

RESEARCH PAPER

Open Access



Vertical error correction of eye trackers in nonrestrictive reading condition

Charles Lima Sanches^{*}, Olivier Augereau and Koichi Kise

Abstract

The eye tracking technology is used for four decades for studying reading behavior. The applications are various: estimating the reader comprehension, identifying the reader, summarizing a read document, creating a reading-life log, etc. The gaze data used in such applications has to be accurate enough to perform the analysis. In order to improve the accuracy, most of the experiments are set up with restrictive conditions such as using a head fixation and a professional eye tracker. It implies that the results are valid only in restrictive laboratory settings and an unrealistic small error is produced by the experiment. However, the use of affordable eye trackers in realistic conditions of reading leads to large errors in the recordings. We propose a new algorithm to correct the vertical error and to align the gazes with the text. The proposed algorithm is robust to rereading and skipping some parts of text, contrary to all the other algorithms of the state of the art. We show that up to 69 % of the gazes are aligned with the correct text lines.

Keywords: Eye tracker, Gaze analysis, Reading understanding, Sequence alignment

1 Introduction

Since the first studies conducted by Rayner [11] around 40 years ago, the analysis of the reading behavior with the use of eye tracking systems has been widely popularized and especially in the last 10 years, thanks to the development of affordable eye trackers. Rayner has shown that the movements of the eyes while reading can be divided in two main categories: fixations and saccades. The fixations correspond to the short stops on words during reading which last about 250 ms, and the saccades correspond to the quick movements of the eyes between two fixations. By recording the sequence of the fixation positions, the reading behavior can be analyzed and different services provided to the user.

For instance, the eye gaze can be used for creating a “reading-life log” (Augereau et al. [1]). The idea is to record the read words in daily life with an eye tracker and to save them into a log file. Then, the user can research information on his reading history, count how many words he reads per day, analyze what kind of text he reads, etc. The eye gaze position can also be used for creating a summary of a text based on the reading behavior

[14] by measuring the reader attention to every word in the document. The attention on a word is measured as the number of gazes on this word. Some other applications are also possible, such as detecting the understanding of a text (Kunze et al. [8]) or providing a real-time translation of a word (Hyrskykari et al. [7]).

All these applications depend highly on the analysis of the eye gaze position. This analysis is hard to perform because of the inaccuracy of eye trackers [3]. The inaccuracy can be caused by miscalibration, head movements, lighting change, etc. Figure 1 illustrates the eye tracker inaccuracy. On the different recordings of the figure, we can notice a difference between the position of the point we stare and the position recorded by the eye tracker. By comparing the position of the fixation lines and the text lines, we can easily observe a vertical error on the recordings. This error is hard to estimate automatically since it varies from one recording to another, depending on the eye trackers and the experimental conditions. Furthermore, the exact error cannot be modeled as a translation or an affine transformation. However, the most representative eye tracking systematic error according to the literature is the vertical error [6, 13].

There are two main solutions to compensate the error: (1) using specific recording conditions and (2) processing

^{*}Correspondence: charles-ls@hotmail.fr
Department of Computer Science and Intelligent System, Osaka Prefecture University, Osaka 599-8531, Japan



the signal. In order to limit the inaccuracy of the eye tracking systems, the researchers controlled the conditions of the experiments by using head fixations, bite bars, etc. However, fixing such strict conditions of experiments limits the usage of eye trackers in a laboratory environment. Because our aim is to analyze the reading behaviors in natural and realistic conditions of reading, we choose to focus on finding a post-processing algorithm to correct this problem. In this paper, we propose an algorithm to estimate and correct automatically the vertical error of the eye tracker.

Unlike the other methods from the state of the art, our method can be used in a "nonrestrictive" reading situation, i.e., even if some parts of the text are skipped or reread. For the experiments we use the Tobii EyeX eye tracker which is available for 139\$¹. It implies that the error we deal with is quite larger than the experiments using a chin rest or a professional eye tracker. But working with an inexpensive eye tracker is important in order to disseminate the reading analysis algorithms to a large community.

The rest of the paper is organized as follows. First, we present some related work. In the following section, we explain the different steps of the proposed algorithm. The fixations are processed from the eye gazes and segmented

as a set of "fixation lines" where a fixation line corresponds to the reading of one text line. Then, the vertical error is coarsely estimated and the fixation lines are matched with the text lines. In the next section, we present the algorithm proposed by Yamaya et al. [15] for correcting the vertical error and compare it to our method. After this, we show through the experiments that we can align 69 % of the eye gaze lines with their corresponding text lines and compare the results with the state-of-the-art method. Finally, we conclude and discuss about the outlooks.

2 Related work

Formerly, the researchers dealt with the inaccuracy of the eye tracker by using a chin rest [9], a high-technology eye tracker [15], a large line spacing text [4], or a large-size font text [10]. Only few recent studies have been dedicated to the correction of the vertical error of the eye tracker by a post-processing of the recordings.

Hyrskykari [6] proposed a method to sequentially map the fixations to the text lines. In this method, the first fixation is mapped to the closest word, and then, the following fixations are aligned with the same text line. Martinez-Gomez et al. [9] proposed to correct the error by using a global text-gaze alignment. In this method, the gazes are represented as a scatter plot and the words are

represented as boxes. The aim of the algorithm is to find the global transformation which best aligns the gazes with the boxes. The most recent research work is proposed by Yamaya et al. [15]. Their idea is to perform a global alignment between the fixation lines and the text lines. The constraint of reading each line one by one, respecting the order of the text (from the first to the last line), is used to perform the alignment.

To sum up, the related work contains one or more of the following major restrictions:

- Rereading, skimming, or skipping some parts of the text strongly affects the algorithms.
- The vertical error cannot be greater than a line spacing.
- A chin rest or a professional eye tracker is used.

To assess the performance of the proposed method, we implemented the algorithm of Yamaya et al. [15] which is the most recent method to correct the vertical error of an eye tracker. The details of this algorithm and the differences with the proposed method will be described in Section 4.

In the next section, we present our algorithm.

3 Vertical error correction

The first steps of the algorithm is to compute the fixations from the gaze data and to group the fixations by fixation lines. Then, the objective is to find for each fixation line the corresponding text line. Let $S = \{s_1, s_2, \dots, s_N\}$ be the set of fixation lines and $T = \{t_1, t_2, \dots, t_M\}$ be the set of text lines. For each fixation line $s_i \in S$, we can find a corresponding text line $t_j \in T$. In other words, we consider :

$$\forall s_i \in S \exists g_i \mid g_i(s_i) = t_j \tag{1}$$

Our problem is formulated as finding the optimal function g_i which associates the corresponding text line of each fixation line. In order to be robust to rereading or skipping some part of the text, the proposed algorithm aligns individually each fixation line with the corresponding text line.

The algorithm can be summed up in three main steps:

1. Creating the fixation lines
2. Estimating and correcting coarsely the vertical error
3. Performing a precise (fixation line; text line) matching

In more details, the algorithm is the following. First, we extract the fixations and segment them into fixation lines. For each fixation line, we compute a matching score with the n_1 nearest lines. By using this score, some matchings are selected and used for estimating coarsely the global vertical error. All fixations are then translated. The final step consists in matching each translated fixation line with the best candidate among the n_2 nearest text lines

to obtain the final vertical positions of the fixation. We can notice that, after correcting coarsely the vertical error, the fixation lines are closer to the text lines so we can choose n_2 such that $n_2 < n_1$, in order to improve the final matching step.

In the following, we use a system of coordinates (x, y) such as Fig. 2. In this space, we define the x -coordinate of a fixation (x_f), the x -coordinate of a word (x_t), the y -coordinate of a fixation line (Y_F), and the y -coordinate of a text line (Y_T). The set of the x -positions of words in a text line is called L_t , and the set of x -positions of fixations in a fixation line is called L_f .

3.1 Creating the fixation lines

The fixations are obtained from the raw gazes by using the method presented by Biedert et al. [2]. Then, we segment the fixation sequence into a set of fixation lines. We detect the large regressions which occur when the reader switches from one line to another one. Considering $x_f(i)$ as the x -coordinate of the fixation i , a line break is detected if:

$$x_f(i + 1) - x_f(i) < -P < 0, \tag{2}$$

where P is a positive integer, chosen large enough not to detect short rereading as reading a new line.

3.2 Estimating and correcting coarsely the vertical error

The aim of this step is to match a fixation line with a text line and to give a score to this matching. Because we do not know which text line corresponds to which fixation line, for a given fixation line, we select the n_1 nearest text lines according to the vertical axis and compute the matching scores. The rating is computed by using the dynamic time warping (DTW) algorithm which is a sequence alignment algorithm. The input is a pair of

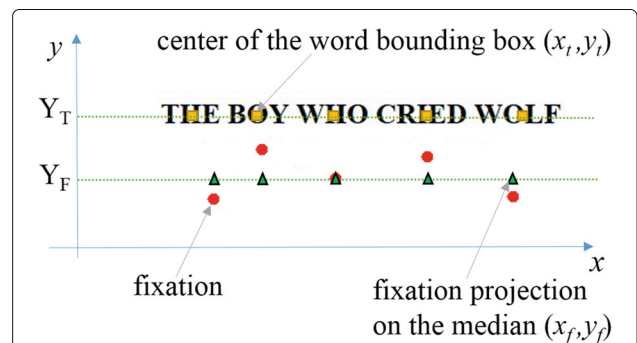


Fig. 2 Definition of the system of coordinates. x_f is the x -coordinate of a fixation. x_t is the x -coordinate of a word. It is based on the bounding box center of a word. Y_T is the y -coordinate of a text line. It is based on the bounding box center of a text line. Y_F is the y -coordinate of a fixation line. It is defined as the median value of the y -coordinate of each fixation in this line

sequences $(L_t; L_f)$ we want to match. The DTW algorithm provides n_1 scores for each matching $(L_f(i); L_t(i))$. The matching is based on the position of the words in a text line. If two text lines have different lengths or if the words are distributed at different positions, the matching can perform well. So the larger n_1 , the more confusion between the matchings there can be. Unfortunately, because the error of the eye tracker is large, we cannot choose a small value for n_1 before correcting the vertical translation.

To estimate the global vertical translation, we want to keep only the pairs $(L_f(i); L_t(i))$ which are correctly matched. So we look for the non-ambiguous matchings. For a fixation line, the difference between the best matching score s_1 and the second best matching score s_2 is computed: $D = s_1 - s_2$. If this difference is large enough ($D > T$), we consider that the matching is correct. T is a threshold which defines the strictness of the selected matchings. If T is set to a low value, we will keep almost all the matchings of the previous step, and among these matchings there will be some fixation lines matched with wrong text lines. If T is set to a high value, we will select few matchings but less ambiguous.

After selecting the matchings, we estimate the vertical error. For each pair $(L_f(i); L_t(i))$, we compute the distance in pixels between the y -coordinate of the fixation line (Y_F) and the y -coordinate of the corresponding text line (Y_T). For each pair, we obtain a translation G , where $G = Y_F - Y_T$. By considering all the possible vertical translations, we then compute G_a as the average value of all the vertical translations. We apply the vertical translation G_a to all the fixations.

3.3 Performing a precise (fixation line; text line) matching

After correcting the vertical error, the fixation lines are closer to the corresponding text lines. For each fixation line, we align vertically all fixations with the best text line by using exactly the same method as in the previous step. The only difference is that a smaller number of n_2 nearest text lines for the matching is selected. In this step, the matching score is not used to estimate the vertical error; we simply align the fixations with the matched text lines.

4 Comparison with the state of the art

In this section, we will present Yamaya et al.'s [15] method for correcting the vertical error of an eye tracker while reading. We will point out the differences with our method.

4.1 Similarity measure and global alignment

In this algorithm, several steps are similar to the proposed algorithm such as the line break detection and a line matching step using a sequence alignment algorithm. The sequence algorithm used in this technique is

the Needleman-Wunsch algorithm which uses a similarity measure to align the line of fixations with the line of text. Let $S = \{s_1, s_2, \dots, s_N\}$ be the sequence of fixation lines and $T = \{t_1, t_2, \dots, t_M\}$ be the sequence of text lines. If we call $M(s_k, t_l)$ the similarity measure between the fixation line s_k and the text line t_l , then $M(s_k, t_l)$ is a function of two parameters:

- The difference of length between s_k and t_l
- The distance between s_k and t_l

The input of the sequence alignment algorithm is the whole sequence S and the whole sequence T . The algorithm tries then to find the best alignment between S and T by using the similarity measure.

4.2 Differences with the proposed method

In the proposed algorithm, the input of the sequence alignment algorithm is a pair (text line; fixation line) we want to match. The main feature used is the distance between the fixation and the word. The process is repeated for each fixation line, so the lines are processed independently and the reading order does not matter. At each step, one fixation line is compared with several text lines: the alignment is local. In other words, with the same notation, suppose that for each fixation line s_i we can find the corresponding text line t_j . The problem of our algorithm could be then formulated as $\forall s_i \in S$ finding the local function g_i so that $g_i(s_i) = t_j$.

In the method proposed by Yamaya et al. [15], the input of the algorithm is all the fixation lines and all the text lines. The algorithm is run just once, and the sequence of the fixation lines has to correspond with the sequence of text lines: the reading order is essential. In this method, the main feature is the difference of length between the lines. All the fixation lines are compared with all the text lines at the same time: the alignment is global. In other words, with the same notation, suppose that for each fixation line s_i we can find the corresponding text line t_j . The problem of this algorithm could be then formulated as finding the global function g such as:

$$\forall s_i \in S, g(s_i) = t_j \quad (3)$$

The method proposed by Yamaya et al. [15] takes into account the sequence order of the fixation lines. As a consequence, in case of specific reading behaviors where the reading order is preserved, the algorithm will have good performances. However, in case of normal reading behavior, the reading order of the text lines cannot be predicted. The reader does not necessarily start to read the text from the first line and reads each line one by one until the last text line. He can start, stop, skip, or reread any line of text. Such natural behavior will greatly impact the performances of Yamaya's algorithm. On the

contrary, because the fixation lines are processed independently in the proposed method, the performances will not be affected by such behaviors. In the next section, we present our experiment and our results to illustrate this remark.

5 Experimental results

In the experiments, seven subjects were asked to read eight different texts. The eye tracker employed in this experiment is a stationary eye tracker Tobii EyeX Controller.

The experiments are divided into four parts, corresponding to four different reading behaviors:

- E1: read forward the text without rereading nor skipping any part of the text.
- E2: read only the second paragraph of a text.
- E3: read the two paragraphs of the text and reread the first one.
- E4: read the second paragraph *before* the first one.

Subjects were asked to read two texts for each experiment. All the texts of all the experiments are different. A sample of the experiment E1 is shown in Fig. 1, and a sample of experiment E2 is shown in Fig. 3. The whole data set is made of 655 fixation lines to be matched with the corresponding text lines. In the line break detection step, P is fixed to 150 pixels which roughly corresponds to half the length of a text line. In the vertical error estimation

step, we set the confidence threshold to $T = 70$ for all the recordings.

5.1 Ground truth

In case of a normal reading situation, some parts of the texts are skipped or reread. However, in order to create the ground truth, we asked the subjects to reread and skip only the part of text planned by the experiment.

5.2 Experiment results

The results are presented in two sections. First, we present the results associated to cases E1 and E2. Second, we present the results of cases E3 and E4. We compare the accuracy of our algorithm with Yamaya et al.'s with all these reading behaviors. The accuracy is computed as a percentage: the number of fixation lines matched with the correct corresponding text lines. The percentages are based on an average of all seven readers and two texts for each experiment.

5.2.1 Preserved reading order: E1 and E2

In E1 and E2, the order of reading is preserved: the texts have been read from one point to the end. Besides, in both cases, there is no rereading. The results are shown in Table 1. We can see that for both experiments, the performance of our algorithm is better or similar to the method proposed by Yamaya et al. In particular, in E1, the difference between the performances of both algorithms is high. This can be explained by the quality of the recordings. The

Can chocolate keep you younger looking? It is too good to be true. Scientists have made an anti-ageing chocolate. It keeps away corrugations and helps the skin look younger. It is called 'Esthechoc,' from the words 'esthetic' (meaning all things beautiful) and 'chocolate'. The makers will start selling it from next month. No one knows the price yet. It will probably be expensive. The lab hopes young, rich business people will buy it.

The chocolate contains a special chemical that gives flamingos their pink color. Scientists say the chemical can make the skin of a 50 to 60-year-old person look like a 20 to 30-year-old. They said people could see changes in their skin in just three weeks. The laboratory said it took over ten years of research, using more than 3,000 volunteers to make Esthechoc. The chocolate will only be sold in boxes of 21 bars. This is one a day for three weeks. Other scientists say more scrutinies are needed for people to know whether or not Esthechoc works.

Fig. 3 Example of text and recording. This is the experiment E2; the subjects were asked to skip the first paragraph of the text

Table 1 Percentages of good matching with both algorithms in case of reading: without rereading nor skipping (E1) and with skipping one paragraph (E2)

	E1	E2
Total number of lines	168	77
Yamaya's method	60 %	83 %
Proposed method	69 %	81 %

use of an inexpensive eye tracker such as Tobii EyeX can lead to errors which highly affect the length of the fixation lines, which is the feature used by the Yamaya et al. algorithm. So, the Yamaya's algorithm uses the sequence of reading to match the lines and fails if the quality of the recording is low. On the other hand, the proposed method does not use the reading order and is still efficient with a low-quality recording. In the next section we will see the behavior of both algorithms in case of a non-preserved reading order.

5.2.2 Rereading and different reading order: E3 and E4

In this experiment, each text is composed of two paragraphs of the same length. The results of this experiments are shown in Table 2. Our algorithm is capable of matching up to 72 % of fixation lines with the corresponding text lines. For E4, the performances of Yamaya's method dramatically drop while the method proposed in this paper has results around 60 %.

These differences are mainly explained because the algorithm proposed by Yamaya et al. uses the reading order to match the fixation lines with the text lines. In case of E4, the reading order is not preserved because readers were asked to read paragraph 2 first. In case of rereading (E3), the algorithm proposed in this paper also has better performances than the Yamaya's method. However, Yamaya's method still manages to align around 60 % of fixation lines with the corresponding text lines. This is because the algorithm successfully matches paragraphs 1 and 2 but fails to match most of the fixation lines corresponding to the rereading. On the contrary, our algorithm can perform well in case of rereading and/or skipping behaviors because the fixation lines are matched with each text line independently. As a consequence, with our algorithm, the reader could read the lines in any order.

Table 2 Percentages of good matching with both algorithms in case of rereading (E3) and reading in a different order (E4)

	E3	E4
Total number of lines	207	203
Yamaya's method	64 %	17 %
Proposed method	72 %	61 %

5.3 General discussion

With these experiments, we can see that our algorithm is capable of matching more fixation lines with the corresponding text lines than the state-of-the-art algorithm. Because we compute the matching scores for the fixation lines independently to each other, the order of reading is not important. A summary of the results is shown in Table 3. In this table, the results are gathered in three sections: the reading order is preserved, the reading order is not preserved and all recordings together. The performance of the state of the art is strongly affected if the reading order is not respected and if the reader reread some parts of the text. Also, if the quality of the recording is too low and the number of lines is high (such as in experiment E1), it highly affects the global performances even if the reading order is preserved. On the contrary, the algorithm proposed in this paper does not have such restriction and, as a consequence, obtains better performances.

However, our algorithm is still not capable of matching all the fixation lines with the corresponding text lines. The remaining error can be explained because the position and number of the fixations do not correspond exactly to the distribution of words in the text. In particular if the reader skips many words unintentionally (e.g., small functional words such as "the" or "of"), it will affect the performance of the algorithms.

In the algorithm, we do not use the reading order: the reader could read the lines in any order. However, during a natural reading behavior, we do not read the lines randomly. So, we could find a compromise between setting up a constrain on the reading order (such as Yamaya's method) or without any constrain (such as in the proposed method) in order to improve the performances of the algorithm. In order to do so, we need a probabilistic model of the reading order in case of a natural reading behavior.

6 Conclusion

We have presented a method for correcting the vertical error of an eye tracker in case of natural reading conditions. Our method can be used to enhance the input of the reading analysis algorithms based on the eye gaze

Table 3 Percentages of good matching with both algorithms in case of reading with preserved order (E1, E2) or not (E3, E4) and global performances with all the recordings (all texts)

	Reading order is preserved	Reading order is not preserved	Total
Total number of lines	245	410	655
Yamaya's method	67 %	41 %	51 %
Proposed method	73 %	66 %	69 %

positions. In our experiment, we have shown that our algorithm is capable of matching in average 69 % of the fixation lines with text lines. Contrarily to the state-of-the-art methods, our method is robust to normal reading behaviors such as skipping, rereading or reading the text lines in a different order. Besides, in our experiment we used an inexpensive eye tracking system which led to a more important error.

However, our algorithm is still not capable of matching all the fixation lines with the corresponding text lines. To deal with the remaining error, we plan to use a saccade and fixation model such as SWIFT [5] or the E-Z Reader [12] to generate some fixation distributions corresponding to a given text. Then, by using our alignment method, we will compare the fixation distribution of the reader with this model.

Endnote

¹[<http://www.tobii.com/fr/eye-experience/eyex/>]

Acknowledgments

This work was supported in part by the JST CREST and the JSPS Kakenhi Grant Number 25240028 and 15K12172.

Competing interests

The authors declare that they have no competing interests.

Received: 24 February 2016 Accepted: 21 April 2016

Published online: 14 September 2016

References

- Augereau O, Kise K, Hoshika K (2015) A proposal of a document image reading-life log based on document image retrieval and eyetracking. In: Document Analysis and Recognition (ICDAR), 2015 13th International Conference on. IEEE. pp 246–250. doi:10.1109/ICDAR.2015.7333761
- Biedert R, Buscher G, Dengel A (2010) The eyebook—using eye tracking to enhance the reading experience. *Informatik-Spektrum* 33(3):272–281
- Biedert R, Hees J, Dengel A, Buscher G (2012) A robust realtime reading-skimming classifier. In: Proceedings of the Symposium on Eye Tracking Research and Applications ACM. pp 123–130. doi:10.1145/2168556.2168575
- Calvi C, Porta M, Sacchi D (2008) e5learning, an e-learning environment based on eye tracking. In: Advanced Learning Technologies, 2008. ICALT'08. Eighth IEEE International Conference on. IEEE. pp 376–380. doi:10.1109/ICALT.2008.35
- Engbert R, Nuthmann A, Richter EM, Kliegl R (2005) Swift: a dynamical model of saccade generation during reading. *Psychol Rev* 112(4):777
- Hyrskykari A (2006) Utilizing eye movements: overcoming inaccuracy while tracking the focus of attention during reading. *Comput Hum Behav* 22(4):657–671
- Hyrskykari A, Majaranta P, Aaltonen A, Rähkä K (2000) Design issues of idict: a gaze-assisted translation aid. In: Proceedings of the 2000 symposium on, Eye tracking research & applications. ACM. pp 9–14. doi:10.1145/355017.355019
- Kunze K, Kawaichi H, Yoshimura K, Kise K (2013) Towards inferring language expertise using eye tracking. In: CHI'13 Extended Abstracts on, Human Factors in Computing Systems. ACM. pp 217–222. doi:10.1145/2468356.2468396
- Martinez-Gomez P, Chen C, Hara T, Kano Y, Aizawa A (2012) Image registration for text-gaze alignment. In: Proceedings of the 2012 ACM international conference on Intelligent User Interfaces. ACM. pp 257–260. doi:10.1145/2168556.2168575
- O'Brien S (2009) Eye tracking in translation process research: methodological challenges and solutions. *Methodol, Technol Innov Translation Process Res* 38:251–266
- Rayner K (1998) Eye movements in reading and information processing: 20 years of research. *Psychol Bull* 124(3):372
- Reichle ED, Rayner K, Pollatsek A (2003) The ez reader model of eye-movement control in reading: comparisons to other models. *Behav Brain Sci* 26(04):445–476
- Stampe DM, Reingold EM (1995) Selection by looking: a novel computer interface and its application to psychological research. *Stud Vis Inf Process* 6:467–478
- Xu S, Jiang H, Lau F (2009) User-oriented document summarization through vision-based eye-tracking. In: Proceedings of the 14th international conference on Intelligent user interfaces. ACM. pp 7–16. doi:10.1145/2168556.2168575
- Yamaya A, Topić G, Martínez-Gómez P, Aizawa A (2015) Dynamic-programming-based method for fixation-to-word mapping. In: Intelligent Decision Technologies. Springer. pp 649–659. doi:10.1007/978-3-319-19857-6_55

Submit your manuscript to a SpringerOpen® journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com