https://dx.doi.org/10.1016/j.engappai.2023.106495

Is Your Mouse Attracted by Your Eyes: Non-intrusive Stress Detection in Off-the-shelf Desktop Environments

Jun Wang[a], Chunxi Yang[b], Eugene Yujun Fu[b,*], Grace Ngai[a,c], Hong Va Leong[a]
[a]Department of Computing, [b]Department of Rehabilitation Sciences, [c]Service-Learning and Leadership Office
The Hong Kong Polytechnic University, Hong Kong, China

ABSTRACT

Increasing number of people work long hours with computers under high cognitive load. This could potentially cause mental stress in workplaces. Prolonged exposure to mental stress contributes to poor working experience and even severe health problems. Despite the growing demand, the existing intelligent stress detection methods are limited when applied to actual workplaces. They often measure physiological and physical signals, via intrusive devices, to detect stress. The intrusiveness hampers their accessibility and applicability in daily life and workplaces. To overcome that, behavior-based methods were proposed. Models that explore mouse and gaze behaviors during computer usages were demonstrated to be particularly effective. However, the current methods rely on using prior knowledge of the user interface (UI) layout to construct models. Their applicability thus is limited, especially in real workplaces where task UI is often dynamic. This paper presents a novel stress detection method to address the challenges. It attains non-intrusiveness and UI-agnostic by modeling the relative movement and coordination of mouse and gaze. The method is evaluated on a dynamic-UI task, namely, web searching. An accuracy of 78.8% is achieved using a commercial eye-tracker for gaze estimation, beating the state-of-the-art approaches by around 20%. We further use webcam to estimate gaze locations substituting for the eye-tracker, to enhance the model accessibility. The method yields 68.6% accuracy of stress detection without using any special devices. Experimental results demonstrate the effectiveness and applicability of our method. It opens up a new avenue for cognitive-aware adaptive user interface, intelligent working environment, and related applications.

Keywords: stress detection, gaze-mouse correlation, human factors, machine learning, intelligent system

1 INTRODUCTION

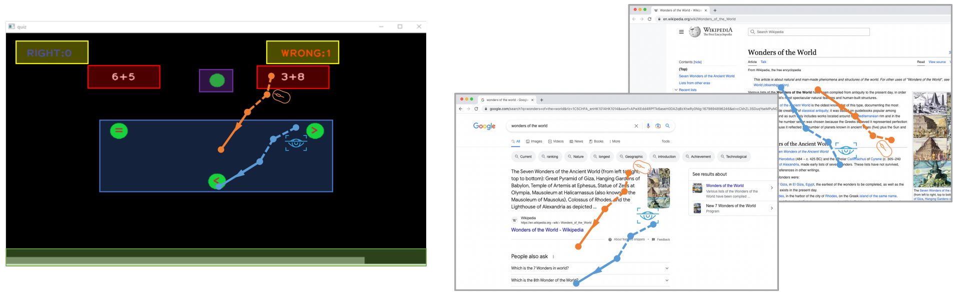
Mental stress occurs when a human feels threatened or frustrated by an external event or activity, i.e., a stressor [18]. The nervous system unconsciously responds to the stressor by releasing a flood of hormones to prepare for the anticipated challenges [40]. As human-computer interaction (HCI) grows inexorably in daily work/study environments, people work with multimedia interfaces more and more often. The processing of high volumes of information over long hours can easily result in high cognitive load, and therefore becomes the stressor that causes mental stress in workplaces [23, 29]. Chronic exposure to mental stress contributes to many health problems, such as high blood pressure, heart disease, obesity, diabetes [39], and leads to various cognitive and emotional symptoms [30]. In the context of working with computers, high-level stress may induce frustration in HCI, reduce the working effectiveness, and affect the working experience. There is a significant demand for intelligent systems that can identify workplace mental stress for computer

[*] Corresponding author: Eugene Yujun Fu, Department of Rehabilitation Sciences, The Hong Kong Polytechnic University, Hong Kong Special Administrative Region, China. Email: eugene.fu@polyu.edu.hk

users. With such systems in place, intelligent adaptive user interface can be developed to enhance users' working experience. For example, one can be reminded to take a break or adjust the work load when high stress level is detected while working with computers. This can be further augmented with referral and intervention support for potential clinical services [2].

However, conventional stress detection methods are limited when applied to actual workplaces. Most of them rely on the processing of human physiological signals and physical information, including electrodermal activity (EDA) signals, heart rate and heart rate variability (HRV), blood activity and pupil dilation [13, 21, 43, 46]. Although these approaches yield encouraging performance, they require intrusive devices (e.g., chest belt) to access accurate bio-signal, which makes them impractical in daily life and workplaces. Literature has pointed out that the devices and intrusiveness can already be the stressors themselves [18], affecting user experience in daily life. The inconvenience of deploying such devices in daily work environments further hampers their accessibility and applicability.

To achieve non-intrusive stress detection, efforts have been taken to explore behavior-based methods that work in common computer environments. Among them, modeling user behaviors while interacting with computers was demonstrated to be particularly effective [8, 26, 31, 42, 47, 48]. In traditional keyboard-video-mouse (KVM) interaction model, hand and gaze commonly dominate in human-computer interaction [16]. Their movements thus are often exploited to infer mental stress for computer users [26, 42, 48]. However, the few available approaches are limited by using prior knowledge of the user interface (UI). They thus work only on static UI tasks. For example, Wang et al. [48] modeled mouse and gaze movements within/between some specific unalterable UI components (e.g., particular buttons), to detect stress in mental math calculation task with a fixed UI (Fig. 1 *a*). Their method applied only to that specific UI layout. However, most of the computer tasks in actual workplaces are carried out with dynamic-UIs. One particular example is the task of web searching, in which the UI layout changes from page to page (Fig. 1 *b*). To the best of our knowledge, the existing behavior-based stress detection methods perform poorly when deployed to dynamic-UI environments. Their applicability thus are constrained when applied to real workplaces. A UI-agnostic stress detection method would be highly valuable to practical workplace applications.


RIGHT:0
WRONG:1
6+5
3+8
=
<
>
Wonders of the World - Wikipedia
en.wikipedia.org/wiki/Wonders_of_the_World
Wikipedia The Free Encyclopedia
Search Wikipedia
Wonders of the World
Contents [hide]
(Top)
Seven Wonders of the Ancient World
Lists from other eras
Recent lists
Article Talk
Read View source
From Wikipedia, the free encyclopedia
wonders of the world - Google
Google
wonders of the world
All Images Videos News Books More Tools
Current ranking Nature longest Geographic introduction Achievement Technological
The Seven Wonders of the Ancient World (from left to right, top to bottom): Great Pyramid of Giza, Hanging Gardens of Babylon, Temple of Artemis at Ephesus, Statue of Zeus at Olympia, Mausoleum at Halicarnassus (also known as the Mausoleum of Mausolus), Colossus of Rhodes, and the Lighthouse of Alexandria as depicted ...
Wikipedia
Wonders of the World - Wikipedia
See results about
Wonders of the World
New 7 Wonders of the World
People also ask
Which is the 7 Wonders in world?
Which is the 8th Wonder of the World?

(a) (b)

Fig. 1. Example static-UI and dynamic-UI tasks. (a) shows a mental math calculation task with a fixed UI, which holds some unalterable components such as equation display areas and selection buttons. (b) depicts a web search task, in which the UI layout, such as the text/image area, changes from page to page.

To address the gaps, this study presents an innovative method for non-intrusive and UI-agnostic stress detection. It models mouse and gaze behaviors that can be captured non-intrusively with personal computers. Moreover, unlike the previous studies that treat these two modalities separately and construct models based on their movements within specific UI components, our method extracts information on their coordination – the relative movements – regardless of the UI layout. Specifically, we propose the MGAttraction model that measures the "*attraction*" between mouse and gaze to formulate their relative movements. Inspired by the law of gravitational attraction, we design the MGAttraction to be positively related to the relative moving speed of mouse and gaze, and negatively related to their distance. A MGAttraction is strong if mouse and gaze are moving fast toward each other and/or they are close to each other. This novel measurement considers both relative speed, position and moving direction between mouse and gaze. Its coordinate system also has rotation- and translation-invariant characteristics. This means that the MGAttraction model can be completely agnostic to the UI layout and invariant to the location and movement direction of the mouse and gaze. Our stress detection method is constructed upon the MGAttraction model, which also makes it UI-agnostic.

Our method is evaluated with human subject experiments "in the wild", on a dynamic-UI task, namely web search. Stress was induced by imposing a time limit and adding background noise in the tasks. Commercial eye trackers, such as Tobii EyeX, can provide accurate estimation of gaze positions, thus are helpful to our method. Its accuracy reaches 78.8%, about 20% higher than the state-of-the-art methods, when applying Tobii EyeX for gaze estimation. Nonetheless, the dependence on external devices constrains the accessibility and applicability of the model. To tackle that, we explore the use of webcam-based approach, instead of eye trackers, to estimate the gaze positions for our stress detection. Our method achieves 68.6% accuracy of stress detection without using any external devices.

The contribution of this paper is threefold. (1) The proposed MGAttraction offers a novel way to model the relative movement between mouse and gaze in a UI-agnostic manner, addressing the limitation of the UI-dependent methods. (2) The proposed MGAttraction-based stress detection model works well in dynamic-UI environments, outperforming the state-of-the-art methods. (3) It still achieves promising performance even in off-the-shelf desktop environments without using external devices. We believe that this work will pave the way to developing practical applications for intelligent stress detection and working environment monitoring.

2 RELATED WORK

Mental stress often occurs in workplaces, when people process multimedia information under high cognitive load [23, 29]. As such, it is beneficial to explore automatic stress detection method that is applicable in actual workplaces. This section starts with the literature review of conventional stress detection methods, including intrusive approaches that exploit physiological and physical signals, and non-intrusive approaches that use behavioral signals. A discussion of the gap to practical workplace stress detection is also presented. This is followed by the review of hand and gaze coordination studies.

2.1 Intrusive Stress Detection Approaches

Most of the conventional methods detect stress based on the analysis of physiological signals and physical information. The most prevalent input modalities include electrodermal activity signals, heart activity, blood activity, and pupil dilation. They are often modeled jointly to achieve better performance.

Healey et al. [21] detected stress during a driving task by continuously processing electrocardiogram, electromyogram, skin conductance and respiration signals taken over 5-minute windows. Statistical features, spectral power features and features to characterize orienting responses were extracted to construct the model, which achieves over 97% accuracy in real-world driving tasks. Wagner et al. [46] exploit the same physiologic modalities to infer four classes of emotion: joy, anger, sadness and pleasure, which is triggered by music. They explore a variety of feature selection methods and feature reduction methods with different machine learning models. They achieve up to 92% accuracy, which is around 12% improvement compared with the performance achieved without using any feature selection or reduction methods. Sun et al. [43] present an activity-aware mental stress detection based on electrocardiogram, galvanic skin response, and accelerometer signals. Stress in their study is induced by mental arithmetic tasks with time limit pressure, while subjects are in different activities: sitting, standing and walking. Their method obtains 80.9% accuracy without requiring the controlled laboratory setting. In their work, they also point out that the accelerometer signal is essential in stress detection to help determine the conditions behind different physical activities, which has a strong impact on spectrum features of physiological signals. Sierra et al. [13] propose a fuzzy expert system to determine an individual's stress level by analyzing galvanic skin response and heart rate signals. In their experiment, stress is induced through hyperventilation and talk preparation. Their system achieves over 90% of accuracy for 3-5 seconds signals acquisition period and 99.5% accuracy for the 10-second period.

Barreto et al. [4] and Ren et al. [36] showed that the pupil diameter, which is controlled by the autonomic nervous system, provided a strong indication of stress. They show that stress recognition performance can be improved dramatically when pupil dilation is incorporated. It is obvious that most of the stress detection methods that involve physiological signals can achieve decent performance around 90% accuracy. However, one major drawback always inherited by these methods is that they are intrusive and require special equipment attached to the subjects. Literature has pointed out that the intrusive devices themselves can already be the stressors, which may generate psychological side effects to the users. Furthermore, it is inconvenient to deploy such devices (e.g., chest belt) in daily work environments. All of these make the intrusive approaches impractical in actual workplaces.

2.2 Non-intrusive Stress Detection Approaches

Non-intrusive stress detection methods mainly use facial expressions as one of the prevailing modalities. Abouelenien et al. [1] detect acute stress through thermal imaging. They extract thermal features from thermal facial images of each subject. Thermal features are used to describe the distribution of colors in the Hue Saturation Value space. Their experiment results illustrate that by fusing with thermal features, the relative accuracy can improve 26.6% performance over the heart rate and skin conductance features and 38.2% performance over the respiration rate features. Despite that, the method is still limited by using special equipment to capture thermal images. Bosch et al. [6] detect affective states, including boredom, confusion, delight, engagement, frustration while subjects are interacting with an educational game. The facial action unit and head pose are extracted from the video as features, without external devices. The method achieves an accuracy of 65% for the overall classification of affect. Viegas et al. [44] build a dataset containing 114 different subjects. Each subject undergoes three 15-minute typing phases: before the stressor, after stressor and after relaxation. The stressor is a multitasking exercise with social evaluation. They extract 18 facial action units (AUs) and build a random forest classifier to determine different phases from videos. They obtained an average accuracy of over 97% and 50% accuracy for the subject-dependent and independent models, respectively. Nonetheless, individuals may have different ways to express emotions, such as stress. Their culture, workplace social norms, and personality all play an important role. This makes the stress-related facial expression vary from person to person. Facial expression-based stress detection thus often generalize poorly across different users in practical applications. Additionally, the limited range of facial expressions captured by AUs may result in the loss of valuable information [7], making it even less effective in real-life applications.

There is a growing trend of directly extracting features from human behavior patterns for stress and emotion analysis, which are shown to be more user-independent. Given that, different behavior indicators have been investigated as indicators for stress detection. Haak et al. [20] observe that when a human is under stress, he/she tends to show a higher frequency of eye blinks. Hernandez et al. [22] find that under the stress conditions, most subjects (>79%) in their experiments consistently type with more force and pressure and click the mouse with a greater amount of mouse contact. Ciman et al. [10] measure the differences in smartphone interaction between a relaxed and stressed state of users. In their study, they mainly investigate four kinds of smartphone interactions, including "scroll", "swipe", "touch" and "text input". Where scroll, swipe and text input behaviors can be utilized for stress classification. Paredes et al. [35] model the human arm while driving by using the mass spring damper (MSD) model and find that when people are driving under stress, the muscle tension of the arm is significantly higher compared to the calm state. A similar finding is also found by Sun et al. [42]. They apply the MSD model to the arm while holding the mouse. After analyzing the mouse trajectories for "point-and-click", "drag-and-drop" and "steering" mouse operations, they suggest that the arm muscle is stiff under the stress condition.

Two studies closest to our work are StressClick [26] and Wang et al. [48]. StressClick detects stress based on the gaze behaviors around each mouse click. They illustrate that when a subject clicks a target, the closest fixation duration preceding/during a click and the reaction latency after a click is negatively correlated to whether under stress condition, since under stress condition, a subject tends to conduct operations more rapidly. A stress detection system is constructed based on the findings, which achieves 74.0% accuracy. However, StressClick is only evaluated under a static UI environment, and may not be effective in dynamic-UI environments. Wang et al. [48] discover that gaze attention sequence on UI components is more consistent under the stress condition. Nevertheless, their work also relies on UI information to formulate gaze movements. Compared with these works, our method in this paper studies the relative movement of mouse and gaze without relying on UI information. This makes our approach suitable for stress detection under dynamic-UI environment, which is more flexible and practical.

Table 1 summarizes the existing stress detection methods and the gap to practical workplace stress detection. In summary, we anticipate a practical workplace stress detection method to be non-intrusive, accessible in off-the-shelf computer environments, applicable in working tasks with dynamic-UI, and user-independent. To the best of our knowledge, this is the very first study to explore such a method.

We are also observing a trend of training end-to-end models with deep learning techniques. In many applications, deep learning models outperform their counterparts that trained with hand-crafted features. However, well-performing deep learning models usually require to be trained with a substantial amount of data. To our best knowledge, there are only a few available deep learning models in the domain of stress detection, due to the scarcity of data. Li et al. [32] trained a 1-D CNN model for physiological signal-based stress detection, boosting the performance of binary stress classification to an accuracy of 99.8%. Nonetheless, their model was trained on an existing large-scale dataset, collected with intrusive devices and under restricted laboratory conditions [38]. Dahmane et al. [12] applied long short-term memory (LSTM) and sequential convolutional neural network (CNN) to detect stress and other emotions using sequential multi-modal data (i.e., facial and audio data). The model was trained and evaluated on a public dataset collected for emotion detection in emergency calling [14]. Despite the encouraging performance it achieved, the model cannot be applied for stress detection in common workplace and desktop environments. Practical workplace stress detection is a novel application problem. There is no existing large-scale dataset in this domain. It is generally difficult, and may take longer time, to collect such a dataset, which involves a large number of real human subjects in real human-computer interaction tasks. This

Table 1. Summary of the Existing Stress Detection Methods and the Gap

Method	Modalities	Classification Model	Non-Intrusive	Off-The-Shelf	Workplace Task	Dynamic UI	User-Independent
Wagner et al. [46]	Physiological signals	KNN, MLP	×	×	×	-	×
Sun et al. [43]	Physiological signals	SVM	×	×	×	-	✓
Sierra et al. [13]	Physiological signals	FIS	×	×	×	-	✓
Barreto et al. [4]	Physiological signals	NB, DT,SVM	×	×	×	-	✓
Bosch et al. [6]	Facial Expressions	NB,LR	✓	✓	×	-	✓
Viegas et al. [44]	Facial AU	RF	✓	✓	✓	×	×
Abouelenien et al. [1]	Physiological signals	DT	✓	×	×	-	✓
Dahmane et al. [12]	Facial AU and auditory	LSTM	✓	✓	×	-	×
Li et al. [31]	Physiological signals	CNN	×	×	×	-	✓
Wang et al. [48]	Mouse and gaze	SVM	✓	×	✓	×	✓
StressClick [26]	Mouse and gaze	SVM	✓	✓	✓	×	✓
Ours	Mouse and gaze	SVM	✓	✓	✓	✓	✓

Note: KNN: K-nearest Neighbors, MLP: Multi-layer Perceptron, SVM: Support-vector Machine, NB: Naive Bayes, DT: Decision Tree, LR: Linear Regression, RF: Random Forest

limits the feasibility of training an end-to-end model with deep learning for workplace stress detection. We thus focus on engineering hand-crafted features in this study.

2.3 Hand and Gaze coordination

To address the research gap, this study investigate on methods that only exploit modalities we could easily access in daily desktop environments with common HCI tasks, which include the signals captured from a computer's camera, keyboard, and mouse. Our examination of relevant studies revealed that modeling the hand and gaze coordination is one of the efficient ways in this direction. Human often needs to cooperate hands and eyes to accomplish a given task. In particular, human eyes receive information from the surroundings to control, guide, and direct the actions of hands [11]. The behavioral relationship between hands and eyes (e.g., the coordinated hand and eye movements) in such a process, is known as hand-eye coordination. In daily computer interaction tasks, hand-gaze coordination usually refers to the coordination between mouse/keyboard activity and gaze (e.g., cursor-gaze/type-gaze coordination).

There has been some work in this area. Inhoff et al. [27] studied based on 19,000 observations of type-gaze coordination and point out that gaze is usually four characters to the right of the typed character in copy-typing task. This relationship may be affected by the success of perceptual processes and typing skill. For the gaze-cursor coordination, Bieget et al. [5] find that in search and selection tasks, there exist two main gaze and cursor coordination strategies, which appear to serve different scenarios. First, if a subject wants to select a target, whose approximate location is known, he/she moves the mouse directly to the target without gaze guidance. Second, if the approximate location of the target is unknown, then he/she parallelizes search and pointer movements to minimize the amplitude of the acquisition movement. Rodden et al. [37] analyze the cursor-gaze coordination on web search result pages. They report that mouse movements are usually for the purpose of clicking, but when the mouse movements are "following the eye horizontally and "highlighting a particular result", this indicates that a user is processing the content. Liebling et al. [33] investigate the gaze and cursor coordination in realistic task settings and they illustrate that gaze leads the mouse click only about two-thirds of the time, which is affected by the type of target and familiarity with the application. Weill-Tessier et al. [49] extend the hand and gaze coordination to the tablet interaction and show that gaze leads the finger about 356 ms to the touching target and the distance between gaze and finger is around 159 pixels. Huang et al. [25] studied mouse-gaze and type-gaze distances in real-world interactive tasks, and concluded that the commonly-accepted principle of "the user is looking where he/she is clicking" is not necessarily true. Previous studies of hand and gaze coordination imply that the change of mouse and gaze coordination, especially the distance between mouse and gaze in both spatial and time domains, can infer whether a person is under the stress condition.

3 METHODOLOGY

The goal of the study is to investigate an automatic workplace stress detection method based on mouse and gaze coordination, which works in dynamic-UI environments. This section first describes the method by which we formulate and measure mouse and gaze coordination in a UI-agnostic manner. This includes the procedure of data stream processing, coordinate system transformation, and attraction computation. We follow with details of how the measured mouse-gaze coordination can be analyzed to infer the overall stress level. Finally, we present the approach of building model with webcam-captured gaze positions.

3.1 MGAttraction: Modeling Mouse and Gaze Coordination

Most previous work modeled mouse and gaze movements (MGMovements) based on their locations in the screen, i.e., the $< x, y >$ coordinates in screen coordinate system (Fig. 2 b). However, these methods are generally not rotation- and translation-invariant, and easily fail in capturing the real behavior patterns behind the movements. For instance, Fig. 2 (a) illustrates two example cases of MGMovements, which differ greatly in the screen coordinate system as shown in Fig. 2 (b). Though they share the same behavior pattern that is “gaze leading mouse” to a potential target (e.g., a link). The variation caused by rotation and translation may potentially confuse the screen location-based methods.

We wish to model coordination that takes into account movement trends, such as "gaze leading mouse". It is difficult to model such behaviors in a UI-agnostic environment. In static UI environment, all the UI components including the potential clickable targets (e.g., buttons) are clearly defined and displayed. It is thus easy to determine the relative MGMovements to the potential targets, such as determining whether the gaze is closer to the target button than mouse. However, the dependence on prior knowledge of UI components make these methods less practical in real applications. To tackle the challenges, we propose an innovative measurement, namely "MGAttraction", to model the relative movement between mouse and gaze. Based on the MGAttraction measurement, we build our model to detect workplace stress detection in common computer environments.

3.1.1 Data Preprocessing and Coordinate System Transformation. The aim of MGAttraction is to model the relative movement between mouse and gaze in the dynamic-UI environment. As the name suggests, the MGAttraction measures the *attraction* between mouse and gaze. The attraction between the mouse and gaze, which is interpreted as the intensity and tendency of the relative movement of the mouse and gaze, is measured from consecutive samples in the mouse and gaze modality streams. The mouse modality stream is the sequence of on-screen coordinates of the mouse cursor, whereas the gaze modality stream is the sequence of on-screen coordinates of the

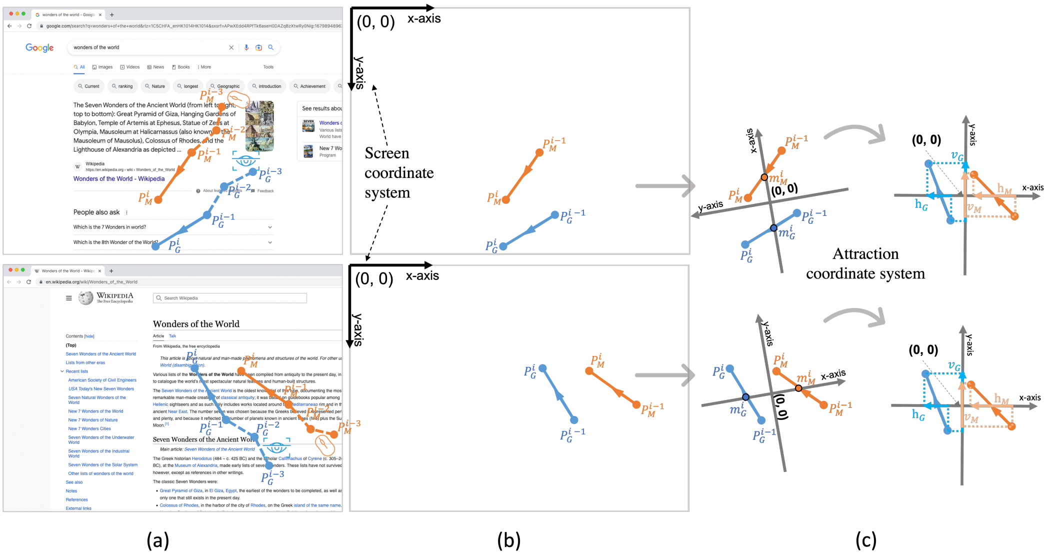


Fig. 2. The demonstration of the proposed attraction coordinate system and MGAttraction measurement. (a) shows two example cases of mouse (orange) and gaze (blue) movement trajectories, where solid and dashed lines denote present and past movements respectively. Both examples indicate the behavior of gaze leading mouse to a clickable link. Though their locations and movements differ greatly in screen coordinate system as shown in (b). (c) illustrates the proposed attraction coordinate system, in which the two examples have the same transformation. We measure the MGAttraction based on the mouse and gaze velocity decomposition (projections) on that system.

gaze attention location. The attraction is a signed scalar quantity. A large positive attraction of gaze relative to mouse implies that the gaze has a strong tendency to approach the mouse at this moment and a large negative value indicates a strong tendency for the gaze to depart from the mouse.

In our experimental setup, the sequence of on-screen mouse positions is recorded by a C++ program at 100 Hz and the sequence of on-screen gaze positions is captured by the eye-tracker Tobii EyeX at 60 Hz and webcam-based detector at 30 Hz respectively. Tobii eye tracker has its coordinate system that is applicable to any resolution. Specifically, the gaze coordinates (i.e., x, y) returned by a Tobii eye tracker are normalized to $[0, 1]$. The point (0, 0) and (1, 1) denote the upper left and lower right conners of the screen respectively. We apply the same rule to normalize the coordinates for the captured mouse cursors and gaze points estimated from webcam video before computing the proposed MGAttraction, to ensure all of the modalities are in the same coordinate system. The normalized mouse and gaze coordinates denote their relative locations at the screen. This ensures the consistency of the feature engineering across different resolutions.

We then apply a two-phase heuristic filter [41] to remove the impulse noise from the gaze signal to remove artifacts from eye blinks. We then downsample the mouse signal and synchronize it with the gaze signal via linear interpolation. After signal preprocessing, we obtain a sequence of on-screen mouse locations $\mathcal{M} = \langle p_M^{(0)}, p_M^{(1)}, \cdots p_M^{(n)} \rangle$ and the sequence of on-screen gaze locations $\mathcal{G} = \langle p_G^{(0)}, p_G^{(1)}, \cdots p_G^{(n)} \rangle$, which indicate the movement trajectories of mouse and gaze as shown in Fig. 2 (a). To achieve rotation- and translation-invariance, we then transform the movement signals from the screen coordinate to the *attraction coordinate system* (ACS).Specifically, we get the midpoints of mouse and gaze movements by $m_M^{(i)} = \frac{1}{2}(p_M^{(i)} - p_M^{(i-1)})$ and $m_G^{(i)} = \frac{1}{2}(p_G^{(i)} - p_G^{(i-1)})$ respectively. We set the x-axis of the ACS as the vector connecting the midpoints of mouse and gaze movement vectors (i.e., $\overrightarrow{m_M^{(i)} m_G^{(i)}}$), and y-axis as its orthogonal vector. Fig. 2 demonstrates the examples of ACS transformation. As the figure shows, ACS cares only about the relative location and movement between mouse and gaze, regardless of their absolute location and moving direction in the screen. The transformations of any two samples of MGMovements in ACS will be similar, as long as their behavior patterns are similar, even if they differ in absolute screen location and moving direction. Hence, the proposed ACS are rotation- and translation-invariance. Our MGAttraction measurement is computed based on ACS, it is thus also rotation- and translation-invariance. Furthermore, the method works independently, without the need of UI information. It is thus also UI-agnostic.

3.1.2 MGAttraction Computation. The attraction captures the intensity of the relative movement of mouse and gaze, which should be negatively correlated with their distance and positively to their movement speed. The overall idea is to leverage the relative velocity and distance to delineate the "attraction" between mouse and gaze over time. For example, if mouse and gaze locations are close and approaching each other at high speed, they exhibit a strong attraction. If the mouse “chases” the gaze at a higher velocity than the velocity of the gaze “escaping” from the mouse, then the mouse exerts a larger positive attraction, and the gaze experiences a smaller negative attraction.

Specifically, the overall attraction between mouse and gaze consists of the attractions exerted by the mouse, $attr_M$, and exerted by the gaze, $attr_G$. We resolve the velocities of mouse and gaze in vector form into the x- and y-components (or horizontal (h) and vertical (v) components). $attr_M$ and $attr_G$ can be formulated in a symmetric manner:

$$attr_M = \frac{\alpha_M V^h_{M|G} \left|V^h_M\right|}{D} + \frac{\beta_M V^v_{M|G} \left|V^v_M\right|}{D} \tag{1}$$

$$attr_G = \frac{\alpha_G v^h_{G|M} \left|V^h_G\right|}{D} + \frac{\beta_G V^v_{G|M} \left|V^v_G\right|}{D} \tag{2}$$

where

- $D^{(i)}$ is the Euclidean distance between $m_G^{(i)}$ and $m_M^{(i)}$
- V^h_M, V^v_M, V^h_G, and V^v_G are the horizontal and vertical component velocities of mouse and gaze in the attraction coordinate
- $V^h_{M|G}$ and $V^v_{M|G}$ indicate velocity components of mouse relative to gaze
- $V^h_{G|M}$, and $V^v_{G|M}$ indicate velocity components of gaze relative to mouse.

The relative velocity components can be computed as:

$$V^h_{M|G} = V^h_M - V^h_G; \quad V^v_{M|G} = V^v_M - V^v_G \tag{3}$$

$$V^h_{G|M} = V^h_G - V^h_M; \quad V^v_{G|M} = V^v_G - V^v_M \tag{4}$$

Finally, α and β denote the signs of the attraction components:

$$\alpha_M = sgn(V^h_{M|G}(h_G - h_M)); \beta_M = sgn(V^v_{M|G}(v_G - v_M)) \tag{5}$$

$$\alpha_G = sgn(V^h_{G|M}(h_M - h_G)); \beta_G = sgn(V^v_{G|M}(v_M - v_G)) \tag{6}$$

where h_M, v_M, h_G, v_G are the current mouse and gaze location projected on the two axes. In other words, the sign of a component is positive when the mouse and gaze are moving towards each other. Otherwise, it is negative.

3.2 Inferring Mental Stress from MGAttraction Signals

Our proposed method detects mental stress based on relative movement between mouse and gaze. Therefore, we are interested in the periods during which both mouse and gaze can be detected. While the position of the mouse can always be detected, the gaze cannot be detected by the eye-tracker or webcam during eye blinks and when the user's gaze is off-screen.

We handle off-screen eye periods in two ways. Since it is known that a human eye-blink is usually shorter than 150ms [9], we discard time periods longer than 150ms during which the gaze cannot be captured. For the remaining time periods, we estimate missing gaze locations using linear interpolation. We then compute and record the MGAttraction signals $attr_M$ and $attr_G$ in a period for both mouse $Attr_M$ and gaze $Attr_G$ separately by following the definitions introduced in section 3.1. This gives us:

- $Attr_M = [attr_M1, attr_M2, ..., attr_Mn]$
- $Attr_G = [attr_G1, attr_G2, ..., attr_Gn]$

where $attr_Gi$ and $attr_Mi$ stand for the i^{th} gaze attraction value and i^{th} mouse attraction in that period.

Equation 1 and 2 shows that $attr_M$ and $attr_G$ are positively correlated to the magnitude of mouse and gaze velocity respectively. However, the speed of gaze is normally much faster than mouse, which leads to a much larger range for $attr_G$ than $attr_M$. To facilitate the following analysis, we normalize the *magnitude* of $attr_G$ and $attr_M$ to bring them into the range [-1, 1].

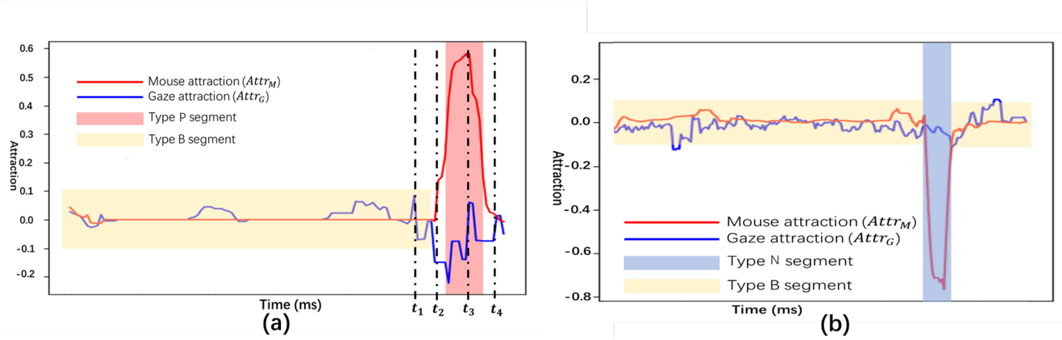


Fig. 3. Two example periods of mouse attraction and gaze attraction with a Type P segment (red) and a Type B segment (yellow) and a Type N segment (blue).

3.2.1 *Segmenting the MGAttraction Signal.*

Based on the measured MGAttraction signals, we extract MGAttraction features to model the relative behaviors of mouse and gaze. The inputs to our method are the mouse and gaze signals that occur over a period of time (e.g., 5 minutes), in which the users are working with their computers. We refer to the entire detection period as "session" hereinafter. For a given session, there may be multiple sub time windows that indicate different mouse and gaze behaviors. For instance, Fig. 3 illustrates two example time periods. We observe that the shape of $Attr_M$ resembles that of a pulse signal, in that there is an acute change in the amplitude of a signal from a baseline value to a higher or lower value, followed by a rapid return to the baseline value. The baseline value of $Attr_M$ is 0, which is measured when the mouse is at rest. A positive pulse period indicates that the mouse is approaching the gaze. A negative pulse indicates the opposite (mouse departing from gaze). In the example shown in Fig. 3 (a), both the gaze and mouse are stable with no relative movement before Time t_1. At t_1, the gaze starts to move to a new target, thus leaving the mouse. The mouse starts moving at Time t_2 to catch up with the gaze. After Time t_3, the intensity of the relative movement between the gaze and mouse decreases till Time t_4, when both gaze and mouse are stable again. The MGMovements before and after t_1 apparently indicate two different behaviors. A similar interpretation can be made for the example period shown in Fig. 3 (b).

In order to model the MGAttraction behaviors more precisely, it is better to extract MGAttraction features from each of the sub time windows separately. To this end, we divide the entire detection period (i.e., a session) into several non-overlapping sub time windows and categorize them based on the behavior pattern exhibited by $Attr_M$. We refer to the sub time window as "segments". Fig. 4 depicts an example of the segmentation and MGAttraction features extraction. Three types of segments are defined. A Type P segment is a period during which $Attr_M$ shows a positive pulse, a Type B segment is a period during which the magnitude of $Attr_M$ stays around the baseline value, and a Type N segment is a period during which $Attr_M$ shows a negative pulse. Examples of Type P, B and N segments are shown in Fig. 3.

Algorithm 1 shows the procedures to identify Type P and N segments. By definition, Type B segments are the time periods that mouse is at rest. Specifically, mouse speed is less than 75 px/sec and the cursor is within a circle with radius 10 px for more than 1.2 seconds [45]. All the parameters involved in the algorithm are determined based on the findings of previous work that studied mouse

Algorithm 1 Automatic MGAttraction Signal Segmentation For Type P and N

```
procedure SIGNAL SEGMENTATION($Attr_M$)   ▷ $Attr_M$: 1-D array of mouse MGAttraction signal
    $Ps \leftarrow []$
    $Ns \leftarrow []$   ▷ initialization
    $Attr_M^{positive} \leftarrow \{e \in Attr_M | \, e > 0\}$   ▷ Get all positive values in $Attr_M$
    $Attr_M^{negative} \leftarrow \{e \in Attr_M | \, e < 0\}$   ▷ Get all negative values in $Attr_M$
    $thres_p \leftarrow mean(Attr_M^{positive}) + std(Attr_M^{positive}) \times 3$   ▷ Get the threshold of peaks
    $thres_n \leftarrow mean(Attr_M^{negative}) + std(Attr_M^{negative}) \times 3$   ▷ Get the threshold of valleys
    $is_p \leftarrow [idx \mid Attr_M[idx] == thres_p]$   ▷ Get all indices that value of $Attr_M$ equals $thres_p$
    $is_n \leftarrow [idx \mid Attr_M[idx] == thres_n]$   ▷ Get all indices that value of $Attr_M$ equals $thres_n$
    for every two consecutive values $i$ and $j$ in $is_p$ do
        if $Attr_M[v] >= thres_p$ for $\forall v \in [i, j]$ then
            $s \leftarrow$ $first\ index\ that\ Attr_M[s] == 0\ and\ s <= v$
            $e \leftarrow$ $first\ index\ that\ Attr_M[e] == 0\ and\ e >= v$
            $Ps.insert(Attr_M[s:e])$   ▷ $Attr_M[s:e]$ is in Type P
        end if
    end for
    for every two consecutive values $i$ and $j$ in $is_n$ do
        if $Attr_M[v] <= thres_n$ for $\forall v \in [i, j]$ then
            $s \leftarrow$ $first\ index\ that\ Attr_M[s] == 0\ and\ s <= v$
            $e \leftarrow$ $first\ index\ that\ Attr_M[e] == 0\ and\ e >= v$
            $Ns.insert(Attr_M[s:e])$   ▷ $Attr_M[s:e]$ is in Type N
        end if
        return $Ps, Ns$
    end for
end procedure
```

activities in HCI tasks [24, 45]. Particularly, Huang et al. [24] investigated the relationship between human mouse and gaze behaviors during web searching, which is similar to our experimental task. They found that the distribution of distances between the cursor and gaze point follows a normal distribution. In our MGAtraction model, we also observed that the attraction values of $attr_M$ and $attr_G$ follow a normal distribution, and remain relatively stable until another segment occurs. To accurately segment the Type P and Type N segments, we use the empirical rule of $mean + 3 * std$ to identify values that deviate significantly from the mean. We then consider these particularly high and low values to be the effective "peaks" and "valleys" respectively.

MGAttraction features are then extracted from each of the segment. We refer to this as *segment-level feature*. Based on that, we construct the *session-leve feature* for stress detection. Fig. shows the overall pipeline for the feature extraction. Fig. 4 shows the overall pipeline for the feature extraction. Specifically, the final session-level feature vector is composed of two parts: ϕ and S. ϕ contains the segment-level features that are extracted from each type of MGAttraction signal segment, which describe the relative movement between gaze and mouse during the segment periods. S contains the statistical features extracted from the session-level MGAttraction signals $attr_M$ and $attr_G$, which model the macro behaviors of gaze and mouse over the entire session period.

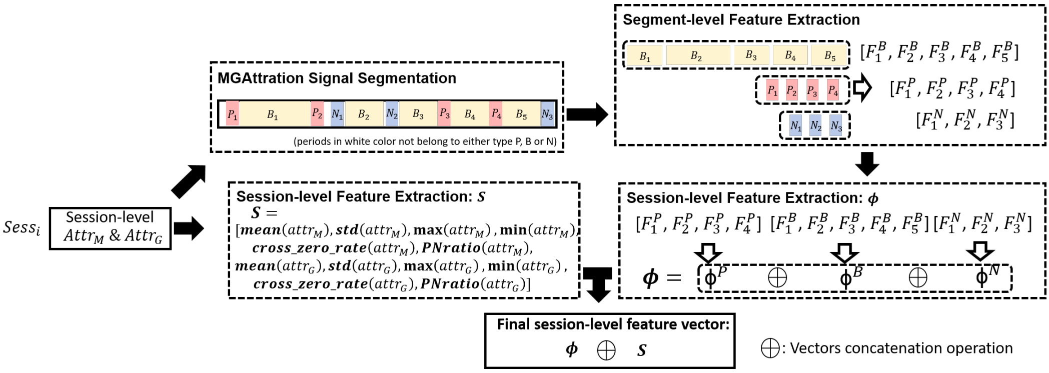

Segment-level Feature Extraction
B_1 B_2 B_3 B_4 B_5
$[F_1^B, F_2^B, F_3^B, F_4^B, F_5^B]$
P_1 P_2 P_3 P_4
$[F_1^P, F_2^P, F_3^P, F_4^P]$
N_1 N_2 N_3
$[F_1^N, F_2^N, F_3^N]$
MGAttration Signal Segmentation
P_1 B_1 P_2 N_1 B_2 N_2 B_3 P_3 B_4 P_4 B_5 N_3
(periods in white color not belong to either type P, B or N)
$Sess_i$
Session-level $Attr_M$ & $Attr_G$
Session-level Feature Extraction: S
$S =$
$[\textbf{mean}(attr_M), \textbf{std}(attr_M), \textbf{max}(attr_M), \textbf{min}(attr_M), \textbf{cross\_zero\_rate}(attr_M), \textbf{PNratio}(attr_M), \textbf{mean}(attr_G), \textbf{std}(attr_G), \textbf{max}(attr_G), \textbf{min}(attr_G), \textbf{cross\_zero\_rate}(attr_G), \textbf{PNratio}(attr_G)]$
Session-level Feature Extraction: ϕ
$[F_1^P, F_2^P, F_3^P, F_4^P]$ $[F_1^B, F_2^B, F_3^B, F_4^B, F_5^B]$ $[F_1^N, F_2^N, F_3^N]$
$\phi = \phi^P \oplus \phi^B \oplus \phi^N$
Final session-level feature vector:
$\phi \oplus S$
$\oplus$: Vectors concatenation operation

Fig. 4. Overall Pipeline of Feature Extraction.

3.2.2 *Segment-level Feature Extraction.*

The second step of the feature extraction process is the segment-level feature extraction. As shown in Fig 4, segments with the same type are considered together. For each segment type, we then extract segment-level features (F^P, F^B, F^N) to describe the relative movement behaviors between gaze and mouse in the Type P, B and N segments.

Type P segments mainly involve behavior exhibited when a subject moves the mouse towards the gaze point. The segment-level features F^P quantify behaviors such as how vigorously the mouse approaches the gaze, and how much the gaze leads the mouse. Both of these behaviors have been shown to be indicative of different mental states [26].

Table 2 shows the extracted features from the segment in Type P. $f_1^p - f_4^p$ quantify the overall MGAttraction level for both gaze and mouse, and $f_5^p - f_6^p$ describe the time needed for mouse and gaze to reach the largest attraction level. When a user is stressed, the speed of movement for both gaze and mouse tend to increase [48], which can be reflected by features $f_1^p - f_6^p$.

$f_5^p - f_8^p$ capture the latency information in the coordination between the mouse and gaze movements, such as the time difference between when the mouse and gaze start moving and when they reach their largest attraction value. Figure 5 gives an example of what these features

Table 2. F^P: Features Extracted from the Type P Segment.

Feature	Meaning	Formulation
f_1^p, f_2^p	Mean, max of mouse attraction	Mean and max values of $attr_M$ during the segment
f_3^p, f_4^p	Mean, max of gaze attraction	Mean and max values of $attr_G$ during the segment
f_5^p	Timepoint when mouse exhibits the strongest attraction	Timepoint when the absolute value of $attr_M$ is the largest
f_6^p	Timepoint when gaze exhibits the strongest attraction	Timepoint when the absolute value of $attr_G$ is the largest
f_7^p	Starting level of gaze attraction	$attr_G$ at the beginning of the segment
f_8^p	Latency of peaks	Time difference between the points when $attr_G$ and $attr_M$ reach their maximum absolute values
f_9^p	Duration of the segment	Total length of the segment

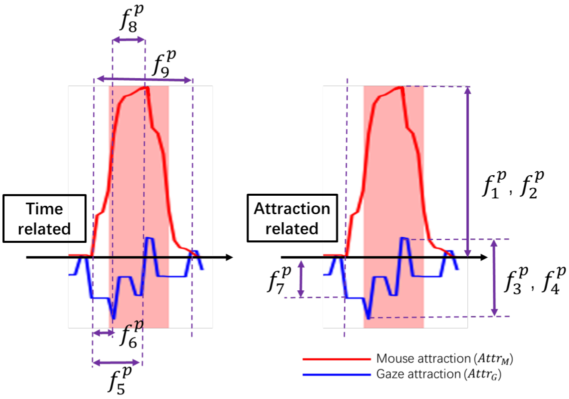


Fig. 5. Illustration of Features Extracted in F^P. The x-axis shows the timeline from the start to the end of the segment. The y-axis shows the MGAttraction value.

would look like in a sample Type P segment, where the x-axis indicates the timeline and the y-axis indicates the attraction of mouse and gaze.

Compared to the Type P segment, the Type N segment describes a time period during which the mouse departs from the gaze. It can be seen as the "upside-down" version of the Type P segment. Therefore, we extract the same features F^N as F^P from the Type N segment, shown in Table 2.

Type B segments are those where the mouse is stationary for the entire segment period. Hence we only extract features from $attr_G$ (Table 3). $f_1^B - f_3^B$ depict the overall intensity of gaze movement attraction, and $f_4^B - f_9^B$ are designed to model the movement by which gaze is approaching or departing from the mouse. When a user has a clear idea about the next target and moves the mouse purposefully towards it, then $attr_G$ should show only one negative pulse with a large amplitude. However, if a user does not have a clear idea about the next target, he/she is likely to look around, which may generate a couple of negative and positive pulses with a small amplitude.

Table 3. F^B: Features Extracted from the Type B Segment.

Feature	Meaning	Formulation
f_1^B, f_2^B, f_3^B	Mean, max, min of gaze attraction	Mean, max, min of $attr_G$ in the segment period
f_4^B, f_5^B	Mean of positive gaze attraction	Mean of all positive and negative values of $attr_G$ in the segment period
f_6^B, f_7^B	Duration of gaze shows positive and negative attraction	Accumulated sum of time duration that $attr_G$ is positive and negative in the segment period
f_8^B, f_9^B	Power of positive and negative gaze attraction	Accumulated power of positive and negative gaze attraction
f_{10}^B	Duration of the segment	Total time duration of the segment

$*$ **Power** stands for integral of attraction over time

3.2.3 Session-level Feature Extraction. According to the feature extraction pipeline from Figure 4, the final session-level feature vector is constructed by concatenating the ϕ and S components. ϕ contains the aggregated features constructed from segment-level feature vectors and S consists of statistical features that model the overall trend of $attr_G$ and $attr_M$ for the entire session.

The first part of the session-level feature vector ϕ contains two statistical features extracted from the generated segment-level feature vectors. The first feature is the average behavior among all the segments and the second one captures the variation of behavior among all the segments. Specifically, suppose a session $Sess$ consists of k Type P segments. For the i^{th} Type P segment in $Sess$, we can extract a segment-level feature vector F_i^P, where $i \in [1, k]$. ϕ^P is the aggregated feature vector extracted from F_i^P by computing the mean value and the standard deviation of each f_j^P and $j \in [1, 9]$. By following the same procedure, we can also generate ϕ^B, ϕ^N and ϕ by concatenating ϕ^P, ϕ^B and ϕ^N together.

For the second part of the session-level feature vector S, we extract statistical features from the session-level $attr_M$ and $attr_G$ signal, including **mean**, **standard deviation**, **max**, **min**, **zero crossing rate** (per second) (i.e. the number of times per second that the signal moves from positive to negative, and vice versa) and **NPratio**, where NPratio is computed as the accumulated duration during which the signal is negative divided by the accumulated duration during which the signal is positive. Our expectation is that these statistical features can capture the overall trend of signals for the whole session. Therefore, the final session-level feature vector is built by concatenating ϕ and S.

3.3 Estimate Gaze Locations from Webcam Video

In a more realistic context, eye tracker is not available in common desktop environments in actual workplaces. To further improve the applicability of our method, we also explore using gaze locations estimated from webcam video to build MGAttraction signals and stress detection model.

Figure 6 shows the process of estimating gaze locations from the webcam video. We treat the video as a sequence of frames recording the subjects' face and upper body. The webcam camera is fixed in the middle of the top of the display, which is about 60 cm away from the subject. In order to estimate gaze locations from the webcam video, we use the state-of-the-art **OpenFace 2.0** [3] to extract facial features related to head pose and eye gaze direction as shown in Table 4. For each valid $frame_i$ at t_i in the video, where valid is defined by the ability of the OpenFace algorithm to capture the subject's head and face, a facial feature vector $F_{t_i}^{Facial}$ is extracted. This feature vector

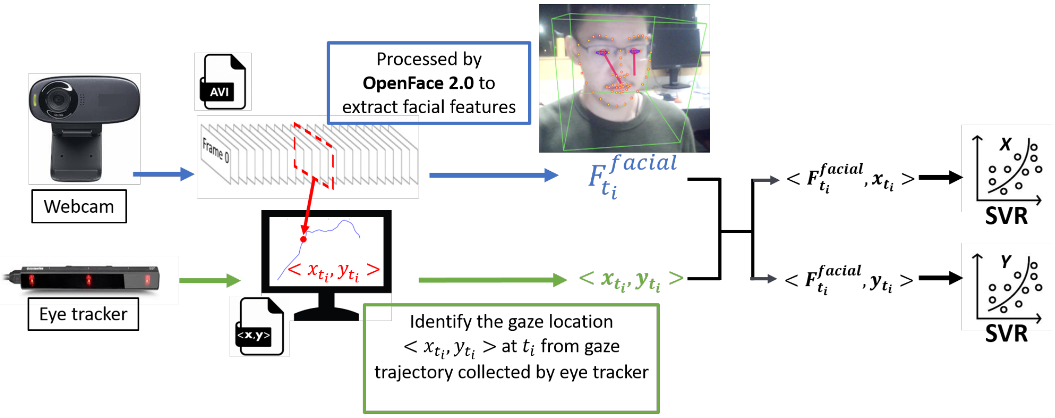

Processed by
OpenFace 2.0 to
extract facial features
AVI
Frame 0
Webcam
Eye tracker
<x,y>
$F_{t_i}^{facial}$
$< x_{t_i}, y_{t_i} >$
$< x_{t_i}, y_{t_i} >$
Identify the gaze location
$< x_{t_i}, y_{t_i} >$ at t_i from gaze
trajectory collected by eye tracker
$< F_{t_i}^{facial}, x_{t_i} >$
$< F_{t_i}^{facial}, y_{t_i} >$
X
SVR
Y
SVR

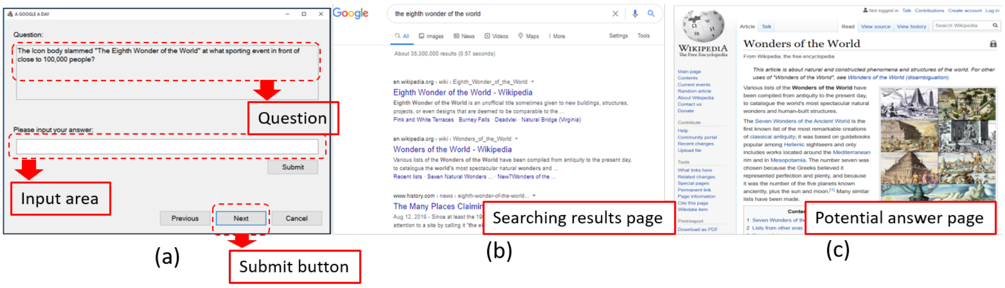
Fig. 6. Overall Pipeline of Estimating Gaze Locations from Webcam Video Frames.

contains 12 features. At the same time, we also record the gaze location $<x_{t_i},y_{t_i}>$ on the screen at t_i, as captured by the eye-tracker, to be the ground truth for training purposes. Two SVR models are then trained to construct the mapping from the facial feature vectors to the estimated x and y gaze locations on the screen: $F^{Facial} \longrightarrow x_{estimated}$ and $F^{Facial} \longrightarrow y_{estimated}$.

Table 4. F^{Facial}: Facial Features Extracted from Webcam Video.

Feature	Meaning	Formulation
$head_{Tx}, head_{Ty}, head_{Tz}$	Location of the head	Location of the head corresponding to webcam in millimeters and positive Z is the direction away from the camera
$head_{Rx}, head_{Ry}, head_{Rz}$	Rotation of the head	Pitch (Rx), yaw (Ry), and roll (Rz) of the head with webcam being the origin
$L\_gaze_x, L\_gaze_y, L\_gaze_z$	Left eye gaze direction vector	Left eye gaze direction vector in the webcam coordinates with webcam being the origin
$R\_gaze_x, R\_gaze_y, R\_gaze_z$	Right eye gaze direction vector	Right eye gaze direction vector in the webcam coordinates with webcam being the origin

Our gaze estimation process gives us a sequence of on-screen estimated gaze locations, which forms the estimated gaze trajectory $\mathcal{GE} = \langle p_{GE}^{(0)}, p_{GE}^{(1)}, \cdots p_{GE}^{(n)} \rangle$. We follow the same procedures in Section 3.1 and 3.2 to generate estimated mouse attraction $attr_{ME}$ and estimated gaze attraction $attr_{GE}$ from the sequence of on-screen mouse locations $\mathcal{M}$ and the sequence of on-screen estimated gaze locations $\mathcal{GE}$ respectively. Session-level features $(\phi_E \oplus S_E)$ are extracted from $attr_{ME}$ and $attr_{GE}$ to discriminate between stress and relax sessions.



The Icon body slammed "The Eighth Wonder of the World" at what sporting event in front of close to 100,000 people?

the eighth wonder of the world

About 35,300,000 results (0.57 seconds)

en.wikipedia.org › wiki › Eighth_Wonder_of_the_World
Eighth Wonder of the World - Wikipedia
Eighth Wonder of the World is an unofficial title sometimes given to new buildings, structures, projects, or even designs that are deemed to be comparable to the ...
Pink and White Terraces · Burney Falls · Deadvlei · Natural Bridge (Virginia)

en.wikipedia.org › wiki › Wonders_of_the_World
Wonders of the World - Wikipedia
Various lists of the Wonders of the World have been compiled from antiquity to the present day, to catalogue the world's most spectacular natural wonders and ...
Recent lists · Seven Natural Wonders ... · New7Wonders of the ...

www.history.com › news › eighth-wonder-of-the-world...
The Many Places Claimin
Aug 12, 2019 - Since at least the 19
attention to a site by calling it "the ei

Wonders of the World

From Wikipedia, the free encyclopedia

This article is about natural and constructed phenomena and structures of the world. For other uses of "Wonders of the World", see Wonders of the World (disambiguation).

Various lists of the **Wonders of the World** have been compiled from antiquity to the present day, to catalogue the world's most spectacular natural wonders and human-built structures.

The Seven Wonders of the Ancient World is the first known list of the most remarkable creations of classical antiquity; it was based on guidebooks popular among Hellenic sightseers and only includes works located around the Mediterranean rim and in Mesopotamia. The number seven was chosen because the Greeks believed it represented perfection and plenty, and because it was the number of the five planets known anciently, plus the sun and moon.[1] Many similar lists have been made.

Conte
1 Seven Wonders of th
2 Lists from other eras

Fig. 7. Experiment interface. (a) Question page (b) Searching results page and (c) Potential answer page.

4 EXPERIMENTS

The aim of the study is to detect stress in a real-world scenario. Hence, we constructed a dataset that satisfies the following requirements: (1) the task used to evaluate should be a commonly-encountered computer interaction task (2) with dynamic-UI. Given that web search is one of the most ubiquitous of activities, we use a web search task for our study. In our experiment, subjects were required to answer some questions via web searching, which are randomly selected from the question-answering game “A Google a Day”.

Figure 7(a) shows the question-answering interface. We first confirmed that the subject did not know the answer to the posed question in advance. If the subject already knew the answer, a new question was re-selected. Subjects were asked to type their answers in the input area and used the submit button to check the correctness. Subjects could repeatedly submit attempts until they found the correct answer, or (when stress induction is applied) reached the 5-minute time limit.

The questions in the "A Google A Day" task are formulated such that it is not possible to find the answer by simply copying and pasting the question into the search query. As an example, one sample question asks *"The icon body slammed "The eighth wonder of the world" at what sporting event in front of close to 100,000 people?"* To answer these questions, subjects have to iteratively rephrase and refine the search keywords according to the information retrieved from previous searches. This means that they would be led to different webpages, which they might browse through, or even follow links off, to obtain the final answer. Since these webpages would have different UI layouts, and it was not possible to forecast which keywords the subjects would use and which websites they would click into, this gives us a dynamically changing UI environment. Some example webpages are shown in Figure 7(b) and (c).

Each subject was required to accomplish 12 games. Each game required the subject to find the correct answer to one "A Google A Day" question. 6 of the games, which we termed as *relaxed*, did not have a time constraint and subjects could take as long as they liked to finish the task. The other 6 games subjected the experiment subjects to a 5-minute time limit per game to induce stress. The inclusion of time pressure has been shown in many previous studies [28, 34] as an effective way to induce mental stress. To further ensure that the stress level was indeed increased, a sound cue countdown was included.

The order of the relaxed and stressed games was determined randomly to even out the fatigue factor. The experiment started with one warm-up game to familiarize the subject with the experimental procedure and the experimental settings. After each game, subjects were required to report their stress level on a 5-point Likert scale, with 1 being "totally not stressed" to 5, “fully stressed”. A 15-minute break was introduced between every two games to allow subjects to relax and to recalibrate the eye-tracker. In total, 15 subjects were involved in this experiment. Games during which the subject self-reported a contradictory stress score were filtered. Particularly, stress (relax) games during which the subject self-reported a stress score lower (higher) than 3 were discarded. Our final dataset contains 175 games, 90 of which were labeled as stress. Table 5 summarizes the overall information of our dataset.

Table 5. Summary of the Games Collected from Our Subjects (15 in total).

Games **From**	Relaxed	Stress	Total
Each subject	6	6	12
Total subjects	90	90	180
Total subjects (non-contradictory)	85	90	175

The experiment was conducted in a conventional office setting, which is shown in Figure 8. The setup was composed of a 22" LCD monitor at 1600×1000 resolution, a full-size QWERTY keyboard and a standard optical mouse. Subjects sit around 60 cm away from the screen with their preferable chair heights and screen heights. Three different modalities of data were collected during the experiment: (1) Eye gaze location data captured by the Tobii EyeX eye-tracker at 60 Hz, which was attached to the bottom of the display, (2) mouse location data captured by a C++ program in 120 Hz and (3) video of subjects' face and upper body captured by a standard webcam fixed on the top of the display at 30 Hz. All the data collecting programs were running in the background to avoid disturbing subjects' interactions.

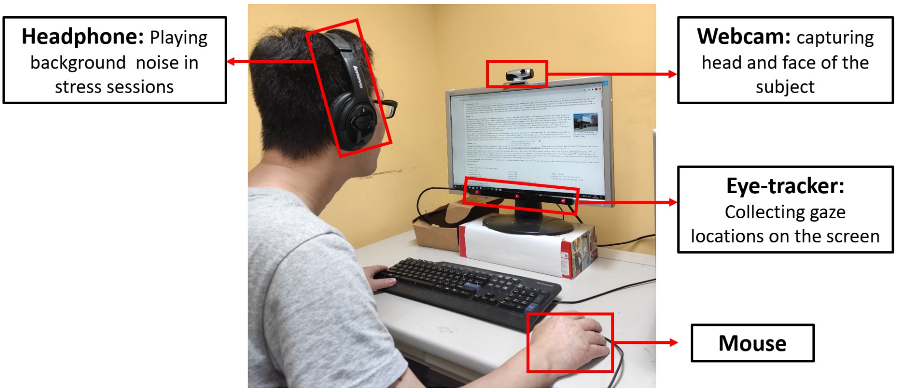

Headphone: Playing background noise in stress sessions
Webcam: capturing head and face of the subject
Eye-tracker: Collecting gaze locations on the screen
Mouse

Fig. 8. Experimental Environment.

In our evaluation experiment, we take the entire period of playing a searching game (i.e., conducing a web searching task) as a session to evaluate our stress detection method. In other words, we input the mouse and gaze signals that occur during each searching game to our stress detection method, to detect the subject's stress state for that period.

5 EXPERIMENTAL EVALUATION

In this section, we first prove the feasibility of detecting mental stress via mouse and gaze behaviors modeled by the MGAttraction coordinate system by evaluating the performance of our method on the dataset constructed in section 4.

The motivation behind our work is to develop methods of stress detection that would work under general-use contexts, hence the UI-agnostic approach that is followed. However, the acquisition of eye gaze locations introduces a major challenge. Eye gaze locations are normally collected by eye trackers, which are not commonly encountered outside of the research lab except for some very specialized contexts, such as accessibility. Even though the development of more affordable eye tracker models such as the Tobii EyeX [19] has brought the eye tracker within the limits of affordability for consumers, an approach that requires the use of an eye tracker cannot reasonably be considered to be general purpose.

We therefore consider both the *upper bound* and *realistic* contexts in our evaluation. The upper bound evaluation is intended to test the feasibility of our UI-agnostic approach. To alleviate confounding issues that might be caused by noisy data, we therefore use the Tobii EyeX tracker to capture eye gaze locations. The EyeX returns eye gaze locations with a less than 0.6 degree margin of error. Though not perfect, this performance has been previous found to be adequate for research applications [19]. It also has the advantage of being potentially something that might be purchased by consumers, albeit for specialized purposes.

On the other hand, we also evaluate the performance of our approach under more realistic contexts. For this purpose, we estimate the eye gaze locations from video captured by a front-facing webcam. The webcam is a commonly-found piece of equipment in most home and office setups, and the placement above the screen in the center, facing the user head-on, is a common location. Since the accuracy of the eye gaze locations estimated from the webcam video would be expected to be lower than that obtained from the eye tracker, we expect that it would have a corresponding impact on the performance of our stress detection method.

Another issue that has to be tackled in the context of our work is that of unknown users. In real applications, especially with communal machines such as lab or classroom contexts, a machine would be used by many users, and there is always the possibility that it is a user who has never used the machine before. Our evaluation therefore uses the leave-one-subject-out approach — specifically, the model is trained on data from N_s-1 subjects (training set) and evaluated on data from the remaining subject (test set). This process is iterated N_s times, each time with a different test subject. The average correct classification rate (CCR) is reported as the overall evaluation performance.

5.1 Feasibility Study: Stress Detection via Mouse and Eye Tracker-captured Gaze Behaviors

We follow the procedures from Figure 4 to extract 66 features (12 in S, 54 in ϕ) for each session. A Random Forest (RF) classifier is used to discriminate the stressed and relaxed session based on the extracted features. RF has been used in many similar contexts [15, 44, 50, 51] and has the advantages that it is able to (1) handle non-linear data, (2) be somewhat robust to outliers, (3) produce a low bias and moderate variance result, and (4) quantify the relative importance of the features, which helps with interpreting the model.

Table 6. Classification Performance for Stress Detection via Mouse and Gaze Behaviors (eye-tracker based).

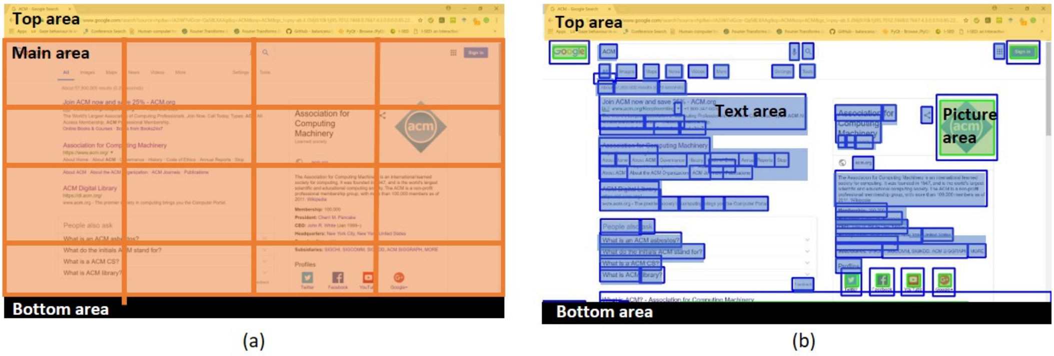
Performance / Class	Precision	Recall	F-measure
Relax	0.82	0.73	0.77
Stress	0.77	0.84	0.80
Weighted Average	0.79	0.79	0.79

Table 7. Confusion Matrix for Stress Detection via Mouse and Gaze Behaviors (eye-tracker based)

Predicted as Ground truth	Relax	Stress	Total
Relax	**62**	23	85
Stress	14	**76**	90
Total	76	99	175

Table 6 shows the performance achieved during the feasibility study, and Table 7 presents the confusion matrix. The average CCR achieved for two classes is 78.8%, which is significantly higher than the baseline of 51.4%, which is achieved by classifying every instance as the majority class (Stress). The false alarm rate is less than 0.25, which suggests that our approach can balance between over-reporting possible stress and the danger of missing reporting. Moreover, the weighted average F-measure of our approach is close to the weighted average precision and recall. It illustrates that our approach does not sacrifice either one of precision or recall for the other. Overall, the results indicate that our approach can successfully detect stress in a real-world scenario via mouse and gaze behaviors.

We further study the achieved performance by comparing with other state-of-the-art, dynamic UI-based approaches in the web search task, in contexts where the experiment subject is required to complete an entire task and the stress level is measured on the level of the overall task. However, some state-of-the-art approaches rely on UI related features, such as the dwell duration of mouse and gaze within a particular UI area, or the speed and frequency of mouse and gaze travel between each UI component, and the gaze transition sequence among UI components. In order to evaluate these approaches on our dataset, we implement a module to extract the UI information from the current webpage, which is provided to the approaches that we are comparing ours against. We experiment with two methods for extracting the UI component information dynamically: heuristic and content-based.



(a)

(b)

Fig. 9. Dynamic-UI Component Detection Methods: (a) Heuristic-based, (b) Content-based.

The heuristic-based method divides the whole UI interface into several sub-areas based on the heuristic knowledge of browsers' standard UI design. As shown in Figure 9(a), we first extract the top and bottom sub-areas and then further evenly divide the middle area into 4×4 sub-areas. Different UI components appear in each sub-area with different frequencies. For example, in the Google result page, links in the text form often appear in the two left columns of the sub-areas, and users always pay more attention to the top two rows of the sub-areas. On the Wikipedia page, pictures often appear in the right two columns. A gaze movement sequence can therefore be constructed in the form of transitions between different sub-areas.

The content-based method extracts UI information based on computer vision techniques. We adopt canny edge detection algorithms to segment different UI component areas, including button area, input area, text-content area and picture-content area. An example of the UI components division result is shown in Figure 9(b).

Table 8. Performance of different approaches in dynamic-UI task.

Performance / **Model**	CCR	Precision (Stress)	Recall (Stress)
Huang et al. [26]	58.9%	0.60	0.62
Wang et al.[48] + heuristic-based UI	62.8%	0.62	0.71
Wang et al.[48] + content-based UI	67.1%	0.64	0.82
Our approach	78.8%	0.77	0.84

With the help of the dynamic-UI information extraction module, we evaluate three different state-of-the-art approaches on our web search task, the performances of which are shown in Table 8. The results suggest that our approach achieves the best performance, which is around 20% improvement over StressClick [26] and more than 10% improvement over [48] using our content-based UI extraction module. One possible reason that StressClick does not perform as well as our method is that StressClick only considers the gaze behaviors relative to the mouse within a small time-window around each mouse click, which may not be sufficient enough to detect the mental state in a complex task with dynamic-UI. Our hypothesis is borne out by the observations that [48] yields better performance than StressClick, especially when it is provided with detailed UI information (content-based UI extraction), which allows it to take into account more information related to mouse and gaze movement behaviors. However, it also tends to generate many false positive (stress) instances and is fairly sensitive to the quality of extracted UI information. This can be seen from the fact that when the heuristic-based module, which is not able to accurately analyze the UI, is used, the performance drops to a CCR of 62.8%. This is a limitation of their method, as real-time extraction of UI components in dynamic-UI tasks is expensive since it usually requires heavy image processing computation. In conclusion, our results illustrate that our MGAttraction approach can successfully detect stress in a real-world scenario with balanced precision and recall and a low false alarm rate with low computation cost.

To better understand the features and how they work to detect stress, we output the top 5 most important features considered by RF. Figure 10 presents the distributions of each important feature across the relaxed and stressed groups. Each bar stands for a distribution (green for relaxed and red for stressed), where the yellow line marks the mean. The box covers the first to the third quartile and the whiskers cover the range from minimum to maximum, except for outliers, which are denoted by hollow circles. A t-test is applied to determine whether there is a significant difference between the means of the two groups. If the p-value of the t-test is less than 0.01, which means the difference of means is statistically highly significant, it is annotated with "**" at the top of the figure. If the p-value is in [0.01, 0.05), which means the difference is statistically significant, it is marked with "*", and statistically non-significant features are marked with "x".

Figure 10 suggests that important features have different distributions between the relax and stress groups. The t-test shows that four of them are significantly or highly significantly different between the two groups. $attr_G$ of the stressed group exhibits a lower **crossing zero rate**. In

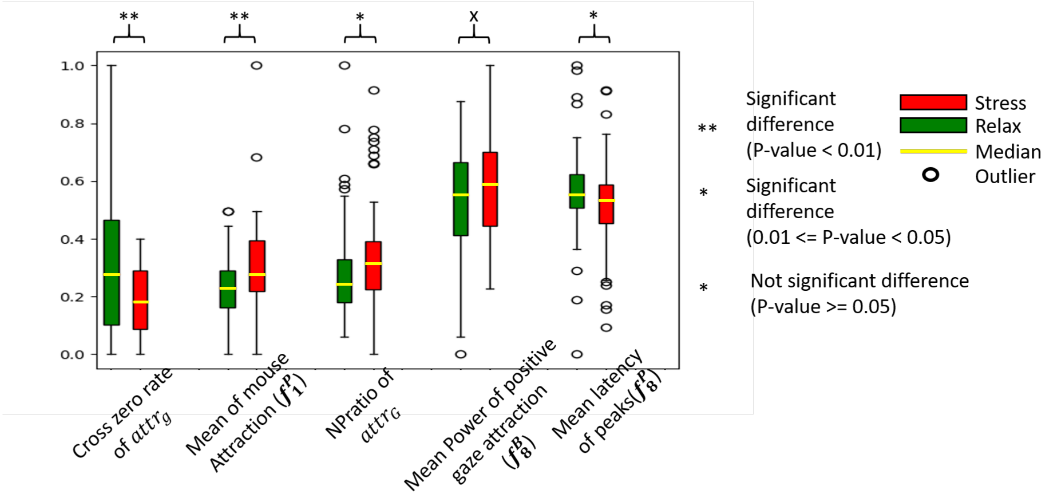


Fig. 10. Distributions of selected important features.

physical terms, every time $attr_G$ crosses the boundary between positive to negative (or vice versa), this indicates a change in the direction of the gaze movement (e.g., from leaving the mouse to approaching it). A higher cross-zero rate therefore implies that the gaze is moving back and forth, without a clear target in mind.

We also note that $attr_G$ exhibits a higher **NPratio**. This implies that in most cases, the gaze is directly moving toward the target and leading the mouse. This, together with the lower zero crossing rate, suggests that gaze movement is more consistent when the subject is stressed.

Finally, it can be seen that when the user is stressed, the value of the **mean latency of peaks** is smaller. This indicates that the distance (in the time domain) between mouse and gaze is smaller. The **mean of mouse attraction** value is larger, which means that the mouse has a greater tendency to move. Putting together, this suggests that when under stress, the mouse catches up with the gaze more quickly, and with less delay.

One possible explanation for the above behaviors is that when subjects are stressed, their alertness may also be increased [17]. In situations where the stress is caused by the imposition of a time limit (such as in our study), this alertness discourages distractions and encourages more focus on the task at hand. A similar kind of gaze and mouse coordination has also been found when a user is in a state with a high cognitive load by [26, 48].

5.2 Realistic Study: Moving to Webcam-captured Gaze Locations

The above experimental results demonstrate that mental stress can be detected efficiently via mouse and gaze behaviors, when accurate gaze locations are available from the eye tracker. We then move to a more realistic context where an eye tracker is not available, and the gaze locations estimated from webcam video.

We first evaluate the performance of our gaze location estimation method described in Section 3.3 using leave-one-subject-out cross-validation. The average error in pixels of estimated gaze locations among all subjects is around 125 px on a 1600×1000-pixel screen and detailed average errors for each subject are shown in Figure 11. We then evaluate the performance of stress detection based on the

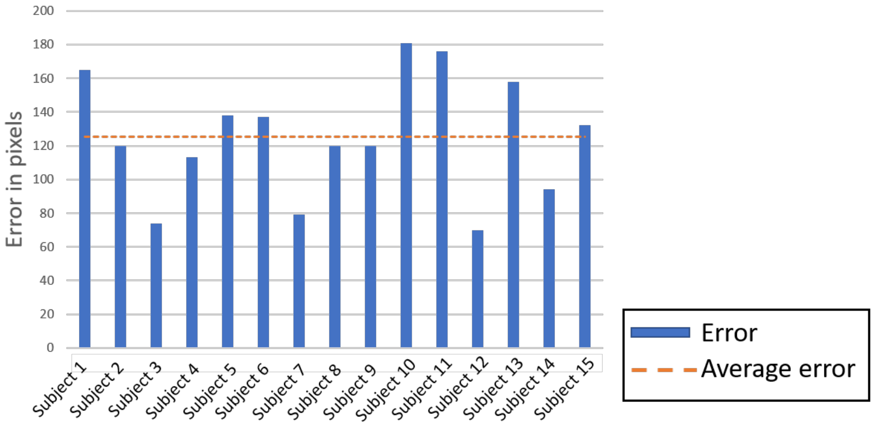


Fig. 11. Average Error of Webcam-based Gaze Estimation for Each Subject.

estimated gaze locations. Again, this is evaluated by the leave-one-subject-out mechanism. Table 9 presents the results of stress detection based on the estimated eye gaze locations. The confusion matrix is presented in Table 10. The overall CCR for stress detection based on the estimated gaze locations is 65.1%, a drop of around 14% from the performance achieved when the eye-tracker is used to capture the eye gaze locations.

Table 9. Classification Performance for Stress Detection via Mouse and Gaze Behaviors (webcam-based).

Performance / **Class**	Precision	Recall	F-measure
Relax	0.64	0.65	0.64
Stress	0.66	0.66	0.66
Weighted Average	0.65	0.65	0.65

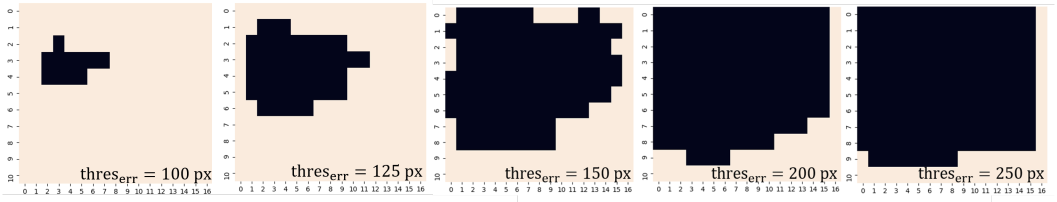
Table 10. Confusion Matrix for Stress Detection via Mouse and Gaze Behaviors (webcam-based).

Predicted as / **Ground truth**	Relax	Stress	Total
Relax	**55**	30	85
Stress	31	**59**	90
Total	86	89	175

The average error of the estimated eye gaze locations is about 125 pixels on a 1600 × 1000-pixel screen, but this degradation in performance may or may not be uniform across the entire screen surface. We therefore conduct a deeper analysis to investigate the relationship between the CCR performance in different screen regions and the inherent error in the estimated eye gaze locations. We first evenly divide the whole screen area into 16 × 10 sub-areas, where each sub-area contains around 100 × 100 pixels. For a given sub-area j, we then compute the average error of estimated gaze locations within it ($error^j$) based on the following equation.

$$error^j = \frac{\sum_i^{C_j} \sqrt{\left(P_{GT}^{(i)} - P_{GE}^{(i)}\right)^2}}{\left|C_j\right|} \tag{7}$$

where C_j is the set of gaze locations P_{GT} collected by the eye-tracker within the sub-area j, $P_{GE}^{(i)}$ is the corresponding estimated gaze of $P_{GT}^{(i)}$ and $|\cdot|$ returns the size of the set. We then set a threshold ($thres_{err}$) and classify all the sub-areas into two categories: those which exhibit an average error greater than $thres_{err}$, and those which exhibit an error less than $thres_{err}$. We refer to the two categories as *HighError* and *LowError* respectively.



(a) (b) (c) (d) (e)

Fig. 12. Error Analysis with respect to screen regions. The average estimated gaze location error is smaller than $thres_{err}$ in the black areas (i.e., *LowError* areas).

Figure 12 shows the results of our analysis based on setting $thres_{err}$ with different values. The region in black shows the area in which the average estimated gaze location error is smaller than the threshold (*LowError* areas). It can be seen that the estimated gaze locations in the central area of the screen are more accurate (having lower estimation error) compared to locations at the edge of the screen. An immediate question would be "*What if we discard the gaze locates beyond the LowError area*".

To address that, we then build and evaluate separate models that specialize in handling user behaviors in particular regions of the screen. We refer to these models as *Region-Specific* models. For a given $thres_{err}$, the *Region-Specific* model makes use of only the gaze locates within the corresponding *LowError* area to compute MGAttraction features for stress detection. We follow the same approaches to extract our MGAttraction features as presented in Section 3. The only difference is that any gaze and the related features beyond the *LowError* area will be discarded. Particularly, we construct four *Region-Specific* models, one for each of the following $thres_{err}$: 125px, 150px, 200px, 250px. The case of $thres_{err} = 100$ is excluded because of the extremely small size of its *LowError* area (Fig. 12 (a)), which results in no much valid data can be used. We further compare their performance with the model using signals from the full screen, which can be regarded as an extreme case of *Region-Specific* model with $thres_{err} = \infty$. Again, leave-one-subject-out mechanism is used to evaluate the model performance. The results are shown in Table 11.

It is worth noting that our *Region-Specific* webcam-based stress detection models generally outperform the model without region-specific. For $thre_{err} = 125$, there are no adequate gaze data can be used for the relatively small size of the *LowError* area, impacting the model on achieving higher accuracy. Our *Region-Specific* adaption restricts the model to only look at "reliable" gaze patterns, it thus helps in attaining more acceptable accuracy of stress detection. This also demonstrates that our webcam-based stress detection model can perform better when the gaze estimation is more accurate.

Table 11. Performance of *Region-Specific* Webcam-based Stress Detection with different $thres_{err}$.

Performance / $thres_{err}$	Precision (stress)	Recall (stress)	F-measure (stress)	Accuracy
125 px	0.64	0.58	0.61	64.5%
150 px	0.70	0.69	0.69	68.6%
200 px	0.67	0.72	0.70	67.4%
250 px	0.67	0.66	0.66	65.7%
no $thres_{err}$	0.66	0.66	0.66	65.1%

$*$ $thres_{err} = 100$ is not included since the size of the *LowError* area is too small.

Table 12 summarizes the evaluation results of eye-tracker-based and webcam-based methods for comparison. Table 13 presents the confusion matrix of our *Region-Specific* webcam-based model with $thres_{err} = 150$, which achieve the best performance among the *Region-Specific* models. Results show that, with the help of the *Region-Specific* adaption, the webcam-based model achieves an improvement, which brings its performance closer to the eye-tracker based model, but without the need for any special equipment.

Table 12. Performance of different approaches in dynamic-UI task.

Performance / **Model**	CCR	Precision (Stress)	Recall (Stress)
Eye-tracker based	78.8%	0.77	0.84
Webcam-based	65.1%	0.66	0.66
Webcam-based (*Region-Specific*)	68.6 %	0.70	0.69

Table 13. Confusion Matrix for *Region-Specific* Webcam-based Model ($thre_{err}$ = 150).

Predicted as / **Ground truth**	Relax	Stress	Total
Relax	**58**	27	85
Stress	28	**62**	90
Total	86	89	175

6 DISCUSSION

The experimental results demonstrate the feasibility of detecting stress on a common user interaction task based on the mouse and gaze behaviors in a dynamic-UI environment. In this section, we discuss the findings we achieved, the imagined deployment of the proposed method for real-time and long-term monitoring in practices, and the potential limitation and future work.

6.1 Findings and Discussion

Unlike the majority of other state-of-the-art approaches, which model the correlation between the mouse or gaze and the UI components separately, our approach considers a projection of both mouse and gaze information into a MGAttraction coordinate system that is translation and rotation invariant. This alleviates the need for accurate detection/identification of UI components, which usually requires image processing techniques and is computationally expensive. Our MGAttraction coordinate system also allows for interpretation of the mouse/gaze behaviors in physical terms – in other words, the tendency of mouse and gaze to approach to or depart from each other. This allows qualitative investigation of behaviors that are more important for stress detection. Our observations suggest that when subjects are stressed, they tend to be more focused on their task and exhibit more consistent gaze movement, more intensity of mouse movement and less time latency between mouse and gaze. We believe the MGAttraction coordinate system would benefit future studies in human computer interaction area and related intelligent user interface development.

Our approach achieves the best performance compared with other state-of-the-art stress detection approaches. StressClick [26] is the system that is closest to ours. However, StressClick only considers the movement of the gaze before and after each mouse click. Our approach expands upon theirs by considering both mouse and gaze behaviors during the entire session. Our first finding shows that considering both mouse and gaze information can build a better-performing stress detection model. It also suggests that features extracted in a single modality within a short time-window may not be powerful enough to detect stress in a more open-ended and complex task.

The second part of our study explores the feasibility of using a webcam-based system to estimate gaze locations on the screen, which makes our system more feasible for consumer applications than other approaches which rely on the use of a specialized eye tracker. We find that the accuracy of gaze estimation strongly impacts the performance of the stress detection. We also find that the accuracy of the webcam-based gaze estimation varies according to the position of gaze on the screen, with some regions (such as the upper-middle part of the screen when the webcam is placed on top of the monitor in the center) exhibiting more accurate gaze estimations. Both findings suggest that it may be beneficial for UI designers to better exploit this area of the screen since gaze information in that area can be estimated with higher confidence just using a standard webcam. Our final webcam-based stress detection model considers modeling the mouse and gaze movements/coordination within a region-specific – discarding the gaze and the related features if it locates beyond the “reliable” region. This approach helps to improve the model performance.

6.2 Model Deployment in Practices

Given the promising performance of our webcam-based stress detection, our method can be deployed to any personal computers equipped with a webcam and a mouse, without using any special devices. Besides, the method works independently of the UI environment. Specifically, it uses MGAttraction model to capture the coordination between mouse and gaze, that takes into account the relative movement between the two modalities such as “gaze leading mouse”, regardless their absolute locations and relations to the UI components in the screen. Hence, it can be deployed to any UI environments.

Our experimental results demonstrate the efficiency of the proposed method in session-based stress detection (e.g., detecting for a 5-minute session). We also notice that the whole detection, including MGAttraction computation, feature extraction and model output, can be completed instantly upon receiving a session of data. In particular, the time complexity of the proposed MGAttraction computation and feature extraction method is O(n). We anticipate that our method can be deployed for real-time and long-term monitoring under periodical jumping window-based detection mechanism. For example, it detects and logs users’ stress state periodically every 5 minutes based on the mouse and gaze signals that have occurred in the past 5 minutes, when deployed to real-world applications, – periodically taking every 5 minutes as a session for the input to our method. Users’ stress state can then be long-termly and real-timely monitored. The overall stress state during a long period can be summarized by voting ensemble.

6.3 Limitation and Future work

We can achieve better webcam-based stress detection when accurate webcam-based gaze estimation is attained. To further understand the webcam-based gaze estimation performance, we particularly look into the effects of wearing and not wearing glasses. Fig. 13 illustrates the gaze estimation error for each subject in ascending order, in which subjects with and without glasses are indicated with different colors. In particular, subject 7, 11, and 14 are the ones who do not wear glasses. According to the figure, we observe a marginally better gaze estimation performance (i.e., slightly lower average error) for subjects without glasses. The average error for subjects wearing and not wearing glasses are about 127 and 116 respectively. For comparison, the average error of all subjects is around 125. Nevertheless, given that there are only three subjects who do not wear glasses in our dataset, we cannot yet draw statistical conclusions on the effect of wearing and not wearing glasses on our webcam-based gaze estimation and stress detection performance. We also notice that subjects who have the high gaze estimation error often had bad posture while participating in the data collection. For example, both subject 1 and 10 tended to lower their heads in the experiments, while subject 11 often lifted the head. This exposes the potential impact of subjects' posture on the performance of our webcam-based method. One of our future directions is to further boost the webcam-based method, including the understanding and eliminating the effects of different factors, such as wearing glasses and having bad postures.

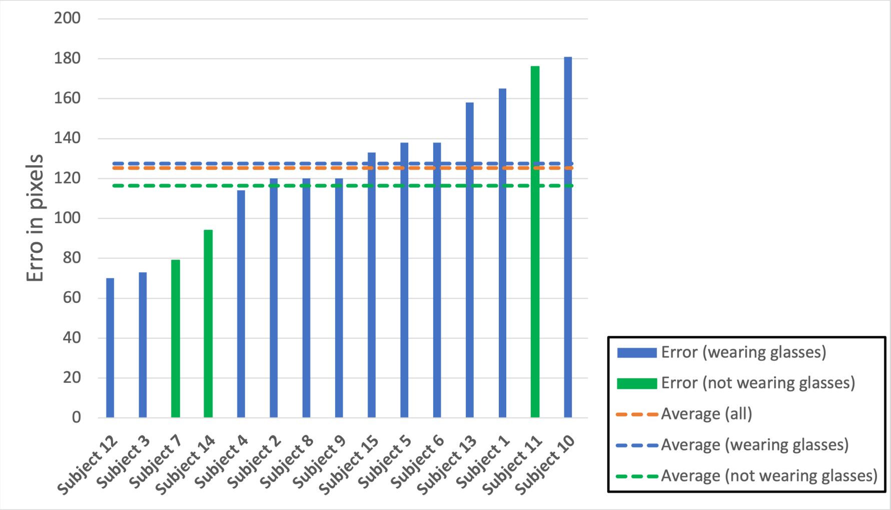


Fig. 13. Average Error of Webcam-based Gaze Estimation for Each Subject in Ascending Order.

This paper mainly focuses on exploring a novel mouse and gaze coordination-based stress detection method that can work in actual workplace tasks with dynamic-UI. As the very first step in this direction, we conduct the study based on hand-crafted features. On one hand, it is difficult and time-consuming to collect a large-scale dataset that involves great amounts of real human subjects in real-life tasks for training efficient end-to-end deep learning models. On the other hand, we are also interested in the intensive understanding of human mouse and gaze behaviors in daily human-computer interaction tasks under mental stress. The investigation of hand-crafted features with the interpreting of their physical meanings (e.g., the hand-eye coordination) is an essential step, before we can reach to an effective way to construct end-to-end model in this domain. We will keep enlarging our dataset by recruiting more subjects from diverse groups. This includes subjects with and without glasses, subjects from different age groups, and subjects with different cultures and backgrounds. We will especially try end-to-end deep learning methods for constructing both webcam-based gaze estmation and stress detection models, when a large-scale dataset is obtained. More comprehensive study will also be conducted to investigate the impact of different factors on workplace stress detection performance.

7 CONCLUSION

This paper proposes an innovative coordinate system, MGAttraction, that measures the mouse and gaze attraction, reflected by their relative movement, in a translation- and rotation-invariant manner. By utilizing the MGAttraction coordinate system, mouse and gaze behaviors can be modeled without relying on any UI information. An UI-agnostic stress detection method is further proposed based on MGAttraction. Our method is evaluated on a real-world task with dynamic-UI environments: web searching. Our stress detection method achieves the accuracy of 78.8%, beating the state-of-the-art approach by around 20%. The experimental results reveal that individuals' mental stress level can be modeled from their mouse (hand) and gaze (eye) coordination, shedding lights on workplace mental stress detection.

To further improve our method's applicability without relying on any special equipment such as the eye-tracker, we use a standard webcam to estimate the gaze locations for our MGAttraction computation and stress detection. Our webcam-based stress detection method yields an accuracy of 68.6%, which approaches the performance obtained using a specialized eye tracker. The performance gap can be further narrowed by boosting webcam-based gaze estimation method, adding other visual features, and applying deep learning methods. This will be one of our future focuses, to eventually develop practical intelligent applications for workplace mental health monitoring.

ACKNOWLEDGMENTS

This work was supported in part by the Hong Kong Research Grant Council and the Hong Kong Polytechnic University under Grants 15600219 and P0039489.

REFERENCES

[1] Mohamed Abouelenien, Mihai Burzo, and Rada Mihalcea. 2016. Human acute stress detection via integration of physiological signals and thermal imaging. In *Proceedings of the 9th ACM International Conference on PErvasive Technologies Related to Assistive Environments*. 1–8.

[2] Serdar Baltaci and Didem Gokcay. 2016. Stress detection in human–computer interaction: Fusion of pupil dilation and facial temperature features. *International Journal of Human–Computer Interaction* 32, 12 (2016), 956–966.

[3] Tadas Baltrusaitis, Amir Zadeh, Yao Chong Lim, and Louis-Philippe Morency. 2018. Openface 2.0: Facial behavior analysis toolkit. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*. IEEE, 59–66.

[4] Armando Barreto, Jing Zhai, and Malek Adjouadi. 2007. Non-intrusive physiological monitoring for automated stress detection in human-computer interaction. In *International Workshop on Human-Computer Interaction*. Springer, 29–38.

[5] Hans-Joachim Bieg, Lewis L Chuang, Roland W Fleming, Harald Reiterer, and Heinrich H Bülthoff. 2010. Eye and pointer coordination in search and selection tasks. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*. 89–92.

[6] Nigel Bosch, Sidney D'Mello, Ryan Baker, Jaclyn Ocumpaugh, Valerie Shute, Matthew Ventura, Lubin Wang, and Weinan Zhao. 2015. Automatic detection of learning-centered affective states in the wild. In *Proceedings of the 20th international conference on intelligent user interfaces*. 379–388.

[7] Ricardo Buettner. 2018. Robust user identification based on facial action units unaffected by users' emotions. (2018).

[8] Monchu Chen and Veraneka Lim. 2013. Eye gaze and mouse cursor relationship in a debugging task. In *International Conference on Human-Computer Interaction*. Springer, 468–472.

[9] Siyuan Chen, Julien Epps, Natalie Ruiz, and Fang Chen. 2011. Eye activity as a measure of human mental effort in HCI. In *Proceedings of the 16th international conference on Intelligent user interfaces*. 315–318.

[10] Matteo Ciman, Katarzyna Wac, and Ombretta Gaggi. 2015. iSenseStress: Assessing stress through human-smartphone interaction analysis. In *2015 9th International Conference on Pervasive Computing Technologies for Healthcare (Pervasive-Health)*. IEEE, 84–91.

[11] J Douglas Crawford, W Pieter Medendorp, and Jonathan J Marotta. 2004. Spatial transformations for eye–hand coordination. *Journal of neurophysiology* (2004).

[12] Mohamed Dahmane, Jahangir Alam, Pierre-Luc St-Charles, Marc Lalonde, Kevin Heffner, and Samuel Foucher. 2020. A multimodal non-intrusive stress monitoring from the pleasure-arousal emotional dimensions. *IEEE Transactions on Affective Computing* 13, 2 (2020), 1044–1056.

[13] Alberto de Santos Sierra, Carmen Sánchez Ávila, Javier Guerra Casanova, and Gonzalo Bailador del Pozo. 2011. A stress-detection system based on physiological signals and fuzzy logic. *IEEE Transactions on Industrial Electronics* 58, 10 (2011), 4857–4865.

[14] Laurence Devillers and Laurence Vidrascu. 2007. Real-life emotion recognition in speech. *Speaker classification II: Selected projects* (2007), 34–42.

[15] Damodar Reddy Edla, Kunal Mangalorekar, Gauri Dhavalikar, and Shubham Dodia. 2018. Classification of EEG data for human mental state analysis using Random Forest Classifier. *Procedia computer science* 132 (2018), 1523–1532.

[16] Eugene Yujun Fu, Tiffany CK Kwok, Erin You Wu, Hong Va Leong, Grace Ngai, and Stephen CF Chan. 2017. Your mouse reveals your next activity: towards predicting user intention from mouse interaction. In *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, Vol. 1. IEEE, 869–874.

[17] Edith Galy, Magali Cariou, and Claudine Mélan. 2012. What is the relationship between mental workload factors and cognitive load types? *International Journal of Psychophysiology* 83, 3 (2012), 269–275.

[18] Giorgos Giannakakis, Dimitris Grigoriadis, Katerina Giannakaki, Olympia Simantiraki, Alexandros Roniotis, and Manolis Tsiknakis. 2019. Review on psychological stress detection using biosignals. *IEEE Transactions on Affective Computing* 13, 1 (2019), 440–460.

[19] Agostino Gibaldi, Mauricio Vanegas, Peter J Bex, and Guido Maiello. 2017. Evaluation of the Tobii EyeX Eye tracking controller and Matlab toolkit for research. *Behavior research methods* 49, 3 (2017), 923–946.

[20] Martijn Haak, Steven Bos, Sacha Panic, and LJM Rothkrantz. 2009. Detecting stress using eye blinks and brain activity from EEG signals. *Proceeding of the 1st driver car interaction and interface (DCII 2008)* (2009), 35–60.

[21] Jennifer A Healey and Rosalind W Picard. 2005. Detecting stress during real-world driving tasks using physiological sensors. *IEEE Transactions on intelligent transportation systems* 6, 2 (2005), 156–166.

[22] Javier Hernandez, Pablo Paredes, Asta Roseway, and Mary Czerwinski. 2014. Under pressure: sensing stress of computer users. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 51–60.

[23] Bruce D Homer, Jan L Plass, and Linda Blake. 2008. The effects of video on cognitive load and social presence in multimedia-learning. *Computers in Human Behavior* 24, 3 (2008), 786–797.

[24] Jeff Huang, Ryen W White, and Susan Dumais. 2011. No clicks, no problem: using cursor movements to understand and improve search. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 1225–1234.

[25] Michael Xuelin Huang, Tiffany CK Kwok, Grace Ngai, Stephen CF Chan, and Hong Va Leong. 2016. Building a personalized, auto-calibrating eye tracker from user interactions. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 5169–5179.

[26] Michael Xuelin Huang, Jiajia Li, Grace Ngai, and Hong Va Leong. 2016. Stressclick: Sensing stress from gaze-click patterns. In *Proceedings of the 24th ACM international conference on Multimedia*. 1395–1404.

[27] Albrecht W Inhoff and Andrew M Gordon. 1997. Eye movements and eye-hand coordination during typing. *Current Directions in Psychological Science* 6, 6 (1997), 153–157.

[28] Saskia Koldijk, Maya Sappelli, Suzan Verberne, Mark A Neerincx, and Wessel Kraaij. 2014. The swell knowledge work dataset for stress and user modeling research. In *Proceedings of the 16th international conference on multimodal interaction*. 291–298.

[29] Richard S Lazarus. 1966. Psychological stress and the coping process. (1966).

[30] Richard S Lazarus. 1993. From psychological stress to the emotions: A history of changing outlooks. *Annual review of psychology* 44, 1 (1993), 1–22.

[31] Jiajia Li, Grace Ngai, Hong Va Leong, and Stephen CF Chan. 2016. Multimodal human attention detection for reading from facial expression, eye gaze, and mouse dynamics. *ACM SIGAPP Applied Computing Review* 16, 3 (2016), 37–49.

[32] Russell Li and Zhandong Liu. 2020. Stress detection using deep neural networks. *BMC Medical Informatics and Decision Making* 20 (2020), 1–10.

[33] Daniel J Liebling and Susan T Dumais. 2014. Gaze and mouse coordination in everyday work. In *Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing: adjunct publication*. 1141–1150.

[34] Yongqiang Lyu, Xiaomin Luo, Jun Zhou, Chun Yu, Congcong Miao, Tong Wang, Yuanchun Shi, and Ken-ichi Kameyama. 2015. Measuring photoplethysmogram-based stress-induced vascular response index to assess cognitive load and stress. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*. 857–866.

[35] Pablo E Paredes, Francisco Ordonez, Wendy Ju, and James A Landay. 2018. Fast & furious: detecting stress with a car steering wheel. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–12.

[36] Peng Ren, Armando Barreto, Ying Gao, and Malek Adjouadi. 2012. Affective assessment by digital processing of the pupil diameter. *IEEE Transactions on Affective computing* 4, 1 (2012), 2–14.

[37] Kerry Rodden, Xin Fu, Anne Aula, and Ian Spiro. 2008. Eye-mouse coordination patterns on web search results pages. In *CHI'08 extended abstracts on Human factors in computing systems*. 2997–3002.

[38] Philip Schmidt, Attila Reiss, Robert Duerichen, Claus Marberger, and Kristof Van Laerhoven. 2018. Introducing wesad, a multimodal dataset for wearable stress and affect detection. In *Proceedings of the 20th ACM international conference on multimodal interaction*. 400–408.

[39] Peter L Schnall, Paul A Landsbergis, and Dean Baker. 1994. Job strain and cardiovascular disease. *Annual review of public health* 15, 1 (1994), 381–411.

[40] Nandita Sharma and Tom Gedeon. 2012. Objective measures, sensors and computational techniques for stress recognition and classification: A survey. *Computer methods and programs in biomedicine* 108, 3 (2012), 1287–1301.

[41] Dave M Stampe. 1993. Heuristic filtering and reliable calibration methods for video-based pupil-tracking systems. *Behavior Research Methods, Instruments, & Computers* 25, 2 (1993), 137–142.

[42] David Sun, Pablo Paredes, and John Canny. 2014. MouStress: detecting stress from mouse motion. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 61–70.

[43] Feng-Tso Sun, Cynthia Kuo, Heng-Tze Cheng, Senaka Buthpitiya, Patricia Collins, and Martin Griss. 2010. Activity-aware mental stress detection using physiological sensors. In *International conference on Mobile computing, applications, and services*. Springer, 282–301.

[44] Carla Viegas, Shing-Hon Lau, Roy Maxion, and Alexander Hauptmann. 2018. Distinction of stress and non-stress tasks using facial action units. In *Proceedings of the 20th International Conference on Multimodal Interaction: Adjunct*. 1–6.

[45] Kim-Phuong L Vu and Robert W Proctor. 2011. *Handbook of human factors in Web design*. CRC Press.

[46] Johannes Wagner, Jonghwa Kim, and Elisabeth André. 2005. From physiological signals to emotions: Implementing and comparing selected methods for feature extraction and classification. In *2005 IEEE international conference on multimedia and expo*. IEEE, 940–943.

[47] Jun Wang, Eugene Yujun Fu, Grace Ngai, and Hong Va Leong. 2019. Investigating Differences in Gaze and Typing Behavior Across Age Groups and Writing Genres. In *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, Vol. 1. IEEE, 622–629.

[48] Jun Wang, Michael Xuelin Huang, Grace Ngai, and Hong Va Leong. 2017. Are you stressed? Your eyes and the mouse can tell. In *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)*. IEEE, 222–228.

[49] Pierre Weill-Tessier, Jayson Turner, and Hans Gellersen. 2016. How do you look at what you touch? A study of touch interaction and gaze correlation on tablets. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*. 329–330.

[50] Takashi Yamauchi. 2013. Mouse trajectories and state anxiety: feature selection with random forest. In *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*. IEEE, 399–404.

[51] Xian Zhao, Zhan Song, Jian Guo, Yanguo Zhao, and Feng Zheng. 2012. Real-time hand gesture detection and recognition by random forest. In *Communications and information processing*. Springer, 747–755.