Reading With Your Eyes

1st Author

1st author's affiliation  
1st line of address  
2nd line of address  
Telephone number, incl. country code

1st author's E-mail address

2nd Author

2nd author's affiliation  
1st line of address  
2nd line of address  
Telephone number, incl. country code

2nd E-mail

Ashkan Bozorgzad

Ph.D. student, Department of Civil and Environmental Engineering, The University of Iowa  
Iowa City, IA, 52242  
+1 630 229 9734

ashkan-bozorgzad@uiowa.edu

**ABSTRACT**

In this paper, we describe the formatting guidelines for ACM SIG Proceedings.

**Keywords**

Keywords are your own designated keywords separated by semicolons (“;”).

# INTRODUCTION

The proceedings are the records of the conference. ACM hopes to give these conference by-products a single, high-quality appearance. To do this, we ask that authors follow some simple guidelines. In essence, we ask you to make your paper look exactly like this document. The easiest way to do this is simply to download a template from [2], and replace the content with your own material.

# RELATED WORKS

All material on each page should fit within a rectangle of 18 × 23.5 cm (7" × 9.25"), centered on the page, beginning 1.9 cm (0.75") from the top of the page and ending with 2.54 cm (1") from the bottom. The right and left margins should be 1.9 cm (.75"). The text should be in two 8.45 cm (3.33") columns with a .83 cm (.33") gutter.

# GOALS

This project focuses on developing techniques to convert the eye tracking into the precision text detecting during the readings for children. In order to achieve this goal three things are required as following:

1. A software environment which contains the code, texts that should be detected, pictures and other objects to make the environment friendly for children.
2. An eye tracker device: in this project the Tobii eye tracker is used. This eye tracker connects to the computer with the USB cable and works with Windows 8 and10.
3. A user, children, who wants to detect the text.

The outputs of the designed software are two main things:

1. the highlighted text of the detected word
2. the voice of the detected word

Some challenges should be considered and addressed in this project as following:

1. Finding some techniques in order to detect the eyes continuously and constantly. An individual differences between eyes, an eye tracker coverage degree and user movements are some reasons of disconnection between the eyes and the eye tracker that should be considered.
2. There are multiple ways to detect object (Text). Selection of the best way that lead to high accuracy is desire in this project
3. Since the Tobii eye tracker was used in this project, using of Tobii eye engine was unavoidable. This engine has some limitations. Some of these limitations are challenging.

In the following sections particularly, in the implementation and design section it has been tried to explain how to overcoming to these challenges.

# DESIGN

Place Tables/Figures/Images in text as close to the reference as possible (see Figure 1). It may extend across both columns to a maximum width of 17.78 cm (7”).

Captions should be Times New Roman 9-point bold. They should be numbered (e.g., “Table 1” or “Figure 2”), please note that the word for Table and Figure are spelled out. Figure’s captions should be centered beneath the image or picture, and Table captions should be centered above the table body.

# IMPLEMENTATION

The heading of a section should be in Times New Roman 12-point bold in all-capitals flush left with an additional 6-points of white space above the section head. Sections and subsequent sub- sections should be numbered and flush left. For a section head and a subsection head together (such as Section 3 and subsection 3.1), use no additional space above the subsection head.

# EXPERIMENTS

In order to increase the accuracy of the system two sets of experiment were design. The result of the first experiment were used to determine the reasonable textbox size and the second experiment was about determination of delay amount for the software. The results and more details about these two experiments were explained in the following sections.

## Experimentation for the Textbox Size

In this project textbox (rectangular object shape) was used for detecting the words. The ideal size of these textboxes are the word size. However, some of the words like determination words (a, an, the) are too small and cannot be detected with high accuracy by the eye tracker. In order to measure the reasonable textbox size, different textbox sizes were crated and the accuracy of the software was measured. The size with higher accuracy is the desired size.

To create different textbox sizes the horizontal and vertical grid lines were used. The constant window size was divided to multiple columns and rows with these horizontal and vertical grid lines. The more number of horizontal and vertical lines create smaller textbox size and vice versa. For example, 9 \* 23 means the window has 9 rows and 23 columns. Figure 1 shows the 11 \* 11 textbox size as a sample.

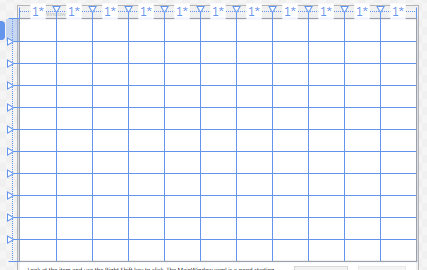


Figure 11\*11 textbox size

In order to determine the reasonable minimum textbox size seven different text box sizes were chosen. First, the 9\*9, 11\*11 and 13\*13 were used to determine the reasonable number of rows (the height of the textbox) then according to the results of these experiments the 9\*15, 9\*17, 9\*19 and 9\*23 were used to calculate reasonable number of columns(the width of the textbox). Two trials were conducted for each textbox size by 4 persons. In each test, 10 sets of random number (row, column) were generated and asked user to look at these cells and detect them. Therefore, for each size total 80 cells were tested for detection (4 people \* 2 trials \* 10 points). The accuracy were defined as following was used to figure out which textbox size should be selected in the final version of the software:

Table 1 shows the sample of requested and actual detected cells by user for the 11\*11 textbox size. If the detected and required cell was same the detected cell was detected correctly and if they were not same it was detected wrongly and highlighted with red color. It should be mentioned that the calibration was done before each test on this experiment.

Table The result of one of theexperiments for 11\*11 textbox size

|  |  |  |  |
| --- | --- | --- | --- |
| **Trial 1** | | **Trial 2** | |
| **Request cell** | **Actual Detected Cell** | **Request cell** | **Actual Detected Cell** |
| (9, 9) | (9, 9) | (10, 11) | (10, 11) |
| (8, 1) | (8, 1) | (5, 4) | (5, 4) |
| (10, 7) | (10, 7) | (7, 6) | (6, 6) |
| (3, 6) | (2, 6) | (2, 4) | (2, 4) |
| (9, 9) | (9, 9) | (8, 9) | (8, 9) |
| (2, 8) | (2, 8) | (5, 7) | (5, 7) |
| (1, 8) | (1, 8) | (4, 3) | (4, 3) |
| (2, 1) | (1, 1) | (7, 1) | (5, 1) |
| (7, 2) | (6, 2) | (5, 1) | (4, 1) |
| (9, 2) | (8, 2) | (8, 7) | (7, 7) |

As can be seen in this table the 60% accuracy was calculated for both trials. Figure 2 illustrates the accuracy of three sample sizes.

Figure the accyracy of three sample sizes

According to the Figure 2, it should be noted that the results of experiments are not expected since higher value of accuracy is observed in smaller textbox size. For example, 11\*11 case has smaller textbox size, but higher values of accuracy than 9\*9 case. However, analyzing the causes of the errors in the experiments results shows that the percentage of errors comes from wrongly detected row of the cells in 11\*11 case is more than 9\*9 case. Therefore, 9 rows was selected as the reasonable and logical size of the rows.

After determination of the height of the textbox, the additional tests for 9\*15, 9\*17, 9\*19 and 9\*23 were conducted in order to find the reasonable width of the textbox. Figure 3 indicates the results of these experiments and 9\*9 case.

Figure 3 the accuracy of 9 rows samples with different number of columns

As illustrated by Figure 3, 9\*19 case has the higher accuracy; therefore, 19 columns was selected for width of the textbox size. The result of these sets of experiment also do not compromise with the author’s expectation. It seems that the following reasons can be mentioned to explain the causes of this variability:

1. Learning process of users during the tests: for all users the 9 \* 9 case was the first test. It seems that a major parts of errors in this case comes because of an unfamiliarity of users. After this case the users had some experience and did the test better.
2. High sensitivity of the calibration process: before doing each test, the user did calibration. This process is very sensitive and can affect the results.
3. Limited number of the tests: in this project only 4 people test results were used because of time limitation. If more number of users was used, the better results would be expected.
4. Limited number of the experiments: 4 users in this project have different physical and appearance features. One of them wore glasses and another one had contact lenses. Also the eye shape of peoples were totally different.

In this experiment the reasonable textbox size was desired in order to increase the accuracy of the system. The 9\*23 grid lines was calculated for the text box size.

## Experiment for the Delay Amounts

Delay in this project is defined as the amount of time that the user should look at the object in order to detect it. The delay is necessary in the system to avoid random selection of object due to the rapid eye movement. In order to set up this experiments 6 value of delay were defined in the system. These value are 200, 300, 400, 450, 500 and 600 milliseconds. Then, the user was asked to rank each case from one to five. Table 2 shows the scale weight and its meaning.

Table 2 Scale weight and its meaning

|  |  |
| --- | --- |
| **Scale weight** | **Meaning** |
| 1 | Strongly Disagree |
| 2 | Disagree |
| 3 | Neither agree nor disagree |
| 4 | Agree |
| 5 | Strongly Disagree |

Four users participated in this experience and the result of the tests can be find in the Figure 4.

Figure 4 The results of delay experiments

In this figure each color represents each delay value and the vertical axes shows the number of people (Maximum 4) that vote for each delay. For example, for the 600 millisecond delay, 3 people were strongly disagree with it. By using the weight average method, the average value of interest for each delay category was calculated and shown in the Figure 5.

Figure 5 the average value of interest for each delay amount

As indicated in this figure, the 300 ms and 400 ms have the higher value; therefore, the delay of 305 ms was applied to the main software.

In this section two experiments which were conducted in order to increase the accuracy of the system were explained among their results and application in the main software. Although the result of textbox size experiment is weird, still they are applicable in the software. The result of the delay experiment was logical and tangible.

# LESSONS LEARNED

Our thanks to ACM SIGCHI for allowing us to modify templates they had developed.

# CONCLUSION

Our thanks to ACM SIGCHI for allowing us to modify templates they had developed.

# FUTURE WORKS

Our thanks to ACM SIGCHI for allowing us to modify templates they had developed.

# REFERENCES

1. Bowman, M., Debray, S. K., and Peterson, L. L. 1993. Reasoning about naming systems. *ACM Trans. Program. Lang. Syst.* 15, 5 (Nov. 1993), 795-825. DOI= <http://doi.acm.org/10.1145/161468.16147>.
2. Ding, W. and Marchionini, G. 1997. *A Study on Video Browsing Strategies*. Technical Report. University of Maryland at College Park.
3. Fröhlich, B. and Plate, J. 2000. The cubic mouse: a new device for three-dimensional input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (The Hague, The Netherlands, April 01 - 06, 2000). CHI '00. ACM, New York, NY, 526-531. DOI= <http://doi.acm.org/10.1145/332040.332491>.
4. Tavel, P. 2007. *Modeling and Simulation Design*. AK Peters Ltd., Natick, MA.
5. Sannella, M. J. 1994. *Constraint Satisfaction and Debugging for Interactive User Interfaces*. Doctoral Thesis. UMI Order Number: UMI Order No. GAX95-09398., University of Washington.
6. Forman, G. 2003. An extensive empirical study of feature selection metrics for text classification. *J. Mach. Learn. Res.* 3 (Mar. 2003), 1289-1305.
7. Brown, L. D., Hua, H., and Gao, C. 2003. A widget framework for augmented interaction in SCAPE. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology* (Vancouver, Canada, November 02 - 05, 2003). UIST '03. ACM, New York, NY, 1-10. DOI= <http://doi.acm.org/10.1145/964696.964697>.
8. Yu, Y. T. and Lau, M. F. 2006. A comparison of MC/DC, MUMCUT and several other coverage criteria for logical decisions. *J. Syst. Softw.* 79, 5 (May. 2006), 577-590. DOI= <http://dx.doi.org/10.1016/j.jss.2005.05.030>.
9. Spector, A. Z. 1989. Achieving application requirements. In *Distributed Systems*, S. Mullender, Ed. ACM Press Frontier Series. ACM, New York, NY, 19-33. DOI= <http://doi.acm.org/10.1145/90417.90738>.

Columns on Last Page Should Be Made As Close As Possible to Equal Length