Supplementary Materials: Method for Correcting Fixation Positions

Anonymous
Anonymous

General Procedure

In the following, we describe the correction procedure applied to each trial prior to analysis. Each trial was considered separately, and the goal of the procedure was to move each fixation to a new position. More formally, each trial was composed of fixations $f_i = (x_i, y_i),\ i = 1 \ldots n$, given by a horizontal coordinate x_i (lower values of x appear on the left of the screen; higher values appear on the right) and a vertical coordinate y_i (lower values of y appear at the top of the screen; higher values appear at the bottom). We denote by $F = \{f_1, \ldots, f_n\}$ the set of all fixations performed in a given trial. Our goal is therefore to define a mapping function $M : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ such that $M(f)$ is a "corrected" fixation, i.e., the value output by $M(f)$ is a new fixation, with coordinates x and y assigned to a new position that is assumed to be closer to the real position at which the participant was *actually* looking when the fixation was recorded.

Each critical trial was presented in two text lines: the first one containing the preamble and the critical segment; the latter containing two regions of interest and the wrap up segment. The correction procedure involved fitting a regression model, yielding two regression lines. To avoid confusion, in the discussion below, we use the wording *text line* to refer to the lines of text in which the sentence is displayed on the screen; and *regressed/regression line* to refer to the output of a linear model (see below), described by a slope and an intercept.

Surrounding each word we also defined an interest area so that any fixation inside that interest area was considered to be associated with that word. Figure 1 shows an example trial, along with the interest areas associated with each word. To simplify the description of the algorithm, we also define a number of variables in Table 1, which are also depicted in Figure 1. Basically, these variables indicate the x position of the first letter, and the y coordinates of the text lines. In addition, before applying any correction, we discard all fixations that are too far away from the text (i.e., whose y coordinate is higher than y_o).

Since all trials were presented in two text lines, and assuming that participants were really reading the sentences, a typical trial is expected to be composed of fixations that mimic those text lines, even if they are not positioned exactly along them. In an ideal scenario in which participants were able to look directly at each text letter as they were reading it, we would expect the fixations to coincide perfectly with the text, forming two horizontal lines with slope 0 and intercepts y_{l1} and y_{l2}, respectively (see Table 1). Following these assumptions, in our correction procedure we fit two regression lines simultaneously,

Table 1
Variables used in the correction procedure. The y variables represent a coordinate of the vertical axis; the only x variable represents a coordinate of the horizontal axis

Variable	Description
y_{l1}	The y coordinate where the words appear in the first text line
y_{l2}	The y coordinate where the words appear in the second text line
y_o	Fixations with y value higher than y are considered outliers and discarded prior to correction. We arbitrarily set $y_o = 450$, as this was already quite far from the text lines (see Figure 1).
x_{w1}	The x coordinate at the center of the first interest area

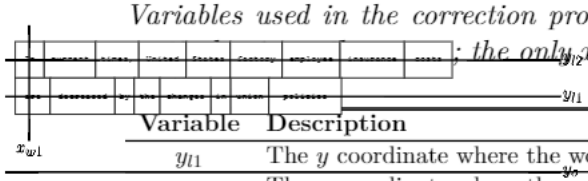


Figure 1. The position of the variables defined in Table 1. The trial is the same as that of Figure 2.

denoted by the equations $y_{r1} = a_{r1}x + b_{r1}$ and $y_{r2} = a_{r2}x + b_{r2}$, with parameters a_{r1} and b_{r1} corresponding to the slope and intercept of the first regression line; and parameters a_{r2} and b_{r2} corresponding to the slope and intercept of the second regression line, respectively. To find the values of these parameters, we associate each fixation with one of the two lines (y_{r1} or y_{r2}). For ease of notation, we define the two subsets of F, $R_1 = \{f | f \in F, f \text{ is associated with } y_{r1}\}$ and $R_2 = \{f | f \in F, f \text{ is associated with } y_{r2}\}$, so that $R_1 \cup R_2 = F$ (recall that F is the set of all fixations in a given trial). We then calculate the vertical distance of each fixation to its associated line as

$$L_f = \begin{cases} d_{r1,f}, & \text{if } f \in R_1 \\ d_{r2,f}, & \text{if } f \in R_2, \end{cases}$$

where $d_{r1,f} = |a_{r1}x + b_{r1} - y|$, and $d_{r2,f} = |a_{r2}x + b_{r2} - y|$. Finally, we minimize the total loss L,

$$L = \sum_{\forall f \in F} L_f,$$

using the Gradient Descent method, in order to arrive at the final values of a_{r1}, b_{r1}, a_{r2} and b_{r2}.

Having found the regressed lines y_{r1} and y_{r2}, we are now able to calculate how different they are from the ideal lines discussed above. That is, we should expect both regression lines to have a slope of 0, the first regression line to have an intercept of y_{l1}, and the second regression line to have an intercept of y_{l2}. Thus, for each fixation f we can measure the vertical deviation of its ideal position by calculating the offset $\Delta y_{r1} = y_{l1} - y_{r1}$ if it

Figure 2. A participant's fixations on the trial of Figure 2 as they were originally collected. Note that, while fixation 46 clearly belongs to *costs*, this kind of decision is not as clear for fixation 43.

is associated with regression line y_{r1}, or by calculating the offset $\Delta y_{r2} = y_{l2} - y_{r2}$ if it is associated with the regression line y_{r1}. Finally, we define the $M(f)$ function in the following way:

$$M(f) = \begin{cases} (x, y + \Delta y_{r1}), & \text{if } f \in R_1 \\ (x, y + \Delta y_{r2}), & \text{if } f \in R_2 \end{cases}$$

Associating a fixation f to a regression line

Most of the time, we associate the fixation f to whichever line it is closest vertically to. In other words, we calculate the distances $d_{r_1,f}$ and $d_{r_2,f}$, and define that $f \in R_1$ if $d_{r_1,f} < d_{r_2,f}$, and $f \in R_2$ otherwise. Importantly, we make always sure that $R_1 \cap R_2 = \varnothing$. The description of the gradient descent algorithm below will assume that these are the only rules we use.

However, we additionally have a special rule for fixations positioned far towards the right of the screen. Consider, for example, fixation 46 (circled in dashed lines) of in Figure 2. When considered solely its horizontal coordinate (its x position), even if the regression line y_{r2} were closer than y_{r1}, it would not make sense to assign it to R_2, since there is no text in that region of the screen. While this is clear for fixation 46, this decision is not as easy for fixation 43, which could belong to both R_1 and R_2.

To deal with these special cases, we define two variables $x_{l1}^{\text{last IA}}$ and $x_{l2}^{\text{last IA}}$, indicating the x coordinate of the right edge of the last interest area of the first and the second text line, respectively. Then, for any fixation $f = (x, y)$, we check if it makes sense to assign it to the other regression line. In particular, we do:

if $f \in R_1$ **and** $x > x_{l1}^{\text{last IA}} + \tau$ **and not**$(x > x_{l2}^{\text{last IA}} + \tau)$ **then**
  $R_1 \leftarrow R_1 \setminus f$ ▷ Remove f from R_1
  $R_2 \leftarrow R_2 \cup f$ ▷ Add f to R_2
else if $f \in R_2$ **and** $x > x_{l2}^{\text{last IA}} + \tau$ **and not**$(x > x_{l1}^{\text{last IA}} + \tau)$ **then**
  $R_2 \leftarrow R_2 \setminus f$ ▷ Remove f from R_2
  $R_1 \leftarrow R_1 \cup f$ ▷ Add f to R_1
end if

The variable τ is used as a tolerance to deal with 'grey zone' cases such as that of fixation 43 in Figure 2.

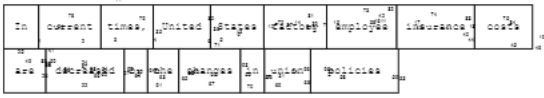


Figure 3. The trial of Figure 2 after correction.

The Gradient Descent algorithm

Let $\theta = \{a_{r1}, b_{r1}, a_{r2}, b_{r2}\}$ be the set of parameters to be optimized. The goal of the loss minimization procedure is to find values for the elements of θ for which the loss is the lowest possible. In other words, we look for:

$$L = \sum_{\forall f \in F} L_f = \arg\min_{\theta} \left(\sum_{f \in R_1} d_{r1,f} + \sum_{f \in R_2} d_{r2,f} \right)$$

As mentioned above, in this description, we assume that each fixation is associated with whichever regression line is closest to it. This changes our loss to

$$\arg\min_{\theta} L = \arg\min_{\theta} \sum_{f \in F} \min \left(d_{r1,f}, d_{r2,f} \right).$$

We use Gradient Descent to calculate these parameters. The Gradient Descent algorithm depends on the calculation of derivatives. Notice, however, that the distances $d_{r_j,f} = |a_{rj}x + b_{rj} - y_{rj}|$ are module functions, and therefore their derivative does not exist in the entirety of their domain. To avoid this discontinuity, we square the distances:

$$L' = \sum_{f \in F} \min \left(d_{r1,f}^2, d_{r2,f}^2 \right).$$

Hence, we seek to find:

$$\arg\min_{\theta} L' = \arg\min_{\theta} \sum_{f \in F} \min \left(d_{r1,f}^2, d_{r2,f}^2 \right).$$

In the discussion below, we denote by $L'(u, v, w, z)$ the evaluation of the loss L' at the point (u, v, w, z), i.e., the value of L' when $a_{r1} = u$, $b_{r1} = v$, $a_{r2} = w$, $b_{r2} = z$. In other words,

$$L'(u, v, w, z) = \sum_{f \in F} \min \left((ux_i + v - y_i)^2, (wx_i + z - y_i)^2 \right).$$

To find the values for the parameters in θ, Gradient Descent starts by initializing the parameters:

$$a_{r_1} \leftarrow a_{r_1}^{\text{initial}} \qquad b_{r_1} \leftarrow b_{r_1}^{\text{initial}} \qquad\qquad a_{r_2} \leftarrow a_{r_2}^{\text{initial}} \qquad b_{r_2} \leftarrow b_{r_2}^{\text{initial}}$$

It then iteratively updates these values using the following two steps, which are repeated several times. First, it calculates the gradient $\nabla L'$ at the point $(a_{r1}, b_{r1}, a_{r2}, b_{r2})$:

$$\nabla L'(a_{r1}, b_{r1}, a_{r2}, b_{r2}) = \Big(\frac{\partial L'(a_{r1}, b_{r1}, a_{r2}, b_{r2})}{\partial a_{r1}}, \frac{\partial L'(a_{r1}, b_{r1}, a_{r2}, b_{r2})}{\partial b_{r1}}, \frac{\partial L'(a_{r1}, b_{r1}, a_{r2}, b_{r2})}{\partial a_{r2}}, \frac{\partial L'(a_{r1}, b_{r1}, a_{r2}, b_{r2})}{\partial b_{r2}}\Big)$$

Then, each of the parameters is updated by the rules

$$\begin{aligned} a_{r_1} &\leftarrow a_{r_1} + \lambda_a \times \frac{\partial L'(a_{r1}, b_{r1}, a_{r2}, b_{r2})}{\partial a_{r1}} \qquad & b_{r_1} &\leftarrow b_{r_1} + \lambda_b \times \frac{\partial L'(a_{r1}, b_{r1}, a_{r2}, b_{r2})}{\partial b_{r1}} \\ a_{r_2} &\leftarrow a_{r_2} + \lambda_a \times \frac{\partial L'(a_{r1}, b_{r1}, a_{r2}, b_{r2})}{\partial a_{r2}} \qquad & b_{r_2} &\leftarrow b_{r_2} + \lambda_b \times \frac{\partial L'(a_{r1}, b_{r1}, a_{r2}, b_{r2})}{\partial b_{r2}} \end{aligned} \tag{1}$$

where λ_a and λ_b are called the learning rate, and are hyperparameters of the model. In our experiments, we used $\lambda_a = 10^{-8}$ and $\lambda_b = 10^{-6}$. We used no systematic procedure to choose these values, but they were chosen so as to maintain the numerical stability of the model. The numerical stability of the gradient $\nabla L'$ is dependent on the number of terms the loss L' is composed of (see Equation 3 below). If the learning rate is too big, the update steps performed in Equation 1 become bigger and bigger as the iterative process develops, leading the gradients to diverge. In that case, the parameters found are likely not a local minimum of the loss function. Conversely, if the gradient is too small, the update steps of Equation 1 are also too small, it may take too many iterations to arrive at the local minimum. The values chosen for λ_a and λ_b were found to produce (seemingly) stable results but still arrive at a local minimum.

The two step iterative variable update process can be repeated until a certain halting condition is achieved. In our experiments, we always repeated it for exactly 500000 iterations. The whole process took around 2 minutes for each trial[1], running in a machine with a Core i7 CPU with a maximum clock of 2800MHz. The parameters a_{r1}^{initial}, b_{r1}^{initial}, a_{r2}^{initial}, b_{r2}^{initial} were chosen by fitting a single linear regression[2] to the trial data and then using its slope and intercept as a basis for the initialization. Concretely, let a and b be the slope and intercept, respectively, output by the linear regression. Then:

$$\begin{aligned} a_{r1}^{\text{initial}} &= a - 0.05 \\ a_{r2}^{\text{initial}} &= a + 0.05 \\ b_{r1}^{\text{initial}} &= b - 50 \\ b_{r2}^{\text{initial}} &= b + 50 \end{aligned}$$

We now turn to the calculation of the gradient $\nabla L'$ as it was implemented in our code. Notice that the derivative of a sum is the sum of derivatives:

$$\frac{d}{dx}(u + v) = \frac{du}{dx} + \frac{dv}{dx}$$

[1] This was calculated as the average over the first 40 trials in our dataset.

[2] We used R's *lm()* function to fit this model.

This is valid for the value L', which is a sum of several terms, each of which corresponds to one fixation. Therefore, to calculate the gradient $\nabla L'$ it is only necessary to calculate the gradient of each of the terms (corresponding to each of the fixations) and sum them all. Consider, for example, a trial in which the a participant made exactly three fixations f_1, f_2 and f_3 in a given trial. In that case, L' would be given by the sum of the losses associated with each fixation,

$$L'_{example} = \min\left(d^2_{r1,f1}, d^2_{r2,f1}\right) + \min\left(d^2_{r1,f2}, d^2_{r2,f2}\right) + \min\left(d^2_{r1,f3}, d^2_{r2,f3}\right)$$

and therefore its gradient would be given by

$$\nabla L'_{example} = \nabla \min\left(d^2_{r1,f1}, d^2_{r2,f1}\right) + \nabla \min\left(d^2_{r1,f2}, d^2_{r2,f2}\right) + \nabla \min\left(d^2_{r1,f3}, d^2_{r2,f3}\right). \quad (2)$$

where each term (i.e., each $\nabla \min\left(d^2_{r1,f}, d^2_{r2,f}\right)$) is the gradient of that term with respect to the four parameter variables. In other words, let $t_f = \min\left(d^2_{r1,f}, d^2_{r2,f}\right)$ be the term corresponding to the loss associated with the fixation f. Then

$$\nabla t_f = \left(\frac{\partial t_f}{\partial a_{r1}}, \frac{\partial t_f}{\partial b_{r1}}, \frac{\partial t_f}{\partial a_{r2}}, \frac{\partial t_f}{\partial b_{r2}}\right).$$

and

$$\nabla L' = \sum_{\forall f \in F} \nabla t_f, \quad (3)$$

Before proceeding with the value of these derivatives, note that the loss associated with each fixation (see the example in Equation 2) will only depend on the (parameters of the) regression line to which it is closest – i.e., they'll either depend on a_{r1} and b_{r1}, or on a_{r2} and b_{r2}.

In addition, note that the min function does not change when the higher parameter is changed:

$$\min(x, x+k) = \min(x, x+j), \quad \forall k, j \in \mathbb{N}$$

This suggests that the derivatives $\frac{\partial t_f}{\partial a_{r1}}$ and $\frac{\partial t_f}{\partial b_{r1}}$ are going to be zero whenever the derivatives $\frac{\partial t_f}{\partial a_{r2}}$ and $\frac{\partial t_f}{\partial b_{r2}}$ are not zero, and vice-versa.

With this in mind, we can define the derivatives above. If $f = (x, y)$ and

$$t_f = \min\left(d^2_{r1,f}, d^2_{r2,f}\right) = \min\left((a_{r1}x + b_{r1} - y)^2, (a_{r2}x + b_{r2} - y)^2\right), \quad (4)$$

then[3]

$$\frac{\partial t_f}{\partial a_{r1}} = \frac{\partial}{\partial a_{r1}}\left(\min\left((a_{r1}x + b_{r1} - y)^2, (a_{r2}x + b_{r2} - y)^2\right)\right) =$$

[3]The following derivatives were adapted from Wolfram Alpha (`https://www.wolframalpha.com/`). The query used was *derivatives min((a*x1 + y1 - b)^2, (a*x2 + y2 - b)^2)*, where *a* and *b* indicate the fixation coordinates x and y in Equation 4, and *x1*, *y1*, *x2* and *y2* indicate the variables a_{r1}, b_{r1}, a_{r2} and b_{r2}, respectively.

$$\begin{cases} 2x(a_{r1}x + b_{r1} - y) & \text{if } (a_{r1}x + b_{r1} - y)^2 \leq (a_{r2}x + b_{r2} - y)^2 \\ 0 & \text{otherwise} \end{cases}$$

$$\frac{\partial t_f}{\partial b_{r1}} = \frac{\partial}{\partial b_{r1}} \left(\min \left((a_{r1}x + b_{r1} - y)^2, (a_{r2}x + b_{r2} - y)^2 \right) \right) =$$

$$\begin{cases} 2(a_{r1}x + b_{r1} - y) & \text{if } (a_{r1}x + b_{r1} - y)^2 \leq (a_{r2}x + b_{r2} - y)^2 \\ 0 & \text{otherwise} \end{cases}$$

$$\frac{\partial t_f}{\partial a_{r2}} = \frac{\partial}{\partial a_{r2}} \left(\min \left((a_{r1}x + b_{r1} - y)^2, (a_{r2}x + b_{r2} - y)^2 \right) \right) =$$

$$\begin{cases} 2x(a_{r2}x + b_{r2} - y) & \text{if } (a_{r1}x + b_{r1} - y)^2 > (a_{r2}x + b_{r2} - y)^2 \\ 0 & \text{otherwise} \end{cases}$$

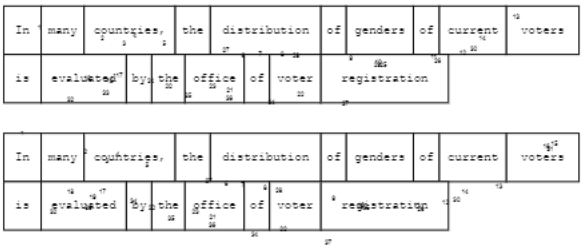
$$\frac{\partial t_f}{\partial b_{r2}} = \frac{\partial}{\partial b_{r2}} \left(\min \left((a_{r1}x + b_{r1} - y)^2, (a_{r2}x + b_{r2} - y)^2 \right) \right) =$$

$$\begin{cases} 2(a_{r2}x + b_{r2} - y) & \text{if } (a_{r1}x + b_{r1} - y)^2 > (a_{r2}x + b_{r2} - y)^2 \\ 0 & \text{otherwise} \end{cases}$$

In our actual implementation, the conditions above (e.g., $(a_{r1}x + b_{r1} - y)^2 > (a_{r2}x + b_{r2} - y)^2$), which depend solely on the distance of a fixation to the regression lines, and instead used the sets R_1 and R_2, which allow for a fixation to be assigned to the farther regression line in some special cases.

When the correction procedure fails

The correction procedure described above, while effective in a majority of noisy trials, may fail whenever the calibration error is not reducible to two straight lines. In these cases, a more complicated curve may be needed, and fitting this curve may be difficult given the relatively few fixation points produced in each trial. Figure 4 shows an example of a particularly bad trial that could not be corrected by our algorithm. Indeed, the corrected result was actually less usable than the original one, and goes in the opposite direction of an ideal correction. Trials like these were discarded from the data prior to analysis.



(a)

(b)

Figure 4. An example of a particularly bad trial before correction (a) and after correction (b), showing how corrrection may fail.