Eye gaze and text line matching for reading analysis

Lima Sanches Charles
Osaka Prefecture University
1-1 Gakuen-cho, Naka, sakai,
Osaka, Japan
Ecole Nationale Superieure de
l'Electronique et de ses
Applications
6, avenue du Ponceau 95014
Cergy-Pontoise, France
charles-ls@hotmail.fr

Augereau Olivier
Osaka Prefecture University
1-1 Gakuen-cho, Naka, sakai,
Osaka, Japan
augereau.o@gmail.com

Kise Koichi
Osaka Prefecture University
1-1 Gakuen-cho, Naka, sakai,
Osaka, Japan
kise@cs.osakafu-u.ac.jp

UbiComp/ISWC '15 Adjunct, September 7-11, 2015, Osaka, Japan.

http://dx.doi.org/10.1145/2800835.2807936

Abstract
Eye tracking data has been widely used to analyze our reading behavior. Usually, experiments are carried out with head fixations or by analyzing eye tracking data in large areas such as paragraphs. But if we want to analyze the eye gaze line by line or word by word with a non invasive apparatus, we have to face the mislocation of the recorded eye gaze. The lack of accuracy involves a difficult analysis of the small eyes movements during reading. This paper proposes a method to match lines of gazes with corresponding text lines, using three different methods. We will show that the Dynamic Time Warping is a promising way to measure similarity between a line of gaze and a text line.

Author Keywords
eye tracker; gaze analysis; sequence alignment; reading understanding

ACM Classification Keywords
H.5.m [Information interfaces and presentation (e.g., HCI)]: Miscellaneous; H.5.2 [User Interfaces (D.2.2, H.1.2, I.3.6)]: Input devices and strategies

1 Introduction
Reading is an everyday task we perform all day long. Many researches have been carried out on our reading behavior. It is now possible to estimate English skill level [3], to sum-

marize a text [9] or to detect skimming behavior [1] by analyzing the way our eyes are moving during reading. According to [8], eye movements during reading are divided in two states: fixations and saccades. Fixation is when the eyes stare at a word during reading and last for about 250ms. Saccade is the quick movement of the eyes between two points of fixation.

During reading analysis, researchers have to deal with inaccuracy or miscalibration of an eye tracker [1]. Usually this issue is compensated by setting up specific conditions of experiment such as head fixation [6] or by analyzing our reading behavior statistically (using the average number of words per line of the document) [4]. Another strategy is to analyze our reading behavior on a paragraph scale [7]. But it could be interesting to analyze our reading behavior line by line or word by word using a non invasive method. However in such conditions, these problems lead to a mislocation of the recorded eye gaze as illustrated in Figure 1. In such case, reading analysis will be difficult to perform since the gap between the recorded eye gaze and the text is greater than the line spacing. For a given line of fixations it is not obvious to know the corresponding text line. Several tasks in reading analysis process, such as counting read words or detecting difficult part in a text, would be much easier if it were possible to match the recorded eye gaze with the text because it would be possible to know which line or which word is read.

This paper proposes an algorithm to find the accordance between lines of gazes and lines of text. Our approach is based on the use of an eye tracker designed for the general public. First we will present the different steps of our algorithm and propose different alternatives to do the matching. Second we will present the experiment and the corresponding results. We will show we can use a sequence alignment algorithm to match the lines of gazes with the lines in the text and get an accuracy of 60% for the matching. Then we will conclude and discuss about ways to improve the algorithm.

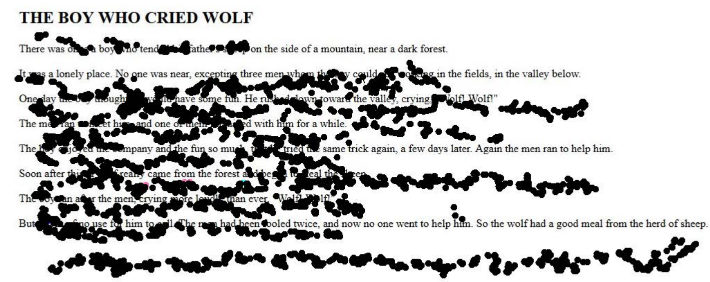

Figure 1: The mislocation of the eye gaze variate with the conditions of recording. If the mislocation is greater than the line spacing, the reading analysis cannot be done precisely.

2 Algorithm overview

Our algorithm is divided into four steps. The input of the algorithm is the raw eye gaze as in the example of Figure 1.

1. To detect fixations during reading.
2. To detect line breaks to gather fixations into lines in order to compare them with text lines.
3. To rate the similarity between a line of fixations and a line of the text. In this step we will present three different methods of rating.
4. The line matching where the line of fixations is matched with the corresponding text line.

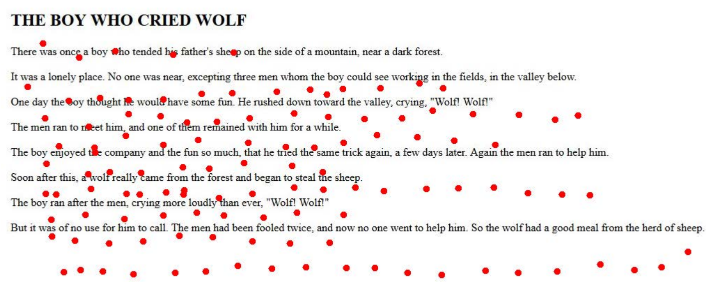

Figure 2: Output of the fixation detection algorithm

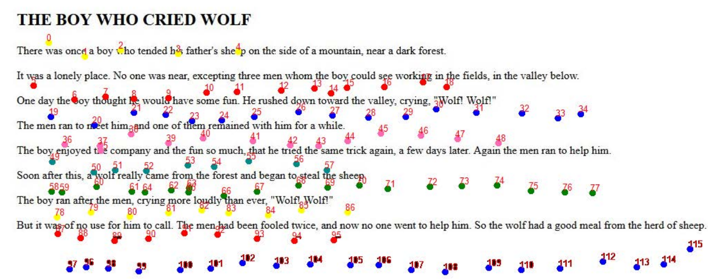

Figure 3: Line break detection, each new color is a different line

2.1 Fixation detection

Our eye movements during reading is not a continuous motion but a succession of fixations and saccades. However the output of the eye tracker is a continuous recording. We use the fixation detection algorithm from [2] to extract fixations from the raw eye gaze. In this algorithm, the gazes which are near each others are gathered into fixations. The output of the algorithm is shown in Figure 2

2.2 Line break detection

When fixations are detected, we gather them into lines of fixations in order to compare them with the text lines. To find lines of fixations we try to detect the line breaks. A line break happen when we finish to read one line and start to read a new one. It can be detected when a large regression occurs. Considering $x_f(i)$ as the x-coordinate of the fixation i, if

$$x_f(i+1) - x_f(i) < -L < 0 \quad (1)$$

then a line break is detected. L is chosen to detect the regression which correspond to a line break. The output of this algorithm is shown in Figure 3. Errors in segmentation can occur in case of rereading (going backward could be detected as a line break regression even though we are rereading the same line), or in case of one or two words text lines (the regression will not be detected and the line we are reading will be merged with the next one).

2.3 Fixations line and text line comparison

In this section we will talk about three ways to compare a line of fixations with a line in the text. The output of each method of comparison is a score we will use in the final matching step.

2.3.1 Fixation number

First, we try a basic idea to match a line of fixations with a line of the text. For a given line of fixations we compare the number of fixations in this line N_f with the number of words in a line of the text N_w. Then, we compute the score

$$s = |N_f - N_w| \quad (2)$$

The line of fixations and the line in the text are similar if there are as many fixations as words.

2.3.2 Line length

Another basic idea to match a line of fixations with a line in the text is to compare the length of these two lines. In this algorithm words were replaced by their center and the x-coordinate of the word is defined as the x-coordinate of its center. For a given line of fixations, we compute the distance between the last and the first fixation, L_f and we compute the distance between the last and the first word in the line of the text L_l. Then we define the score

$$s = |L_l - L_f| \tag{3}$$

The line of fixation and the line in the text are similar if they have the same length.

2.3.3 Sequence alignment

The principle is to find the best alignment between two sequences. We choose to align the x-position of words in the text with x-position of fixations by using Dynamic Time Warping (DTW) algorithm. Again, in this algorithm the x-coordinate of a word is defined as the x-coordinate of the center of the word. So we have two align two sequences:

$$X_f = x_1, x_2, x_3 ... x_n \tag{4}$$

and

$$X_t = x'_1, x'_2, x'_3 ... x'_m \tag{5}$$

In the DTW algorithm it is necessary to define a "cost of alignment".

In our case the cost of alignment between a fixation $x_i$and a word x'_j is defined as

$$d = |x_i - x'_j| \tag{6}$$

which is the distance in pixel between the fixation i and the word j. With these two sequences we compute a matrix $M(n, m)$ as in Figure 4. And the matrix M has to be filled

$$M = \begin{array}{c|ccc} & x1 & x2 & x3 \\ \hline x'1 & 0 & \infty & \infty \\ x'2 & \infty & e & f \\ x'3 & \infty & g & h \end{array}$$

Figure 4: M matrix with for instance h = $|x_3 - x'_3|$ + Min (e,f,g)

as follows. The first line is filled with infinite value, the first column is also filled with infinite values. $M(0, 0)$ is filled with the value 0. Then for each position in the matrix M a score is defined as

$$M[i, j] = d + \min(M[i-1, j], M[i, j-1], M[i-1, j-1]) \tag{7}$$

This methodology is maintained until we reach the coefficient $[n, m]$ of the matrix M (low right corner). This last value represents the score of the matching between the line of fixation X_f and the line in the text X_t.

$$s = M[n, m] \tag{8}$$

2.4 Line matching

The point of this step is to find the best matching between one line of fixations and several lines of text. This matching is based on the scores from the previous step. For a given line of fixation we will compare it with the nearest n text lines and try to find which one correspond to this fixations line. The mislocation due to the inaccuracy of the eye tracker can vary but is not greater than a couple of centimeters. According to this measure, we choose $n = 5$. The y-coordinate of a line is considered as the center of its bounding box. Then we look at the pair (fixations line-text line) with the minimum score (according to the previous step) to find the best matching.

3 Experiment and Results

For the experiment, 8 participants were asked to read 3 different texts. The distance between the screen and the subjects was between 60 and 70 centimeters. Readers were asked to read carefully forward the text. In order to avoid errors due to errors of segmentation during the line break detection step, we selected only the well segmented lines as input of the algorithm. Then we manually labelled the position of each fixations line. The number of selected lines is 138. A good matching is detected when the line of fixations is matched with the corresponding line in the text according to the ground truth. The eye tracker employed in this experiment is a Tobii EyeX Controller [1].

The accuracy of the algorithm is computed as the percentage of all good matches on a total amount of 138 lines. Comparison between the three methods of matching is shown in Table 1. The accuracy of the DTW alignment method is better than comparing the number of fixations with the number of lines. It is slightly better than comparing the length of line of fixations with the length of the line in the text. We can see a visualization of an alignment with a high score in Figure 6 and a visualization of an alignment with a low score in Figure 5. Both using DTW algorithm.

3.1 Results analysis

Comparing the number of fixations with the number of words is not very accurate because there is not necessarily one fixation per words. We tend to skip small words during reading which can affect this method of comparison. Moreover confusions can occur when successive lines contains

[1][http://www.tobii.com/]

Algorithm	% of good line matching
Number of fixations	39%
Line length	56%
DTW	60%

Table 1: Results of the algorithm with three different methods of matching

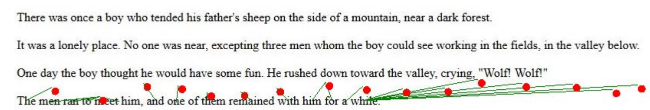

Figure 5: Result of an alignment with a low score using DTW algorithm.

the same number of words.

Comparing the length of the fixations line with the length of the text line is more accurate, but if the text is justified all the lines will have the same length. Then it will be very hard to match a line of fixations with the corresponding text line using this method.

Using DTW for the matching is more accurate, because the number and the position of the fixations are both taken into account. Even if some small words does not have any corresponding fixation or some long words have more than one fixations, these differences can be absorbed by the elasticity property of the DTW. But this is true only for small changes and have its limits. And, still, some text lines can be very similar to each other and introduce some confusions for the DTW.

4 Conclusion and discussion

We have presented a way to match recorded fixations with the text using the DTW. Our algorithm can match a line of

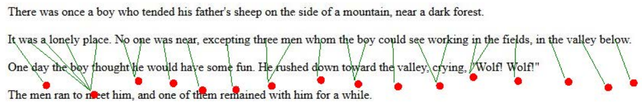

Figure 6: Result of an alignment with a high score using DTW algorithm.

fixations with the corresponding text line with an accuracy of 60%. In this paper we focused on finding the correct matching between one line of fixations with several line of text. This algorithm can be used for estimating and correcting the global mislocation for one specific document. We know that a majority of fixation lines are successfully matched to the correct text lines with the presented algorithm. So, the corresponding vertical translation can be applied to adjust all the fixations of the document closer to the correct lines.

A way to enhance our results is to use psychological reading behavior. For instance, according to [8] we only read 80% of contents words and 35% of function words in a whole text. Moreover, the x-position of a word in the text is defined as the x-position of the center of the word. But according to [5], the fixation are not exactly located at the center of the word.

Another different perspective is to use the DTW for applying an horizontal alignment. As we can see in Figure 6, it might be possible to match each fixation with the corresponding word. But for now we can't evaluate it because no ground truth is available at the word level.

5 REFERENCES

1. Ralf Biedert, Jörn Hees, Andreas Dengel, and Georg Buscher. 2012. A robust realtime reading-skimming classifier. In *Proceedings of the Symposium on Eye Tracking Research and Applications*. ACM, 123–130.
2. Georg Buscher, Andreas Dengel, and Ludger van Elst. 2008. Eye movements as implicit relevance feedback. In *CHI'08 extended abstracts on Human factors in computing systems*. ACM, 2991–2996.
3. Kai Kunze, Hitoshi Kawaichi, Kazuyo Yoshimura, and Koichi Kise. 2013a. Towards inferring language expertise using eye tracking. In *CHI'13 Extended Abstracts on Human Factors in Computing Systems*. ACM, 217–222.
4. Kai Kunze, Hitoshi Kawaichi, Kazuki Yoshimura, and Kenji Kise. 2013b. The wordometer–estimating the number of words read using document image retrieval and mobile eye tracking. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*. IEEE, 25–29.
5. George W McConkie, Paul W Kerr, Michael D Reddix, and David Zola. 1988. Eye movement control during reading: I. The location of initial eye fixations on words. *Vision research* 28, 10 (1988), 1107–1118.
6. Takehiko Ohno. 2004. EyePrint: Support of document browsing with eye gaze trace. In *Proceedings of the 6th international conference on Multimodal interfaces*. ACM, 16–23.
7. Ayano Okoso, Kai Kunze, and Koichi Kise. 2014. Implicit gaze based annotations to support second language learning. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*. ACM, 143–146.
8. Keith Rayner. 1998. Eye movements in reading and information processing: 20 years of research. *Psychological bulletin* 124, 3 (1998), 372.

9. Songhua Xu, Hao Jiang, and Francis Lau. 2009. User-oriented document summarization through vision-based eye-tracking. In *Proceedings of the 14th international conference on Intelligent user interfaces*. ACM, 7–16.