76, 0.84, 0.91, 0.94}; in the MTurk study they were {0.65, 0.79, 0.87, 0.90, 0.92}. Thus, despite being hypothetical, decisions in the MTurk study were very similar in quality. Response times (RT) in the original study declined with absolute value difference from 2.55s to 1.71s. Similarly, RTs in the MTurk study declined from 1.42s to 1.17s. RTs in the MTurk study were significantly shorter than Krajbich et al. (2010)'s in-lab study (RTs as a function of value difference and experimental condition (MTurk or in-lab), $t = 17.910$, $p = 10^{-}$

[16]) .So, while MTurk respondents were considerably faster in their decisions, they still exhibited the expected relationship between difficulty and RTs (mixed effects regression of log(RT) on absolute value difference: $\hat{\beta} = -0.029$, $p = 10^{-9}$).

Next, we turn to the eye-tracking data. Prior research has modeled this task with the attentional drift diffusion model (aDDM). This work has established a number of patterns in the eye-tracking data. Key relationships that we sought to replicate here include correlations between dwell times and choice, and between the last fixation location and choice.

The first analysis models the choice (left vs. right) as a function of rating difference (left – right) and total dwell time difference (left – right) over the course of the trial, using a mixed-effects logistic regression. We found a strong significant effect of relative dwell time ($\beta = 0.76$, $p = 10^{-5}$), even after accounting for item ratings (Fig. 3a).

The second analysis examines the effect of individual dwells. Here we model the choice (first-seen vs. other) as a function of the rating difference (first – other) and the duration of the first dwell, again with a mixed-effects logistic regression. We again find a significant effect of the initial dwell time ($\beta = 0.33$, $p = 0.033$), even after accounting for the item ratings (Fig. 3b).

The third analysis examines the effect of the final fixation location. Here we model the choice (last seen vs. other) as a function of the rating difference (last seen– other), again with a mixed-effects logistic regression. We find a strong significant intercept term ($\beta = 0.23$; $p = 10^{-5}$), indicating a bias to choose the last-seen item (Fig. 3c). However, this last-fixation effect is noticeably smaller in this dataset compared to the original dataset.

One noticeable difference between this dataset and the original in-lab results (Krajbich et al., 2010) is in the duration of the average dwell. However, this may reflect that RTs were considerably shorter in this experiment than in the lab experiment. The average dwell time, as a

fraction of RT, was comparable between the lab (M = 0.28, SD = 0.24) and MTurk (M = 0.25, SD = 0.15) experiments.

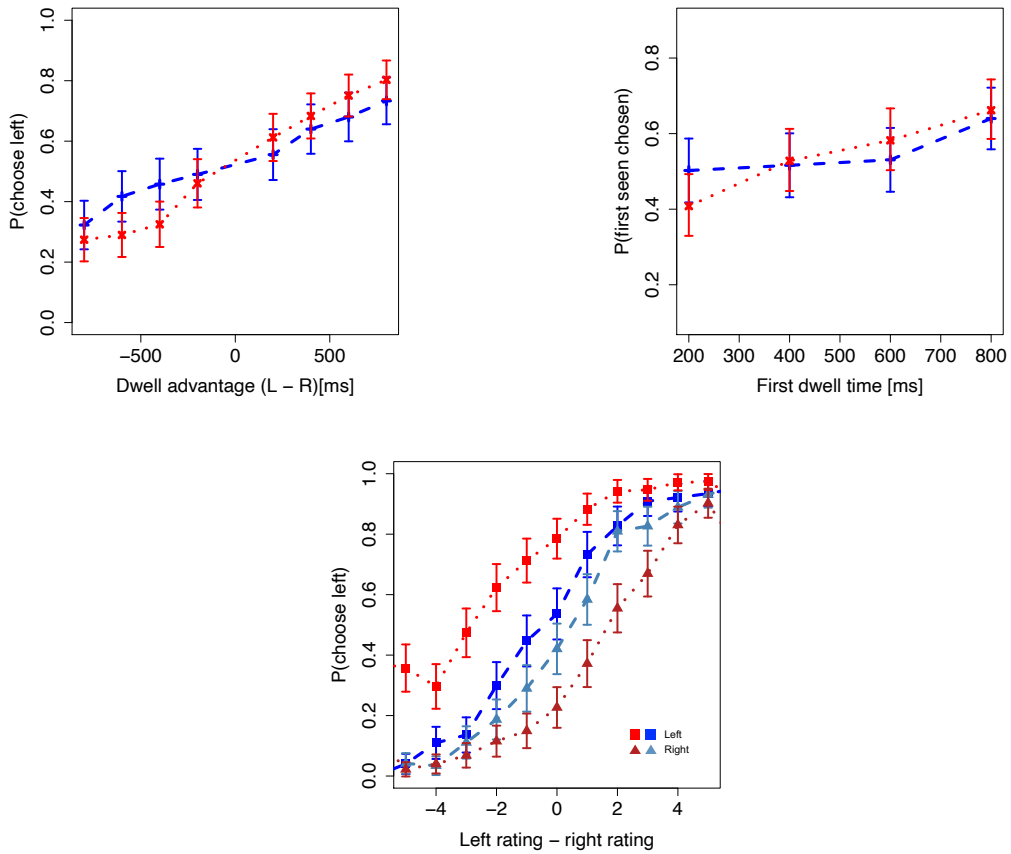


Figure 3. Relations between gaze and choice. A) Choice as a function of the total dwell-time difference between the left option and the right option in a given trial. B). Choosing the first seen item as a function of the first gaze dwell time. C). Choice as a function of the value differences between the two options, split by the location of the last fixation. In each plot, the red line/dots represent the results in Krajbich et al. (2010)'s dataset; the blue line/dots represent the results in the online MTurk study.

Discussion

We have presented a methodology for online eye-tracking in behavioral research. Online data collection is increasingly common, especially during the COVID-19 pandemic. This should not be a barrier to studying visual attention.

Although there are some options available for online eye-tracking, none have been adopted by behavioral researchers. Some software (e.g., TurkerGaze) requires extensive programming knowledge. Other software such as Realeye (https://www.realeye.io) is not open source and can be very expensive to use. In general, when trying to build an online eye-tracking experiment there are several features to consider: 1). The flexibility of stimulus presentation (is it possible to adjust the paradigm/software for different experiments?) 2). The difficulty of the experimental programming (does the implementation of the paradigm/software require extra expertise?) 3). The retrieval of the eye-tracking data (can the data be retrieved and stored in a useable format?) 4). The accessibility of the resources (is the software/paradigm open-source?). Our WebGazer toolbox performs well on all these dimensions, as it provides total flexibility, is integrated in user-friendly jsPsych, stores the eye-tracking data with the other behavioral measures, and is open-source.

An important issue that we addressed in this study is the amount of calibration and validation required to run a successful experiment. In prior work, calibration and validation has taken up to 50% of the experiment time (Semmelmann & Weigelt, 2018). However, with our modifications, we found that it is possible to get by with a lot less, as there appears to be little to no degradation in spatial or temporal precision over time, at least on the time scale of our experiment. Moreover, we found that most participants were able to pass the initial calibration in their first attempt, minimizing the time that they spend on calibration and validation (see Supplementary Note 1). Going forward, we would suggest assigning a single calibration +

validation phase at the beginning of the study (to screen out unusable subjects), one recalibration just before the eye-tracking trials begin, and additional recalibrations every ~15 minutes (though this last step may not be necessary). Occasional intertrial validation dots may also be useful as a measure of data quality. Of course, the amount of calibration should depend on the spatial precision required. If there are more areas of interest (AOI) then more calibration may be necessary.

Along those lines, one unresolved issue is how many distinct AOIs can be effectively used online. Here we used a simple design with two AOIs. Based on WebGazer's spatial precision, we estimate that one could use six AOIs without any degradation in data quality. More than that and gaze in one AOI might start to register in another AOI. As stated in Supplementary Note 3, we found spatial offsets in the range of 102.08 px – 194.03 px. These offsets are moderately better than those reported in Papoutsaki et al. (2016) and Semmelmann & Weigelt (2018), perhaps due to our adjustments to the code or our additions to the calibration + validation instructions. This offset range would be compatible with approximately six AOIs for most current laptops. Presumably, better data analysis methods could be used to filter out spurious data points, if one needed more AOIs.

Another issue is how far the time resolution can be pushed. Here we went with 50 Hz, which seemed to work very well. We could likely increase the sampling rate. Note that we used a different prediction method than in prior work (Semmelmann & Weigelt, 2018) (see Method). The prediction method we used seems better suited to eye-tracking research where the experimenters can control the sampling rate. However, at some point a sampling rate that is too high may overtax some hardware and cause degradation in resolution over time, as we observed

in our pilot study. Thus, it may be better to play it safe and use something close to 50 Hz, unless better resolution is required.

Previous research has documented the advantages and disadvantages of conducting behavioral research online (i.e., Mason & Suri, 2012). We would like to highlight several benefits of online eye-tracking compared to in-lab eye-tracking. First, tasks on MTurk are synchronous, which means that multiple subjects can participate in the study simultaneously. In contrast, in-lab eye-tracking studies typically are one-on-one sessions, with one participant and one experimenter in the laboratory (but see Hausfeld et al. 2020). Therefore, collecting data in the lab is time-consuming and requires much effort from experimenters to assist with the calibration + validation process. In particular, we completed data collection in three days for this simple 30-minute online eye-tracking study, while it could take many weeks or months to finish equivalent data collection in the lab. Second, the low cost of online eye-tracking is also another distinct advantage, as it requires no special hardware on the experimenter's side and the software involved is all free and open access. On the other hand, in-lab eye-tracking still offers the most controlled environment with the highest spatial and temporal resolution, for those who need it. It is up to the researchers to determine whether online eye-tracking is right for them.

Methods

Subjects

125 subjects from Amazon MTurk participated in this study. Of these, 48 successfully passed the initial calibration + validation and completed the study. The Ohio State University Institutional Review Board approved the experiment and subjects provided informed consent prior to the study. Subjects received $7 for completing the study. We required subjects to be located in the United States and have a 95% or higher HIT approval rate. In addition, we required subjects to have a laptop with a webcam.

Experimental Software/materials

The experiment was programmed in JavaScript, based on the jsPsych and WebGazer libraries. To improve WebGazer's temporal resolution we removed some seemingly unnecessary computations that occur in each animation frame of a webpage. The original code calls the getPrediction() function at every animation frame to load the measured gaze location. This step is necessary when providing gaze-contingent feedback, but otherwise just consumes computational resources. These extra computations appear to gradually degrade WebGazer's temporal resolution.

To deal with this, we modified the loop() function for each animation frame to avoid the getPrediction() call when possible (for the case we just need face tracking data to draw face overlay, the CLM tracker is called separately, and similarly for pupil features needed in the face feedback box).

In addition, we also used the recently added ridge thread regression method, which reduces computational demands.

Task

The study included three parts:

1). Recruitment and initial preparations

We asked subjects to close any unnecessary programs or applications on their computers before they began. Also, we asked them to close any browser tabs that could produce popups or alerts that would interfere with the study (see Figure S3). Once the study began, subjects entered into full-screen mode.

Before subjects began the calibration/validation process, we provided detailed instructions about how to position themselves. We first showed them instructions from Semmelmann & Weigelt (2018). For example, they should sit directly facing the webcam to ensure full visibility of their face. We also added several tips we learned from the pilot study. In detail, we asked subjects to 1). use their eyes to look around the screen and avoid moving their head; 2). keep lights in front of them rather than behind them so that the webcam could clearly see their faces; 3). avoid sitting with a window behind them (Figure S2).

After reading the instructions, subjects saw a screen where they could position themselves appropriately using the live feed from their webcam. Once they were properly positioned, they could advance to the calibration and validation stage.

2). Calibration + validation

Subjects next had to pass an initial calibration + validation task. At the beginning of the calibration, a video feed appeared in the top left corner of the screen. Subjects could use this video feedback to adjust their position and center their face in a green box in the center of the

video display. Once properly positioned, subjects could press the space bar to advance to the next step.

Next, subjects saw a sequence of 13 calibration dots appear on the screen, each for three seconds (Semmelmann & Weigelt, 2018). The task was simply to stare directly at each dot until it disappeared.

Next, subjects entered the validation procedure. The validation procedure was essentially identical to the calibration procedure, except for the following differences. Each validation dot lasted for two seconds. Within those two seconds, WebGazer made 100 measurements (one every 20ms). Each measurement was labeled as a hit if it was within X pixels of the center of the dot (X increased with each failed calibration/validation attempt, see below). If at least 80% of the measurements were hits, we labeled the dot as valid, and it turned green. Otherwise, the dot turned yellow (in the validation instructions, we told subjects to try to make every dot turn green). Out of 13 validation dots, if the valid dot proportion was at least Y, the experiment proceeded.

Subjects had three chances to pass this initial calibration + validation task. With each new attempt, we increased the pixel threshold (X) for a hit and the valid-dot threshold (Y). In particular, the pixel thresholds (X) were: 130px, 165px, and 200px; the valid-dot thresholds were: 80%, 70%, and 60%. If a subject failed the calibration + validation three times, we compensated them with 50 cents and ended the experiment.

We adopted this procedure to give poorly calibrated subjects a chance to reposition themselves and try again, while also acknowledging that some subjects might not be able to sufficiently improve their setup to pass the most stringent requirements. This also allowed us to

assess if initial calibration attempt(s) predicted any of the later results (see Supplementary Note 1).

3). Hypothetical food choice task.

After passing the initial calibration and validation, subjects proceeded to the choice task. The design was very similar to Krajbich et al. (2010). Subjects first rated their desire for 70 snack food items on a discrete scale from 0 to 10. Subjects were told that 0 means indifference towards the snack, while 10 indicates extreme liking of the snack. They could also click a "dislike" button if they didn't like a food item. Subjects used the mouse to click on the rating scale.

After the rating task, subjects were recalibrated and validated. They were eye-tracked for the remainder of the study.

Next, subjects began the binary choice task. 100 trials were randomly generated using pairs of the rated items, excluding the disliked items. Subjects were told to choose their preferred food in each trial. They selected the left option by pressing the left arrow key and the right option by pressing the right arrow key.

Between trials, subjects were either presented with a fixation cross at the center of the screen or, every ten trials, with a sequence of three red validation dots. In the latter case, the first two validation dots appeared randomly at one of 12 possible positions, while the last dot always appeared at the center of the screen. For each of those validation dots, the pixel threshold was set at 130px with a threshold of 70%, and the presentation time was 2 seconds. A recalibration would be triggered if subjects failed more than four validation dots in two successive intertrial validation. Subjects were removed from the analysis if they failed 50% or more intertrial validation dots. We excluded 14 subjects from the analysis based on this criterion.

After 50 trials, subjects were given the option to take a short break. After the break, they were recalibrated and validated.

References

Buchanan, E. M., & Scofield, J. E. (2018). Methods to detect low quality data and its implication for psychological research. *Behavior Research Methods*, *50*(6), 2586–2596. https://doi.org/10.3758/s13428-018-1035-6

Buhrmester, M., Kwang, T., & Gosling, S. D. (2011). Amazon's Mechanical Turk: A New Source of Inexpensive, Yet High-Quality, Data? *Perspectives on Psychological Science*, *6*(1), 3–5. https://doi.org/10.1177/1745691610393980

Carter, B. T., & Luke, S. G. (2020). Best practices in eye tracking research. *International Journal of Psychophysiology*, *155*, 49–62. https://doi.org/10.1016/j.ijpsycho.2020.05.010

de Leeuw, J. R. (2015). jsPsych: A JavaScript library for creating behavioral experiments in a Web browser. *Behavior Research Methods*, *47*(1), 1–12. https://doi.org/10.3758/s13428-014-0458-y

Dupuis, M., Meier, E., & Cuneo, F. (2019). Detecting computer-generated random responding in questionnaire-based data: A comparison of seven indices. *Behavior Research Methods*, *51*(5), 2228–2237. https://doi.org/10.3758/s13428-018-1103-y

Fellows, L. K. (2006). Deciding how to decide: Ventromedial frontal lobe damage affects information acquisition in multi-attribute decision making. *Brain: A Journal of Neurology*, *129*(Pt 4), 944–952. https://doi.org/10.1093/brain/awl017

Goodman, J. K., & Paolacci, G. (2017). Crowdsourcing consumer research. *Journal of Consumer Research*, *44*(1), 196–210. https://doi.org/10.1093/jcr/ucx047

Hausfeld, J., von Hesler, K., & Goldlücke, S. (2020). Strategic gaze: An interactive eye-tracking study. *Experimental Economics*.

Just, M. A., & Carpenter, P. A. (1984). Using eye fixations to study reading comprehension. *New Methods in Reading Comprehension Research*, 151–182.

Krajbich, I. (2019). Accounting for attention in sequential sampling models of decision making. *Current Opinion in Psychology*, *29*, 6–11. https://doi.org/10.1016/j.copsyc.2018.10.008

Krajbich, I., Armel, C., & Rangel, A. (2010). Visual fixations and the computation and comparison of value in simple choice. *Nature Neuroscience*, *13*(10), 1292.

Majaranta, P., Räihä, K.-J., Hyrskykari, A., & Špakov, O. (2019). Eye Movements and Human-Computer Interaction. In C. Klein & U. Ettinger (Eds.), *Eye Movement Research: An Introduction to its Scientific Foundations and Applications* (pp. 971–1015). Springer International Publishing. https://doi.org/10.1007/978-3-030-20085-5_23

Mason, W., & Suri, S. (2012). Conducting behavioral research on Amazon's Mechanical Turk. *Behavior Research Methods*, *44*(1), 1–23. https://doi.org/10.3758/s13428-011-0124-6

Mormann, M., Griffiths, T., Janiszewski, C., Russo, J. E., Aribarg, A., Ashby, N. J. S., Bagchi, R., Bhatia, S., Kovacheva, A., Meissner, M., & Mrkva, K. J. (2020). Time to pay attention to attention: Using attention-based process traces to better understand consumer decision-making. *Marketing Letters*. https://doi.org/10.1007/s11002-020-09520-0

Noguchi, T., & Stewart, N. (2014). In the attraction, compromise, and similarity effects, alternatives are repeatedly compared in pairs on single dimensions. *Cognition*, *132*(1), 44–56. https://doi.org/10.1016/j.cognition.2014.03.006

Papoutsaki, A., Sangkloy, P., Laskey, J., Daskalova, N., Huang, J., & Hays, J. (2016). Webgazer: Scalable webcam eye tracking using user interactions. *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence-IJCAI 2016*.

Polonio, L., Di Guida, S., & Coricelli, G. (2015). Strategic sophistication and attention in games: An eye-tracking study. *Games and Economic Behavior*, *94*, 80–96. https://doi.org/10.1016/j.geb.2015.09.003

Semmelmann, K., & Weigelt, S. (2018). Online webcam-based eye tracking in cognitive science: A first look. *Behavior Research Methods*, *50*(2), 451–465. https://doi.org/10.3758/s13428-017-0913-7

Teoh, Y. Y., Yao, Z., Cunningham, W. A., & Hutcherson, C. A. (2020). Attentional priorities drive effects of time pressure on altruistic choice. *Nature Communications*, *11*(1), 3534. https://doi.org/10.1038/s41467-020-17326-x

Valliappan, N., Dai, N., Steinberg, E., He, J., Rogers, K., Ramachandran, V., Xu, P., Shojaeizadeh, M., Guo, L., Kohlhoff, K., & Navalpakkam, V. (2020). Accelerating eye movement research via accurate and affordable smartphone eye tracking. *Nature Communications*, *11*(1), 4553. https://doi.org/10.1038/s41467-020-18360-5

Xu, P., Ehinger, K. A., Zhang, Y., Finkelstein, A., Kulkarni, S. R., & Xiao, J. (2015). TurkerGaze: Crowdsourcing Saliency with Webcam based Eye Tracking. *ArXiv:1504.06755 [Cs]*. http://arxiv.org/abs/1504.06755

Supplementary materials

Supplementary Notes

1. Calibration & validation

   For participants who passed the initial calibration, 67% (32 out of 48) of them passed the calibration at their first attempt, 23% (11 out of 48) of them passed the calibration at their second attempt, and 10% (5 out of 48) need the third attempt. However, there's no evidence that the initial calibration performance affected the later spatial accuracy (quantified by the averaged hit ratio across the intertrial validation dots; $F(2, 45) = 0.62$, $p = 0.54$).

2. Relationship between rating difference vs. RT and choice

   We examined the relationship between value differences and choice probability, and response times (See Fig. S1). Consistent with Krajbich et al. (2010), we found that response time and choice accuracy are functions of the choice difficulty (mixed effects regression of choice probability on value differences: $\beta = 0.64$, $p = 10^{-16}$ for the Mturk study). However, response times in the MTurk study were significantly shorter than Krajbich et al (2010) as we detailed in the main text.

3. Validation dot offsets

   We recorded all the validation gaze prediction samples in another study (reported elsewhere). Here we summarize the sample mean and sample deviation for each validation dot (see Supplementary Table 2). We found offsets in the range of 102.08 px – 194.03 px. However, the validation dots at the corners of the screen had significantly larger offsets than the other dots (mixed effects regression of offsets on the validation dot

position (at the corner vs. not at the corner): $\hat{\beta} = 45.06$, $p = 10^{-4}$). Samples of those dots also showed higher variance.

Supplementary Tables

Calibration + validation attempts	Pixel level (.px)	Threshold	Subject proportion
1	130	80%	0.67
2	165	70%	0.23
3	200	60%	0.10

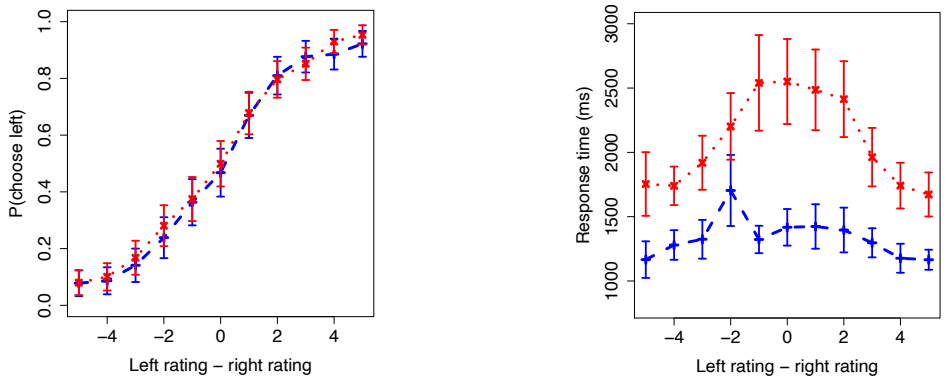
Supplementary table 1. This table summarized the statistics related to three different initial calibration + validation attempts. Pixel level represents the maximum Euclidean distance between the prediction and gazed validation dot. Threshold represent the smallest proportion of the valid dots out of all validation dots in order to pass the initial calibration. Subject proportion represents proportion of the subjects who pass the calibration at the corresponding attempt.

Condition	Mean(distance in px)	SD (distance in px)
20%; 20%	160.49	189.78
20%; 50%	155.54	190.72
20%; 80%	160.47	183.16
50%; 20%	102.08	110.25
50%; 80%	127.44	129.89
80%; 20%	180.07	215.03
80%; 50%	171.79	210.45
80%; 80%	194.03	208.46
35%; 35%	127.33	129.34
65%; 35%	130.96	149.98
35%; 65%	104.09	122.11
65%; 65%	116.44	147.98
50%; 50%	118.26	121.18

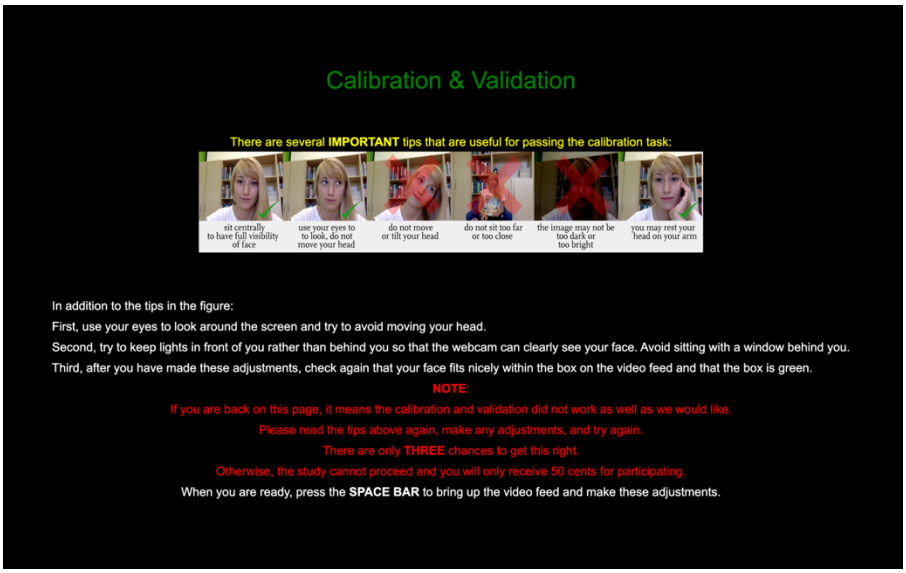
Supplementary table 2. This table summarized the statistics related to validation samples in another study (reported elsewhere). Each condition represents one validation dot position on the screen (relative to the screen size). For example, 20%;20% represents 20% of the screen width and 20% of the screen height, with the origin at the top left of the screen. Each validation sample represents a single gaze measurement produced by WebGazer. Ideally, WebGazer would give a measurement every 20ms. Mean distances represent the average Euclidean distance between the measured gaze location and the validation dot across all samples. Standard deviations are calculated for each condition using all validation samples for that condition.

Hit ratio	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th
Mean	0.87	0.73	0.74	0.74	0.67	0.87	0.79	0.74	0.75	0.70
median	0.97	0.80	0.84	0.79	0.74	0.96	0.86	0.86	0.84	0.78

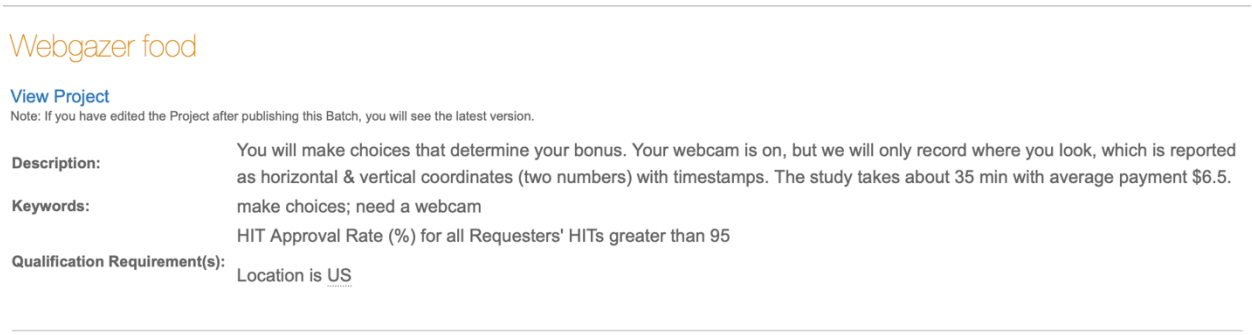
Supplementary table 3. This table summarizes the statistics related to intertrial validation dots. There were 10 intertrial validations in total. For each validation, there were three dots, and for each dot there were multiple gaze measurements (approximately 150). For each subject, we computed the proportion of hits for each validation. Mean hit ratio represents the average of that measure, across subjects. Median hit ratio represents the median of that measure across subjects.



Supplementary figure 1. A). relationship between choice accuracy and value difference. B). relationship between response times and value difference. In each plot, the red line/dots represent the results in Krajbich et al. (2010)'s dataset; the blue line/dots represent the results in the current online MTurk study.



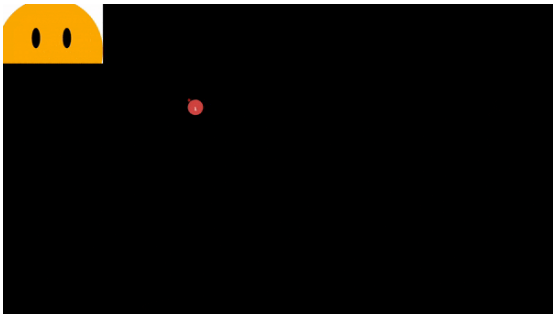
Supplementary figure 2. Eye-tracking instructions shown to the participants.



Supplementary figure 3. MTurk experiment description and requirements.

```
var inital_eye_calibration = {
  timeline: [
    eyeTrackingNote, // instruction
    {
      type: "eye-tracking",
      doInit: true,
      doCalibration: true,
      doValidation: true,
      calibrationDots:  13,
      calibrationDuration: 3,
      doValidation: true,
      validationDots:  13,
      validationDuration: 2,
      validationTol: 130,
     ],
};
```

Supplementary figure 4. An instance of initializing the eye-tracking process with the template.



Supplementary Figure 5. Visualization of the calibration + validation task.

4. **Online tutorial & experiment demo**
   https://github.com/xiaozhi2/webgazertutorial