Information Security Display via Uncrowded Window

Zhongpai Gao #1, Guangtao Zhai #2, Jiantao Zhou *3, Xiongkuo Min #4, Chunjia Hu #5

Institute of Image Communication and Information Processing,
Shanghai Jiao Tong University, Shanghai, China
{gaozhongpai, zhaiguangtao, minxiongkuo, hcj-sjtu}@sjtu.edu.cn

* *Faculty of Science and Technology, University of Macau, Macau, China*
* jtzhou@umac.mo

***Abstract*—With the booming of visual media, people pay more and more attention to privacy protection in public environments. Most existing research on information security such as cryptography and steganography is mainly concerned about transmission and yet little has been done to prevent the information displayed on screens from reaching eyes of the bystanders. This "security of the last foot (SOLF)" problem, if left without being taken care of, will inevitably lead to the total failure of a trustable information communication system. To deal with the SOLF problem, for the application of text-reading, we proposed an eye tracking based solution using the newly revealed concept of uncrowded window from vision research. The theory of uncrowded window suggests that human vision can only effectively recognize objects inside a small window. Object features outside the window may still be detectable but the feature detection results cannot be efficiently combined properly and therefore those objects will not be recognizable. We use eye-tracker to locate fixation points of the authorized reader in real time, and only the area inside the uncrowded window displays the private information we want to protect. A number of dummy windows with fake messages are displayed around the real uncrowded window as diversions. And without the precise knowledge about the fixations of the authorized reader, the chance for bystanders to capture the private message from those surrounding area and the dummy windows is very low. Meanwhile, since the authorized reader can only read within the uncrowded window, detrimental impact of those dummy windows is almost negligible. The proposed prototype system was written in C++ with SDKs of Direct3D, Tobii Gaze SDK, CEGUI, MuPDF, OpenCV and etc. Extended demonstration of the system will be provided to show that the proposed method is an effective solution to SOLF problem of information communication and display.**

***Index Terms*—Display technology, information security, uncrowded window, eye tracking**

I. Introduction

With the development of information technology, especially with the wide popularity of mobile devices, more and more people are concerned about information security. Although there are a mount of research, cryptography and steganography are mainly concerned about secure information transmission and much less has been done to protect those information displayed on screens from reaching the eyes of bystanders. Currently, one popular solution to this problem is 3M's privacy filter, which is a micro-louvre that reduces visibility angles, fencing bystanders. However, the privacy filter still cannot prevent from peeking from behind. Another solution is using the technology of temporal psychovisual modulation (TPVM) [1]. This method overcomes the problem of 3M's privacy filter, but it is not convenient that user need wear glasses to see the private information.

This paper aims to develop a better solution to information security display based on the idea of the uncrowded window of object recognition [2] combined with the technology of eye tracking. Most of our visual field is crowded most of time, sparing only a central uncrowded window. This window and the limitation it places on recognition are especially clear in the case of reading. Inside of the window, letters are uncrowded and we can read them. Outside of the window, letters are crowded and we cannot. To read the letters which now are outside the window we must move our eyes to bring the window to those letters. The size of the uncrowded window is determined by the observer's critical spacing and the spacing of the viewed objects together [3]. It is the region where object spacing exceeds critical spacing. Critical spacing at the cortex is independent of object position, and critical spacing at the visual field is proportional to object distance from fixation. It is found that when the target and flankers have similar features, these type cases produce maximum crowding [4]

In this work, we build an information security display system for e-book reader using an eye tracker, *Tobii REX Developer Eidtion* [5]. *Tobii REX Developer Eidtion* is a portable and sleek USB eye-tracking device that fits on both desktop computer screens and laptops. When reading in the e-book reader application, a window will appear in the fixation point of user whose eyes are tracked by the eye tracker. The size of the window is equal to the uncrowded window. Inside of the window, there are words of private texts. Outside of the window, there are words unrelated to the private texts, called dummy texts. When the fixation point moves, the window will follow it because the eye tracker can follow the eyes and locate where it is. So the user will see the private information all the

IEEE VCIP'14, Dec. 7 - Dec. 10, 2014, Valletta, Malta.
978-1-4799-6139-9/14/$31.00 ©2014 IEEE.

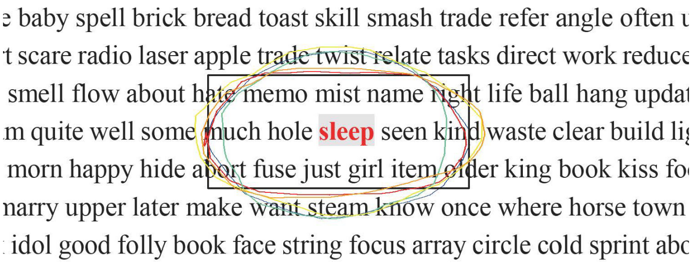


Fig. 1. Test for determining the size of the uncrowded window. Five circles marked in the figure are drawn by five testers. The black rectangle is the result of the uncrowded window.

time. The words of dummy texts outside of the window are crowded, which means they will not affect the reading of the user. For bystanders, they will mainly see the dummy texts because they do not know where the user see.

In this developed system, we implement the application of e-book reader. The system is written in C++ language in the integrated development environment (IDE) of Microsoft Visual Studio 2010. SDKs from DirectX [6], CETUI [7], MuPDF [8], OpenCV [9], Boost [10] and Tobii Gaze SDK [11] are used to realize the system. DirectX is a collection of application programming interfaces to handle multimedia related tasks. And in this work, DirectX is used to combine the private texts and dummy texts. CEGUI is used to build the GUI of this system due to its flexibility and high compatibility with Direct3D. OpenCV is an open source library containing more than 500 optimized algorithms for image and video analysis and the module of Kalman filter is used to reduce the noise of gaze position. MuPDF, a free and open source software library supporting PDF and XPS parsing and rendering is used to get the pages of PDF files. Boost provides free peer-reviewed portable C++ source libraries, and the module of thread library is used to get gaze position of eyes in real time. Tobii Gaze SDK is used for tracking the eyes and getting the raw data of gaze position.

The reminder of the paper is organized as follows. Section II is to determine the size of the uncrowded window. Section III shows the design and implement of the system. The system test and result analysis will be presented in section IV. At last, the conclusion is given in section V.

II. Size of the Uncrowded Window

The core issue of this system is to determine the size of the uncrowded window. The size of the real window should be equal to the uncrowded window, that makes the user read more comfortably. If the size is larger than the uncrowded window, bystanders will be easy to notice the private texts. If it is smaller than the uncrowded window, some peripheries will be inside of the uncrowded window which will distract the attention of the user and make reading difficult. In this section, we designed a test to determine the size of the uncrowded window.

First, we make a text as shown in Fig. 1. The formats of the text is uniform and the average length of words is about four. Let tester fixate on the red *'sleep'* in the center of the text and draw the area where the words can be recognized. To avoid the tester moving the fixation point to other words, we design a program to find the user's eyes and move the text along with the gaze positon. So that the tester will focus on the red *'sleep'* all the time and the uncrowded area he/she drawn will be more accurate. We set the value of the distance from tester to the *Tobii REX Developer Eidtion* to $60cm$. There are five students participating in this test. The circles in Fig. 1 are drawn by these students. We find that it is hard to recognize the words over one line in vertical direction and over two words in horizontal direction. So the uncrowded window can be described as the black rectangle shown in Fig. 1. When the distance from user to the screen changes in the range from $50cm$ to $80cm$ in which *Tobii REX Developer Eidtion* works, the size of the uncrowded window changes a little. So the size of the uncrowded we get should be universal for a wide range of distances.

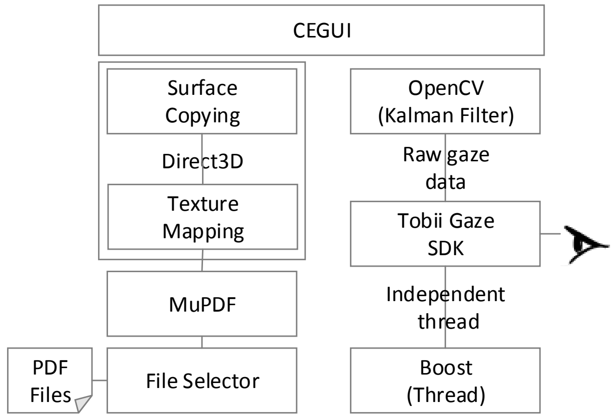


Fig. 2. Diagram of the proposed system.

III. Design and Implementation of the System

In the following two subsections, system overview and implementation detail will be introduced respectively. To filter gaze position, map texture and copy surface are two main jobs in this system.

A. System overview

Fig. 2 shows the functional level diagram of the proposed system. The modules function in low to high levels from bottom to top. File selector is a class which is convenient for user to select a PDF file from the disk. MuPDF is a PDF and XPS parsing and rendering engine which can extract pages from PDF files and save them as images. Direct3D plays an important role in this system and mainly contains two modules, surface copying and texture mapping. It is used to combine the dummy texts and the private texts and present the result in the screen. The thread component of Boost allows C++ programs to execute as multiple, asynchronous, independent threads-of-execution. We create a thread for eye tracker using *Boost.Thread*, so we can get the gaze position at any time. Tobii Gaze SDK can get the gaze position from eye tracker.

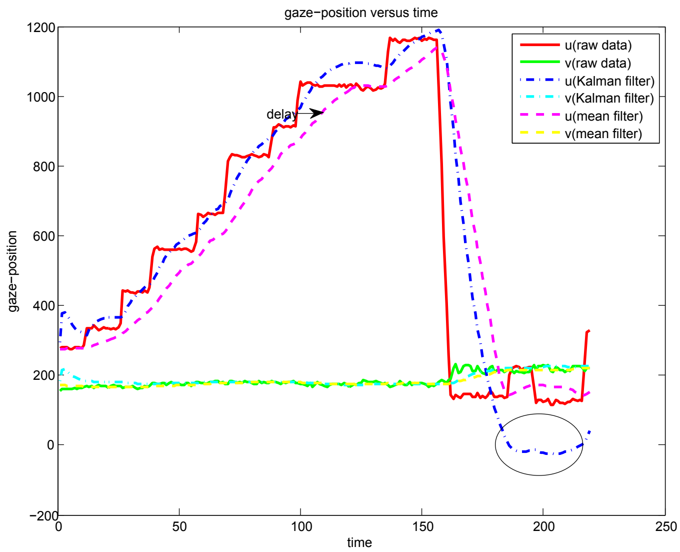


Fig. 3. Gaze position versus time. u and v represent the horizontal component and vertical component of gaze position. Solid line, dashed line and dashed line represent the gaze-positions of raw data, data filtered by Kalman filter and mean filter respectively.

Because the measured position will resemble Gaussian noise with a slow change of the mean, Kalman filter in OpenCV is used to reduce the noise of gaze-data. CEGUI is the graphical user interface of the system, which is convenient for user to interact with the system.

B. Design and implement the system

1) Gaze position filtering: When the user fixates on a point at the screen, the measured position will resemble Gaussian noise with a slow change of the mean [12]. The high frequency noise is measurement noise and the low frequency changes are due to a drift caused by pupil size changes and head movements. In this system, we need the gaze position to control the position of the uncrowded window in e-book reader application, it is important that the window follows the eyes in a way that feels natural by the user. Due to the Gaussian noise, a filter is required not only to make the window stable enough to allow the user to aim at the area of interest, but also to respond promptly to quick jumps that the user might make to another area of interest on the screen, such as jumping to the next line. So a carefully designed filter is required. A simple low-pass filter used to filter out the hight frequency measurement noise makes the response to large changes in gaze position sluggish. This can be very frustrating for the user, because the user has to wait for the window to move to the new area of interest. We use Kalman filter [13] to reduce the noise.

We get the gaze position data by the software of Tobii Studio from a test that a subject read a text carefully. Then we process the gaze positions by Kalman filter and mean filter, the results are shown in Fig. 3. Compared to the gaze positions, both Kalman filer and mean filter can smooth the data well. But this is a real-time system. We can find that mean filter has a large delay as marked in Fig. 3, while the blue dashed line of kalman filter just goes through the red solid line of raw data. And it performances well. One drawback of Kalman filter is that when reading to the next line from right side to left side, the gaze position will moves too much to the left (the area in the black circle in Fig. 3). This is an issue to be solved in future work.

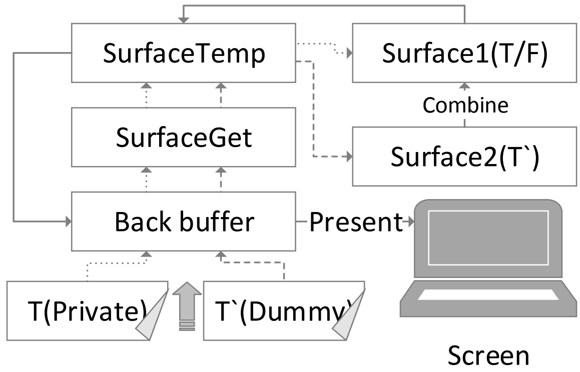


Fig. 4. Flow diagram of surface copying.

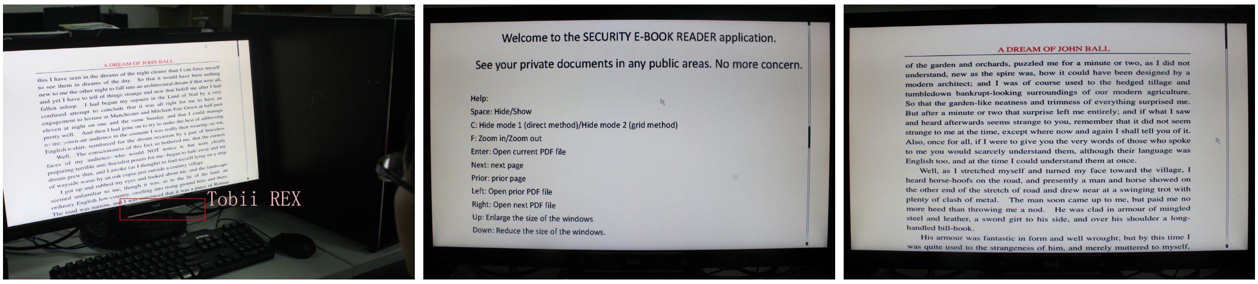
2) Texture mapping and surface copying: When we get the image of one PDF file page, how to present the image becomes the main problem to the system technically. This can be separated into two steps: texture mapping and surface copying.

First, we create four vertexes in the 3D space of Direct3D to make a rectangle whose aspect ratio is the same as the image. Then adjust the camera in Direct3D so that the rectangle just fill the whole screen. And we load the image which is extracted from PDF files as a Direct3D texture to map to the rectangle. Therefore, we can render the images of PDF files to the back buffer of Direct3D and present to the screen. We can adjust the camera to zoom in or zoom out for easy reading. This is called texture mapping.

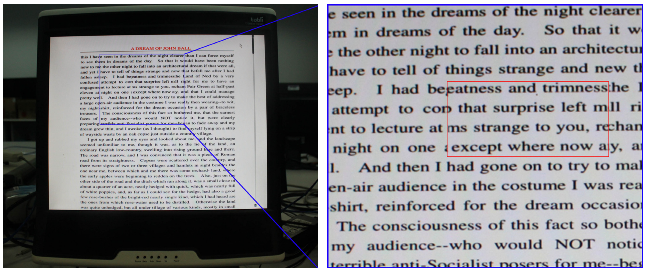
To realize information security, we need to present private texts and dummy texts at the same time. And this can be achieved by surface copying, as shown in Fig. 4. First, we create four Direct3D surfaces (*Surface1*, *Surface2*, *SurfaceGet* and *SurfaceTemp* in Fig. 4). *Surface1* and *Surface2* are created in system memory to speed up the process of copying data pixel by pixel. *SurfaceTemp* is as a temporary surface that copies surface from GPU to CPU or CPU to GPU and *SurfaceGet* gets surface from the back buffer. We render the private texts T and dummy texts T' (texture mapping) to the back buffer one after another, and copy them to *Surface1* and *Surface2* through *SurfaceGet* and *SurfaceTemp* respectively. Then combine these two surfaces, the result F stores back in *Surface1* and copy it to back buffer through *SurfaceTemp*. At last, we present the final image in back buffer to the screen.

IV. THE SYSTEM TEST AND RESULT ANALYSIS

Fig. 5 shows the demonstration and snapshot of the prototype information security display system. Fig. 5(a) is a live show of the system. Fig. 5(b) shows the welcome page of the



(a) System in action (b) The welcome page (c) Reading in *normal mode*



(d) Reading in *secret mode* (e) Zooming in on (d)

Fig. 5. Demonstration and snapshot of the prototype information security display system.

system. And the user guide is provided here. Fig. 5(c) shows the picture that user is reading in *normal mode*. There are no private texts and the contents can be shared with different people in *normal mode*. Fig. 5(d) shows that user reads in *secret mode*. Fig. 5(e) is the zooming in on Fig. 5(d) as marked in blue rectangle, and there is a window marked by a red rectangle and red arrow in it. It is hard to find the window in Fig. 5(d). Even the window moves, if without full attention, few people will notice it.

From Fig. 5(d), we can find that the private texts can be hidden quite well. It is important to point out that no single security method can provide total protection. Each has advantages and disadvantages and can be used in different situations. In this system, we just consider PDF files of texts, so files containing multiple features (such as images, texts, tables etc.) can be taken into account in the future work. To make the user feel more natural, filter for measured gaza position also can be improved.

V. CONCLUSION

The paper introduces a prototype system of information security based on the idea of uncrowded window combined with the technology of eye tracking. The theory of uncrowded window suggests that human vision can only effectively recognize objects inside a small window. Object features outside the window may still be detectable but the feature detection results cannot be efficiently combined properly and therefore those objects will not be recognizable. We use eye-tracker to locate fixation points of the authorized reader in real time, and only the area inside the uncrowded window displays the private information we want to protect. A number of dummy windows with fake messages are displayed around the real uncrowded window as diversions. The proposed prototype system was written in C++ with SDKs of Direct3D, Tobii Gaze SDK, CEGUI, MuPDF, OpenCV and etc. Extended demonstration of the system shows that the proposed method is an effective solution to “security of the last foot” (SOLF) problem of information communication and display.

ACKNOWLEDGMENT

This work was supported in part by NSFC (61371146, 61422112, 61331014), FANEDD (201339) and the Research Committee at University of Macau under grant MRG021/ZJT/2013/FST.

REFERENCES

[1] X. Wu and G. Zhai, “Temporal psychovisual modulation: A new paradigm of information display [exploratory dsp],” *Signal Processing Magazine, IEEE*, vol. 30, no. 1, pp. 136–141, 2013.
[2] D. G. Pelli and K. A. Tillman, “The uncrowded window of object recognition,” *Nat Neurosci*, vol. 11, pp. 1129–1135, Sept. 2008.
[3] D. G. Pelli, C. W. Burns, B. Farell, and D. C. Moore-Page, “Feature detection and letter identification,” *Vision research*, vol. 46, no. 28, pp. 4646–4674, 2006.
[4] M. Martelli, G. Di Filippo, D. Spinelli, and P. Zoccolotti, “Crowding, reading, and developmental dyslexia,” *Journal of Vision*, vol. 9, no. 4, 2009. [Online]. Available: http://www.journalofvision.org/content/9/4/14.abstract
[5] *Tobii Gaze SDK Developers Guide General Concepts*, Tobii, 2013.
[6] F. D. Luna, *Introduction to 3D Game Programming with Directx 9.0*. Plano, TX, USA: Wordware Publishing Inc., 2003.
[7] “Crazy eddie’s gui system mk-2: Developer documentation,” http://static.cegui.org.uk/docs/current/, Sept. 2013.
[8] “Mupdf documentation,” http://www.mupdf.com/docs/, Sept. 2013.
[9] D. G. R. Bradski and A. Kaehler, *Learning Opencv, 1st Edition*, 1st ed. O’Reilly Media, Inc., 2008.
[10] A. Williams and V. J. B. Escriba, “Boost: Chapter 31. thread 4.2.0,” http://www.boost.org/doc/libs/1_55_0/doc/html/thread.html, Sept. 2013.
[11] *Tobii Gaze SDK Developers Guide C API Specifics*, Tobii, Jan. 2013.
[12] P. Olsson, “Real-time and offline filters for eye tracking,” Ph.D. dissertation, KTH Royal Institute of Technology, 2007.
[13] G. Welch and G. Bishop, “An introduction to the kalman filter,” Chapel Hill, NC, USA, Tech. Rep., 1995.