Concentration Reminder: Distraction and Drowsiness Detection for Computer Users

Rui Wu **Lisa Palathingal** **Sergiu Dascalu** **Frederick C. Harris, Jr**
Department of Computer Science and Engineering
University of Nevada, Reno
{rui, lisa, dascalus, fredh}@cse.unr.edu

Abstract

Computers are used more and more in our daily lives, and it is normal that we focus on our monitors for a long time, such as when taking online courses. People may find it difficult to fully concentrate in front of computers because of distractions and drowsiness. Therefore, it is useful to create an application that alerts users to concentrate on their work. Some of the existing applications detect and attempt to avoid these distractions by cutting off several distracting elements, including blocking emails and restricting internet usage for a certain amount of time. However, these are not helpful when users do not focus on their computers or fall asleep. The proposed system, Concentration Reminder, is a user-centric and intuitive software that detects distraction and drowsiness in an efficient way. For detection of distraction, we have created an algorithm in order to check if the user looks anywhere outside the boundary of the monitor. The Haar Cascade classifier is used for checking if the user's eyes are closed or not (drowsiness). In both cases, Concentration Reminder alerts the user to stay focused on his or her work by providing a pop-up warning window and a beep message.

Keywords: Distraction detection, drowsiness detection, concentration reminder, image processing

1. Introduction

People watch online videos in their everyday life. Some users watch movies whereas others, like students, watch online tutorial videos and so on. It would be difficult for any user to stay focused on a lengthy video after a whole day of work or class. And it would make him or her distracted or drowsy thereby drawing away his or her attention from the video. These distractions and drowsiness may cause the user to miss part of a video or tutorial and he or she may find it difficult to search for the beginning of the missed part.

Our main aim is to help users focus on their computers, especially for online courses. One of the reasons that online courses sometimes are not as effective as traditional classes, is distraction [1]. In case of distractions and drowsiness, our software will display a pop-up warning window and a beep message, respectively. We have created our own algorithm to check if the user looks anywhere outside the boundary of the monitor (distraction). For drowsiness, we use the Haar Cascade classifier to check if the user's eyes are closed or not.

The proposed application consists of two parts: distraction detection and drowsiness detection. For distraction detection, we use Kinect, which is a powerful yet popular 3D camera, and utilize a free developer toolkit offered by Microsoft [2]. Kinect is used to measure the exact distance between the monitor and the user's face, which is hard for traditional 2D cameras. For drowsiness detection, we use OpenCV (Open Source Computer Vision). This is a cross-platform library of programming functions mainly aimed at supporting real-time computer vision [3]. Eye detection is the first step towards determining the closing and opening of users' eyes. For eye detection we use Haar Cascades, a classifier trained for detecting objects in a video stream [4].

Some applications also expect users to concentrate on the computer, however most of them are not effective in limiting distractions. Some examples, such as SelfControl [5] and Eliminate Distractions [6], try to avoid the impact of email messages and website notifications by blocking them. Unfortunately, they can do nothing when users leave their computers and do something else. Distraction, a part of our application can warn users, as long as they look outside of the boundary of monitors. DriveSafe, shown in Figure 1, is an application on Google Glass used for the detection of driver's drowsiness [7]. Although the application is useful, it is particularly intended for drivers. The proposed system is an alert system, which can be

978-1-880843-98-7 / copyright ISCA, CATA 2015
March 9-11, 2015, Honolulu, Hawaii, USA

used by any users, and provides a warning message to the users in case they fall asleep.

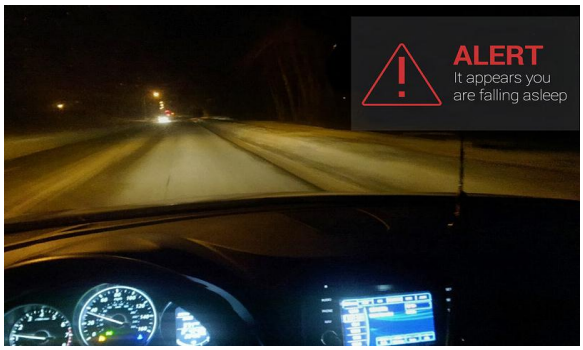

Figure 1.DriveSafe [13]

There are some applications that detect a user's drowsiness while driving [8, 9]. In these existing studies, drowsiness is determined by estimating the drowsiness degree from the driver's facial expression. The theories mostly used in these papers are based on pattern recognition. The main idea is to extract features of distraction and drowsiness; if the users have the features of distraction or drowsiness, the system will create feedback to the users. We decided to use a different method, because watching videos is not the same with driving. Also, we wanted to explore some ideas that are easier to implement and fitted more to our current research status.

The paper, in its remaining part, is organized as follows: Section 2 provides an overview of the proposed Concentration Reminder system, Section 3 presents Concentration Reminder use cases and current functionalities, Section 4 contains a comparison with related work, and Section 5 presents the conclusions.

2. Proposed system

2.1 System-level Introduction

Our system has two main parts: *distraction detection* and *drowsiness detection*. Distraction and drowsiness are two major symptoms when users watch online videos. Therefore, we designed the system to detect them. The system-level diagram is represented in figure 2.

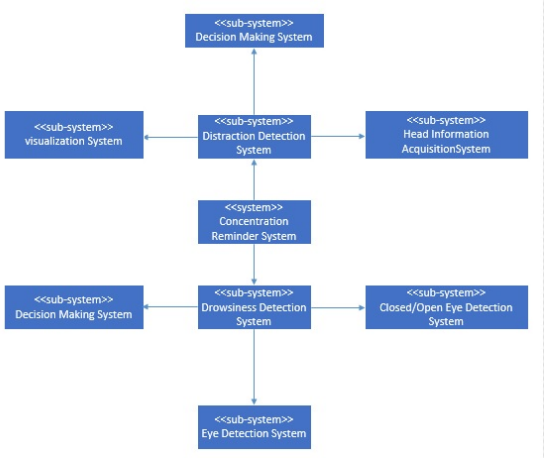

Figure 2. Concentration Reminder: System-level Diagram

2.2 Distraction Detection

In the distraction detection system, we have three subsystems: (i) the Visualization System, (ii) the Head Information Acquisition System, and (iii) the Decision Making System.

The Visualization System deals with displaying information on the window. The information contains head translation, head rotation, distance between monitor and head, and the user's status (distracted or not).

The Head Information Acquisition System needs to get data about the user's head. Kinect could just get head translation and rotation according to the coordinates system, whose origin is the Kinect optical center. The x axis is the right direction of Kinect (when Kinect faces users), the y axis is the up direction of Kinect, and the z axis refers to the direction to the users. Because the position and elevation angle of Kinect are not fixed, we need to modify this data to make them useful. The modification method is that we initialize our program with the user's data in the first frame, named T_{1x}, T_{1y}, T_{1z} and R_{1x}, R_{1y}, R_{1z}, (the user should keep his or her position in the middle of the monitor in the first few seconds), and for other frames such as frame N, we get the data $T_{Nx} = T_x - T_{1x}$, $T_{Ny} = T_y - T_{1y}$ and $T_{Nz} = T_z * \cos R_{1x}$ and $R_{Nx} = R_x - R_{1x}$, $R_{Ny} = R_y - R_{1y}$, and $R_{Nz} = R_z - R_{1z}$, where T and R are the translation and rotation data obtained from Kinect's frame N.

The Decision Making System determines whether users are distracted in each frame. We assume that when users concentrate on something their eyes' direction should be

vertical to the face, which means we can use head rotation to represent eye rotation. The principle is that the application extracts data from each frame and calculates the focus point of a user's eyes. If the focus point is within the boundary of the monitor, then the user is not distracted. We separate the eye focus point into vertical and horizontal directions. Figure 2 shows the details about how to calculate the eye focus point. P is the center of the monitor, D3 is T_{Nx}, D1 is T_{Nz}, θ is R_{Nx}, and D2 = D1 * tan θ. If D3 + D2 is longer than the half length of the monitor, then it means the horizontal eye focus point is outside of the monitor boundary. Similarly, D5 is T_{Ny}, D4 is T_{Nz}, θ is R_{Ny}, and D6 = D4 * tan θ. If D5 + D6 is longer than half height of the monitor, then it means the vertical eye focus point is outside of the monitor boundary. If either of these two eye focus points are not within the monitor boundary, the application will pop up with a warning message.

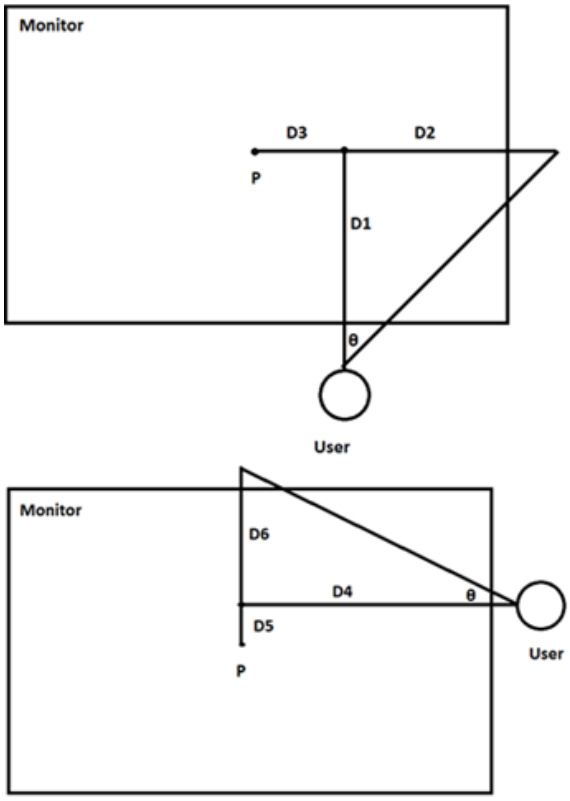

Figure 2. Horizontal Distraction Detection

2.3 Drowsiness Detection

Drowsiness detection includes three subsystems: (i) the Eye Detection System, (ii) the Closed/Open Detection System, and (iii) the Decision Making System.

As the name implies, the Eye Detection System is responsible for detecting and classifying the user's eyes from the video frame. For this, video frames have to be captured and the system has to identify the location of the user's eyes in the frame. OpenCV already contains many pre-trained classifiers for face, eyes, smile, and other. The Concentration Reminder uses a Haar Cascade classifier [10] for detection of eyes from a video frame.

Once the eyes are detected, the next step is to determine closed or open eyes (Closed/Open Detection System). In the system, a fixed number of frames from the video are considered at a time. To track the history of the frames, an array of integers is initialized to 0.

From the Eye Detection System, the number of open eyes is placed in an "eye vector". The size of the eye vector is automatically determined by the Vector type, a standard library included in C++ by default. Eye vector's size include positive integers. If eyes are detected from a frame, size of eye vector is incremented automatically. If the size of the eye vector is 0, no eyes are detected (i.e. "closed" eyes are detected), and the corresponding history array cell is replaced by the value 1. A history array cell keeps the value 0 if any open eyes are detected in a frame. After a fixed number of frames are processed, the system tallies the total number of 1s in the history array – the number of frames with closed eyes. A flowchart explaing the procedure of drowsiness detection is shown in Figure 3.

If the counter of closed eyes is greater than the specified threshold value (for example, 30 out of last 100 frames), the system detects that the user is sleeping and alerts the user with a warning message. If the counter is less than the threshold value, the system detects that the user is active. In either case the process continues for the next set of frames from the video. These detections are handled by the Decision Making System.

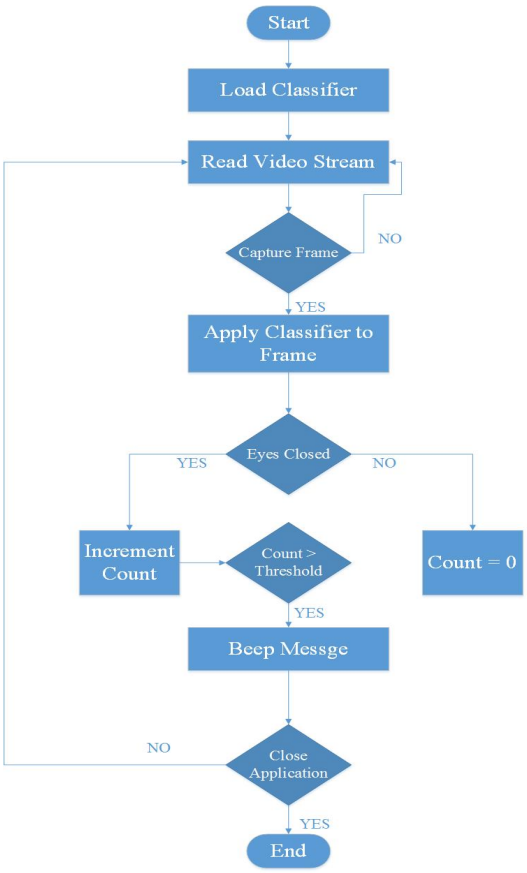

Figure 3. Flowchart of Drowsiness Detection

The system considers eye blinking and drowsiness as two separate cases. In the eye blinking case, a person's closed eyes will be detected in a small number of successive frames in the video. This is distinct from a sleeping person, whose eyes would be closed for a large number of successive frames. In case of a blink, the closed eyes counter is reset to 0 and the process continues.

3. Software System and Results

We designed our application based on the use cases represented in Figure 4. Camera offers the video information, and our application processes the information from each frame. The users can view their head and eyes status from our application. If they are distracted or drowsy, the application will send out a warning message.

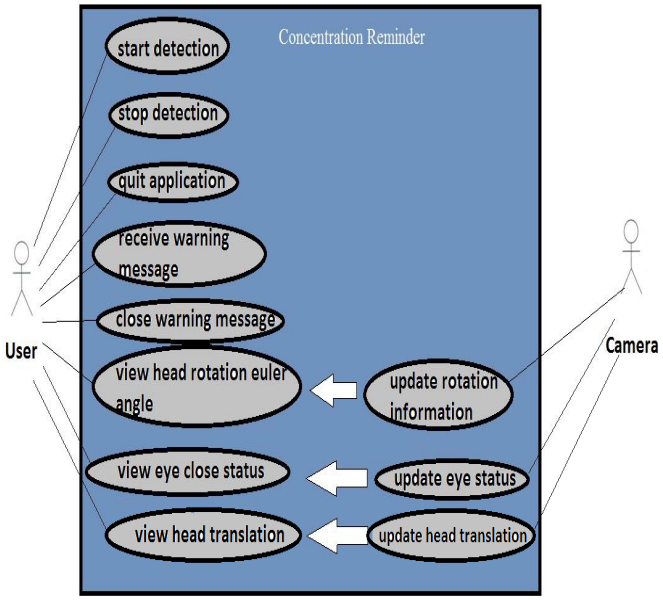

Figure 4. Use Case Diagram

Figure 5 presents a screenshot of distraction detection part of our application. The users' head information is shown in the left-hand part of the window while the video information is displayed in the right-hand part of the window. The users should place their head in the middle of the screen for the first few seconds to initialize the program. And it does not matter where the Kinect is placed, as long as it can detect the users' face. Our program references parts of the code from the Kinect official website [12], and the Kinect will trace the face and represent the user's face in the middle of the window.

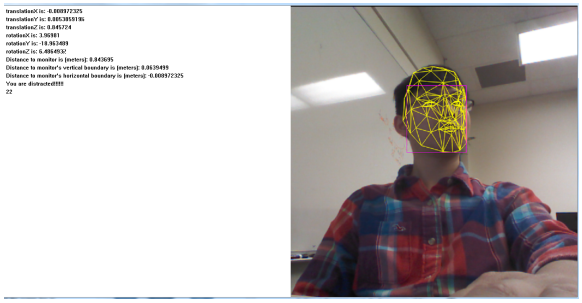

Figure 5. Distraction Detection Screenshot

Figure 6 and Figure 7 show screenshots of the drowsiness detection part of our application. The right-hand part of the screenshots tracks the user using a 2D camera. The history array cell values are updated based on the status of the user's eyes (closed or open) in each frame. The information about array cell values is shown in the left-hand part of the window. If the count of array cells with value '0' is greater than the specified threshold

value, the system detects that the user is sleeping and alerts the user with a warning message, as shown in Figure 6. If the count is less than the threshold value, the system detects that the user is active and alerts the user with a warning message, as shown in Figure 7. In either case, the process continues for the next set of frames from the video. Our program references parts of the code from the OpenCV documentation website [15].

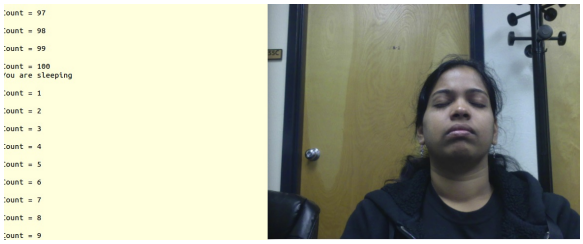

Figure 6. System Detecting that the User is Sleeping

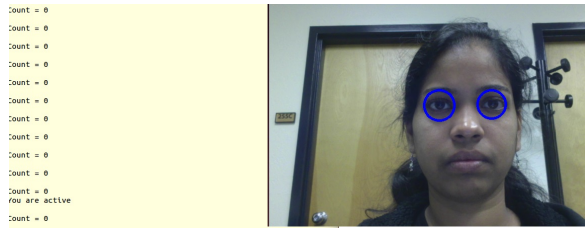

Figure 7. System Detecting that the User is Active

4. Comparison with Related Work

An innovative feature available in Samsung Galaxy 4 is called Smart Pause [11], which pauses a video being watched by a user when he or she looks away from the screen. In order to detect if the user is looking at the screen or not, the front camera of the mobile is used. Although the feature is appealing to many Samsung Galaxy users, it may not work in case of failure of face or eye detection by the front camera, for example if the light source is behind the user. Concentration Reminder does not include a 'pause' feature. We are planning to extend our project by adding such a feature in the future.

SelfControl [5] and Eliminate Distractions [6] are two similar applications. They help users to concentrate on their work by blocking possible factors that could make the users distracted. These factors include emails, Facebook, Twitter, and so on. Users can set a timer to control the blocking time. SelfControl and Eliminate Distractions try to avoid users' distractions from the factors inside the computers, regardless of factors outside the computers. For example, users may be away from their keyboards, or they may fall asleep. Concentration Reminder tries to handle these factors outside computers – distraction and drowsiness.

An application offered by Google Glass for detection of driver drowsiness is DriverSafe. When the driver falls asleep, the application uses the eyewear's built-in sensors for detection and alerts the driver through Google Glass' bone conduction speaker [7, 13]. DriveSafe is activated using a command by the driver asking Google Glass to keep him or her awake. Concentration Reminder does not require the users to provide any audible messages. Once the users open the application, the application starts tracking the users' face through the camera.

Overall, Concentration Reminder focuses on outside elements and can accurately detect distraction. It can also remind the users by sending out warning messages. SelfControl and Eliminate Distraction focus on inside elements, such as the computer network. DriverSafe is designed for drivers, not computer users.

5. Conclusion and Future Work

Concentration Reminder is an application that can detect users' distraction and/or drowsiness in real time. It can pop up a warning message to remind users to concentrate on their work. Our application keeps users concentration on their work by avoiding outside factors, distraction and drowsiness, regardless of inside factors such as emails, apps, and so on. This application could be used for different purposes, like during taking online courses and self-study. To the best of our knowledge, there are few applications or papers about detection of distraction and drowsiness for computer users. That was our main motivation to create the Concentration Reminder System.

Because Kinect is not prevalent, in the future we plan to replace Kinect with a web-camera in the distraction part of our application. The challenge is to measure the distance between the monitor and the user's face exactly. We found a valuable paper introducing this technique [14], and we plan to try the method in our program in the future. We also would like to add a "stop" function to our application, which could stop the ongoing video tutorial animation when the application detects distraction or drowsiness.

Acknowledgment and Disclaimer

This material is based in part upon work supported by the National Science Foundation under grant number IIA-1301726.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] COTLER, J., KASSAB, D., AND YUAN, X. 2013. Under what conditions does web conferencing inhibit learning in a computer science classroom? In *Journal of Computing Sciences in Colleges*, vol. 28, Issue 6, 179-185.

[2] Kinect Introduction (accessed November 11, 2014), http://en.wikipedia.org/wiki/Kinect

[3] OpenCV Introduction (accessed November 11, 2014), http://en.wikipedia.org/wiki/OpenCV

[4] CORREIA, J., MACHADO, P., AND ROMERO, J. 2012. Improving Haar Cascade classifiers through the synthesis of new training examples, In *Proceedings of the 14th Annual Conf. Companion on Genetic and Evolutionary Computation (GECCO '12)*, 1479-1480.

[5] Self Control (accessed November 11, 2014), http://visitsteve.com/made/selfcontrol/

[6] Concentrate—Eliminate distractions to work and study more productively (accessed November 11, 2014), http://www.getconcentrating.com/

[7] DriveSafe (accessed November 11, 2014), http://www.drivesafeforglass.com/

[8] AHLSTRÖM, C. AND KIRCHER, K. 2014. Review of real-time visual driver distraction detection algorithms. In *Proceedings of the 7th International Conference on Methods and Techniques in Behavioral Research*, No. 2.

[9] NAKAMURA, T., MATSUDA, T., MAEJIMA, A., AND MORISHIMA, S. 2013. Driver drowsiness estimation using facial wrinkle feature. In *Proceedings of ACM SIGGRAPH 2013 Posters*, No. 30.

[10] CHENG, D., AN, M., YU, W., AND FANG, L. 2014. Learning Dynamic Models of Gaze Point Mapping from Eye Movement, In *Proceedings of International Conf. on Internet Multimedia Computing and Service (ICIMCS '14),* 289.

[11] SmartPause (accessed November 19, 2014), http://www.samsung.com/us/support/faq/FAQ00053263/62636/SPH-L720ZPASPR

[12] Face Tracking Visualization C++ sample (accessed November 19, 2014), http://msdn.microsoft.com/en-us/library/jj131045

[13] DriveSafe (accessed November 19, 2014), http://www.gizmag.com/drivesafe-app-google-glass/30465/

[14] RAHMAN, K. A., HOSSAIN, M. S., BHUIYAN, M. A., ZHANG, T., HASA-NUZZAMAN, M., AND UENO, H. 2009. Person to Camera Distance Measurement Based on Eye-Distance. In Multimedia and Ubiquitous Engineering. In *Proceedings of the 2009 Third International Conference on Multimedia and Ubiquitous Engineering (MUE '09)*, 137-141.

[15] Cascade Classifier (accessed Nov. 23, 2014), http://docs.opencv.org/doc/tutorials/objdetect/cascade_classifier/cascade_classifier.html

[16] SATORI HACHISUKA, KENJI ISHIDA, TAKESHI ENYA, MASAYOSHI KAMIJO, 2011. Facial Expression Measurement for Detecting Driver Drowsiness. In *Proceedings of the 9th International Conference on Engineering Psychology and Cognitive Ergonomics (EPCE '11),* 135-144.