

Software for the automatic correction of recorded eye fixation locations in reading experiments

Andrew L. Cohen

Published online: 13 December 2012
© Psychonomic Society, Inc. 2012

Abstract Because the recorded location of an eyetracking fixation is not a perfect measure of the actual fixated location, the recorded fixation locations must be adjusted before analysis. Fixations are typically corrected manually. Making such changes, however, is time-consuming and necessarily involves a subjective component. The goal of this article is to introduce software to automate parts of the correction process. The initial focus is on the correction of vertical locations and the removal of outliers and ambiguous fixations in reading experiments. The basic idea behind the algorithm is to use linear regression to assign each fixation to a text line and to identify outliers. The freely available software is implemented as a function, `fix_align.R`, written in R.

Keywords Eye tracking · Automatic correction · Reading · Outlier detection · Fixation locations · Regression

The recorded location of an eyetracking fixation is not a perfect measure of the actual fixated location. Even well-calibrated systems contain some error. For the EyeLink eyetracking system (SR Research, Toronto, Ontario, Canada), for example, a calibration is labeled as “good” if the worst error, as compared to a validation stage, is less than 1.5 deg of visual angle and the average error is less than 1 deg (Shen, 2011; reading researchers, though, tend to use more stringent criteria—e.g., a max error up to 1 deg and an average error up to 0.5 deg for a full-screen calibration). Due to within- and between-trial drift, calibration accuracy can also decline over the course of an

experiment. On the current EyeLink system, the default setting does not correct for drift between trials, but only checks the level of drift and doesn’t trigger the trial if the eye appears to be outside the target region for checking gaze.

Such measurement error may not be uniform. For example, the consistent experience of eyetracking researchers is that vertical error is typically much worse than horizontal error. This asymmetry is fortunate for reading experiments, in which vertical error is, in principle, correctable. That is, in a typical reading experiment with only a few lines of spaced text, it should be possible to determine which line is being fixated. Horizontal error, however, is not easily corrected; letters are adjacent, and there is not likely to be any simple way of determining what letter is the intended fixation target.

Because of such measurement issues, recorded fixation locations (typically, the vertical fixation location) must be corrected before analysis. Corrections often include the removal of spatial outliers (i.e., fixations that appear to lie far from any line of text) and adjusting of fixation locations to better represent the fixated locations. Fixations are typically corrected manually, sometimes within a program such as EyeDoctor (www.psych.umass.edu/eyelab/software/). Such correction, however, is time-consuming and necessarily involves a subjective component. For example, in a multiline reading experiment, it is sometimes unclear to which line of text a fixation belongs. Even in single-line reading experiments, subjective judgments can come into play; for example, it is sometimes unclear whether a fixation should be considered to be located on the text or elsewhere on the screen. The goal of this article is to introduce software to automate parts of the correction process. The initial focus of the software is on the correction of vertical locations and the removal of outliers and ambiguous fixations in reading experiments.

A. L. Cohen (✉)
Department of Psychology, University of Massachusetts,
135 Hicks Way,
Amherst, MA 01003-7710, USA
e-mail: acohen@psych.umass.edu

This software is implemented as a function written in R (R Development Core Team, 2011),¹ a popular and freely available language and environment for statistical computing. The software modifies the output from the EyeLink eyetracking system (SR Research, Toronto, Ontario, Canada), a system that is both state-of-the-art, in terms of temporal resolution and spatial accuracy (depending on the system configuration, up to 2,000 Hz and 0.25 deg of visual angle), and widely used in reading experiments (see http://www.sr-research.com/pub_reading.html for examples). In particular, the software takes .asc files as input. This file format is the ASCII-converted output from the EyeLink system. The .asc file includes information about the eyetracking system and experiment, as well as fixation locations and durations, pupil size, saccade duration, and blink information.

The basic idea behind the algorithm is to use linear regression to assign each fixation to a text line. There is one regression line for each text line. The regression lines are parameterized by a slope, vertical offset, and standard deviation. The slopes of the regression lines can vary from horizontal, and the regression line initially passes through the start of the text line. The vertical offsets allow for an overall bias for the measured fixations to be above or below the text lines, by moving the start of the regression lines above or below the start of the text lines. The standard deviations—that is, the standard deviations of the normal distributions centered on the regression lines—are assumed to be constant along the regression lines. The slope, vertical offset, and standard deviation of each regression line is found that maximizes the likelihood of the measured, uncorrected fixations. Because it is assumed that calibration error is relatively consistent across the display, each regression line shares a common slope, vertical offset, and standard deviation. The vertical difference between the regression lines is assumed to be the vertical difference between the text lines.

Each fixation is assigned to the text line associated with the highest-likelihood regression line. The y location of the fixation is then set to the y location of that text line, but the x -location is not changed. Outliers are fixations that are too far from any regression line, in standard deviation units. If the highest- and second-highest regression line likelihoods are too similar, the fixation is marked as being ambiguous. An option available in the software allows ambiguous points bounded by points that are unambiguously classified into the same text line to be reclassified into that text line (the “run rule”).

The function is named `fix_align.R` and can be found on the web page of the author’s lab (<http://www.psych.umass.edu/PACLab/resources/>). The function definition is provided in Fig. 1. The full set of arguments is described in the Appendix, but the critical arguments are listed here: `start_pts` is an $n \times 2$ matrix that defines the x and y locations of the start of each text line, where n is the number of

```
fix_align <- function(start_pts,
                      asc_files='.',
                      xy_bounds=NULL,
                      keep_y_var=FALSE,
                      use_run_rule=TRUE,
                      trial_plots=TRUE,
                      save_trial_plots=FALSE,
                      summary_file=TRUE,
                      show_image=FALSE,
                      fa_dir='FA_Dir',
                      start_flag='TRIALID',
                      den_sd_cutoff=Inf,
                      den_ratio_cutoff=1,
                      k_bounds=c(-.1, .1),
                      o_bounds=c(-50, 50),
                      s_bounds=c(1, 20))
```

Fig. 1 Function definition of `fix_align.R`

text lines. `asc_files` is a vector of .asc file names and/or folders containing .asc files. `xy_bounds` provides a way to remove points a priori from the analysis. Any points outside the rectangles provided by `xy_bounds` are considered out of bounds and are removed from the analysis before the regression lines are determined. If `use_run_rule` is set as TRUE, ambiguous fixations are potentially reclassified by their surrounding fixations as described above; otherwise, they remain ambiguous. If `trial_plots` is set as TRUE, trial-by-trial plots of the original and reformatted fixations are shown. `start_flag` is the text that marks the start of data collection in the .asc file, which will typically be either “TRIALID” or “SYNCTIME.” Fixations that end before the start flag will be labeled “nit” (for *not in trial*). Fixations that start before the start flag but that end after the start flag will be labeled as “part” (for *partial*). Neither not-in-trial nor partial fixations are included in analysis, but in order to visualize where the participant was looking immediately prior to data collection, these fixation do show up in the trial plots as points along the fixation path (dotted line) without markers. `den_sd_cutoff` defines the number of standard deviations from a regression line that determines an outlier. `den_ratio_cutoff` defines the ratio (in density) that determines ambiguous fixations.

Depending on the settings, the function can create three outputs. First, the trial-by-trial visualization of the fixation adjustment and classifications (described below) can be saved as .tiff files. Second, a .fas (*fix align summary*) file can be created that saves the slope, vertical offset, standard deviation, regression fit, and the numbers of total, kept, out-of-bounds, ambiguous, outlier, not-in-trial, and partial fixations for each participant and trial. The final, and most important, option is the reformatted .asc file. This file is a copy of the original .asc file, with the y locations of the fixations changed to reflect the regression analysis and fixation removal process. This file has the same name as the original .asc file, with “_fa” (*fix align*) added to the end.

¹ Created and tested in R version 2.14.1.

```
fix_align <- function(start_pts=rbind(c(27, 436), c(27, 556)),
  asc_files='Data/23_50H_B',
  save_trial_plots=TRUE,
  start_flag='SYNCTIME',
  den_sd_cutoff=4.5,
  den_ratio_cutoff=5500)
```

Fig. 2 Call to `fix_align.R` used to produce Fig. 3 and the associated output

Only the y location field of the EFIX (*end of fixation*) lines of the .asc file are altered. If the fixation is not removed, the new y location is determined from the text line classification of the regression analysis, as well as the y location for that text line as given in `start_pts`. If the fixation is

removed, the y location is as follows: A fixation that ends before the start of data collection (not in trial) is labeled “-1000”; a fixation that starts before but ends after the start of data collection (partial) is labeled “-1001”; a fixation that is out of bounds is labeled “-1002”; an ambiguous fixation is labeled “-1003”; and a fixation that is too far from any of the regression lines—that is, an outlier—is labeled “-1004.” All other lines and fields of the .asc file remain unchanged.

To test `fix_align.R`, it was used to analyze a multi-line reading experiment that produced particularly difficult-to-classify fixations (Pazzaglia, Staub, & Rotello, 2012). On each trial of the experiment, 20 words, ten on each text line,

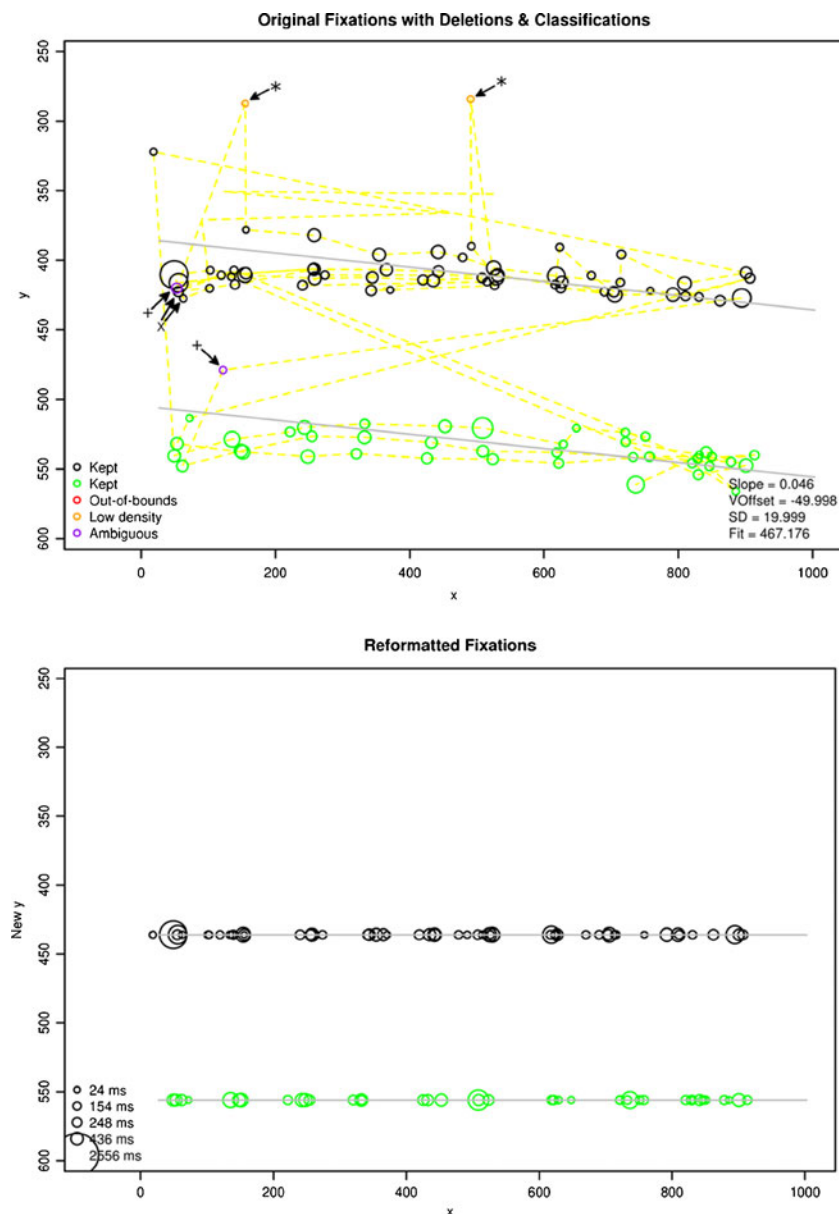


Fig. 3 A single trial plot from a call to `fix_align.R`. In the top panel, a “*” and a “+” indicate that a point was removed due to, respectively, the density or the ambiguity rule. An “x” means that a

point was ambiguous but was kept due to the run rule. The *arrows* and associated symbols were not in the `fix_align.R` plot. The bottom panel shows the y -corrected fixations

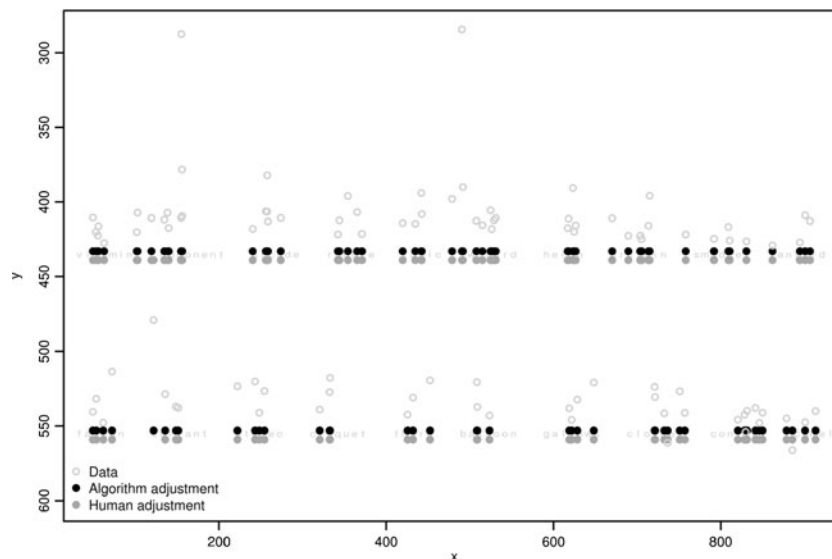


Fig. 4 A comparison of the fixations from Fig. 3, reformatted either by `fix_align.R` or by an experienced eyetracking researcher

were shown to the participant, who was asked to memorize the words for a later memory test. These data presented a stringent test for `fix_align.R` because they included many more fixations per trial than are present in a traditional reading experiment (in some cases, hundreds of fixations) and because the participants' scan patterns were highly irregular, rather than proceeding from left to right and top to bottom. A total of five participants were presented seven trials apiece; the first and last trials were not analyzed, leaving five trials analyzed per participant. The function call to `fix_align.R` is shown in Fig. 2, and the trial plot for one trial from one participant is shown in Fig. 3. The top panel of Fig. 3 shows the original fixations, with deletions and classifications. Each circle is a fixation. The diameter of the fixation is linearly related to the fixation duration (the scale is given in the bottom panel). The measured fixations are clearly not horizontal and are above the text lines (436 and 556 on the y -axis; see also the bottom panel). The gray lines are the fitted regression lines. The parameters of the regression lines are given in the top panel. Because the y -axis is reversed (i.e., it is in screen coordinates), the slopes of the lines are positive, even though they angle down. The vertical offset appropriately reflects the vertical difference between the measured fixations and text lines, and the fit is the negative log likelihood of the regression lines. No fixations were out of bounds, so all fixations were involved in the determination of the regression parameters. The points marked with a "*" were more than 4.5 standard deviations from the nearest regression line. The points marked with a "+" were ambiguous; that is, the ratio of the higher and lower regression line likelihoods was less than 5,500.² Note

that `use_run_rule` was set as TRUE in the function call. Although they were initially labeled ambiguous, the points marked with an "x" were both bounded by fixations from the upper text line, and so were classified within that text line. The dashed line indicates the order of fixations. The bottom panel of Fig. 3 shows the fixations after their y locations were adjusted. The gray lines are the locations of the text lines defined by `start_pts`.

The output of `fix_align.R` was compared to the adjustments made by an experienced eyetracking researcher using EyeDoctor. Because the software was not designed to remove fixations based on blinks, overly long saccades, and so forth, the researcher was asked to adjust all fixations (i.e., not to remove any, but simply to adjust all fixations vertically). Likewise, the arguments of `fix_align.R` were set so that no points were removed (`den_sd_cutoff = Inf`, `den_ratio_cutoff = 0`). Note that these arguments are somewhat different from those used to create Fig. 3. A comparison of the software and human adjustments for the same trial shown in Fig. 3 can be seen in Fig. 4,³ where the word stimuli are shown in their approximate locations. The open circles show the original fixation locations. It is clear that the measured fixation locations do not fall on the text lines. For ease of comparison, the human- and software-adjusted fixations have been moved slightly vertically. For this trial, the correspondence between the human- and software-adjusted points is nearly perfect. The difference is a single point, the lower "+" fixation in Fig. 3: The human classified this point with the upper text line, whereas the algorithm classified it with the lower text line. Although it is physically closer to the

² This is a very high cutoff and was used to produce ambiguous points for this demonstration.

³ Because it is to the left of the first text character, the leftmost fixation in Fig. 3 was removed during EyeDoctoring and is not shown in Fig. 4.

upper text line, this point is 5,163 times as likely to belong to the lower regression line.⁴

Across all participants and trials, the correspondence between the human and software adjustments is very high. The two agreed on 99.78 % (1,803) of the 1,807 total fixations. Of the four discrepancies, three (including the one above) were ambiguous points that were more likely to come from the more distant text line. These differences could be removed by appropriately setting `den_ratio_cutoff`. The final difference was due to an extreme outlier that pulled the regression line far from the text lines. This difference could have been avoided by setting `xy_bounds`.

Fixation locations are never perfectly calibrated and usually need some adjustment. This adjustment is usually done manually and involves some subjective judgments. `fix_align.R` is a freely available function, running in R, that uses linear regression to automate the classification and adjustment processes. The comparison here to human adjustments demonstrates that this function does an excellent job classifying and adjusting eyetracking fixations.

Author note Thank you to Adrian Staub, for his guidance designing the software and for help with the manuscript; to Angela Pazzaglia, for kindly providing her data to test the software; and to Chuck Clifton and Adina Galili, for EyeDoctoring the data for comparison.

Appendix

`xy_bounds` provides a way to remove points from analysis a priori. Any points outside the rectangles provided by `xy_bounds` are considered out of bounds and are removed from analysis before the regression lines are determined. All other points are included in the creation of the regression lines. The bounds can be set to no bounds (NULL), a single rectangle for the entire experiment (a 1×4 vector: x -min, x -max, y -min, and y -max), or one rectangle per trial (an $n \times 4$ matrix of rectangles). If `keep_y_var` is set as TRUE, the residual of

⁴ Although the relative likelihood is less extreme, this point was still classified with the lower text line if all points with y values less than 350 were removed before the regression analysis.

each fixation around its assigned regression line is kept when the y location is changed; otherwise, it is removed. If `save_trial_plots` is set as TRUE, the trial plots are saved as .tiff files, in a subdirectory called “Trial_Plots” in the directory defined by `fa_dir`; in this case, the plots will only be saved—that is, they will not be shown on the screen. If `summary_file` is set as TRUE, a .fas (*fix align summary*) file is created that saves the slope, vertical offset, standard deviation, regression fit, and the numbers of total, kept, out-of-bounds, ambiguous, outlier, and other fixations for each participant and each trial. The summary file is saved in the directory defined by `fa_dir` and is named on the basis of the current date and time. If `show_image` is set as TRUE, the trial plots are shown on top of background images. These background images are .tiff files located in the same directory as the data files. One image can be used per .asc file, or one image per trial per .asc file, and images must be saved as .tiff files with the same name as the .asc file and the .tiff ending. If one image is used per trial, the end of the name must have an underscore followed by a trial number (e.g., “_1” for Trial 1). `fa_dir` is the name of the directory used to store the reformatted .asc files, .fas files, and trial plots. The .asc files are marked by having “_fa” (*fix align*) at the ends of their names. `k_bounds`, `o_bounds`, and `s_bounds` are 1×2 vectors that define the minimum and maximum values that the regression line slope, vertical offset, and standard deviation, respectively, can take. If set to NULL, `k_bounds` and `o_bounds` have values of 0.

References

- Pazzaglia, A. M., Staub, A., & Rotello, C. M. (2012). *Encoding time and the mirror effect in recognition memory: Evidence from eye-tracking*. Manuscript in preparation.
- R Development Core Team. (2011). *R: A language and environment for statistical computing [Software]*. Vienna, Austria: R Foundation for Statistical Computing. Available at <http://www.R-project.org>
- Shen, J. (2011, July 28). *Re: Limits of calibration*. Message posted to <https://www.sr-support.com/forums/showthread.php?t=2340>