7.3. Mokslinis straipsnis: Modelling Eye Fatigue in Gaze Spelling Task

Modelling Eye Fatigue in Gaze Spelling Task

M. Vasiljevas, T. Gedminas, A. Ševčenko, M. Jančiukas, T. Blažauskas, R. Damaševičius

Department of Software Engineering,
Kaunas University of Technology,
Studentu 50, Kaunas Lithuania
Email: robertas.damasevicius@ktu.lt

***Abstract*—The paper analyses analytical models of fatigue and their applicability to natural user interfaces (NUI). For evaluation of eye fatigue we use a gaze speller as an example of NUI system for text input using human gaze and eye tracking device. We use the system to evaluate the state and performance of the users in performing text entry task as well as to analyse dynamical changes of user's state (such as induced by fatigue) over time. The results suggest validity of classical model of muscle fatigue suggested in the sports medicine domain by Banister et al.**

***Keywords*—natural user interface, gaze tracking, eye fatigue.**

I. Introduction

Physiological computing systems (PCS) use physiological data of its users as input for performing computational tasks [1]. Data is collected using physiological data sensors (e.g., electroencephalography (EEG), electrooculography (EOG), electromyography (EMG), etc.) attached to the user's body. Sensors directly measure signals triggered by the events occurring within the human body. Such input represents an additional channel of communication [2] between a user and a computer that allows a PCS to monitor, detect and respond appropriately to the physical, emotional and cognitive modalities of its users. Since a PCS monitors the user's state, it creates a *bio-cybernetic feedback loop* [3] that allows to produce a more accurate representation of the user's context using information received from sensors attached to the human body. Such loop represents an innovative mode of human-computer interaction (HCI) where interaction is achieved by observing, analysing and responding to psychophysiological activity of a user [4]. The increase of the number of HCI modes beyond common mechanical and tactile ones [1] promises extended capabilities of using computers for disabled users [5], as well as increasing engagement and immersion of common computer users (e.g., in computer games [6]), and increasing general intelligence of computer systems in smart environments such as Ambient Assisted Living Environments (AALE). At the heart of AALE are assistive technologies that can be used by people with disabilities to perform functions that might otherwise be difficult or impossible for them to do.

One of such assistive technologies is gaze (eye) tracking, i.e., the tracking eye movements or the point of gaze (POG). Gaze tracking is useful in a broad range of application areas, starting from medical diagnostic to usability studies and gaze-controlled software applications for human computer interaction, e-learning, psychology investigation, virtual and augmented reality, etc. Gaze interfaces can be defined as a kind of Natural User Interface (NUI) [7]. NUIs promise to introduce more natural ways of human computer interaction (HCI) beyond traditional mechanics based interfaces devices such as keyboard or mouse. Examples of NUIs include Brain-Computer Interfaces (BCI), Neural Computer Interfaces (NCI), tangible user interfaces (TUI), gesture interfaces and gaze interfaces. It is expected that NUIs will allow taking full advantage of alternative interaction modalities thus providing new levels of expression and immersions in HCI while liberating from frustration and rigidness of the mechanics based interaction and bridging the digital divide. However, conscious use of NUIs is hindered by many factors such as the use of low-cost high-quality sensors and the need to develop more effective physiological signal processsing and user state detection algorithms as well as steep learning curve for using NUIs. One of the factors often neglected is fatigue (both mental and muscular one) which reduces accuracy and information transfer rate of the NUI communication channel and leads to increased input errors.

Fatigue is a gradually increasing subjective feeling of tiredness of a subject under load. Fatigue can have physical or mental causes. Fatigue can be manifested in a number of different ways. For example, fatigue is considered to result from working, mental stress, boredom, disease, and lack of sleep [9]. Physical fatigue is the inability of a muscle to maintain optimal physical performance [9]. For any computer systems which use displays (including gaze tracking based systems), eye fatigue is a serious problem that results from a prolonged use of computers. It affects up to 70% of computer users [10]. Mental fatigue is a decrease in maximal cognitive performance resulting from prolonged periods of cognitive activity [11]. Mental fatigue leads to deterioration of cognitive functions as due to deterioration of cognitive functions user responses become slower and more error prone [12]. The result of fatigue is decreased productivity of using a PCS as well as critical errors in worst cases. Despite the importance of the fatigue factor there is little research of how to detect, measure and use it as factor for adapting the behaviour of a PCS. For example, Lanthier *et al.* [8] has related increase in fixation duration (dwell time) to fatigue, Abdulin and Komogortsev [13] claimed that increasing distance between the fixation components of the eye movement signal and the stimulus indicate the onset of fatigue, and McKinley *et al.* [14] related reductions in complexity or irregularity of the eye movements evaluated using approximate entropy in conjunction with total eye closure duration as a predictor of fatigue.

A poor-designed UI can make physical eye workload higher due to, e.g., using an excessive visual search, while a better-designed GUI could alleviate or postpone the onset of the symptoms of eye fatigue. By being able to detect user eye fatigue while navigating the gaze tracking based interface, it becomes possible to react to the development of fatigue by alerting users to make a break in their work as well as to aid in developing more user-friendly and more usable interfaces, which would reduce levels of physical eye fatigue.

In this paper we analyse the known analytical models of fatigue and their applicability to gaze-based NUIs. For evaluation of fatigue we use a gaze speller as an example of PC based NUI that uses eye gaze positions as input. We use the system to evaluate the state and performance of the users in performing control operation as well as to research dynamical changes of the user's state (such as induced by fatigue) over time. The structure of the remaining parts of the paper is as follows. In Section II we analyse the mathematical (analytical) models of fatigue. n Section III we describe the proposed model to evaluate subject fatigue during gaze tracking tasks. In Section IV we describe the developed gaze speller application. In Section V we describe the experimental setting. In Section VI we present and analyze the results. Finally, in Section VII we present conclusions and discuss future work.

II. Overview of Analytical Fatigue Models

In this study, the decrease in gaze tracking performance is considered as the result of eye muscle fatigue. Conversely, the improvement in performance is attributed to a learning effect. Mathematical and analytical models provide a method for describing and predicting the effect of mental and muscular load on the performance characteristics of a human [15]. Analysing physical performance data one can identify and quantify different effects of loads such as increased performance (fitness or learning) and decreased performance (fatigue). Such models have been extensively researched in sports medicine [16, 17] and in applied physiology [18, 19].

A. *Models of muscular fatigue*

The Banister model [16] and its elaborations [17-19] is aimed to relate training loads to performance, taking into account dynamic and temporal characteristics of load sequences over time. These effects are described by two transfer functions: 1) a positive influence (i.e., learning or fitness) that sums up all positive effects leading to an increase in performance, and 2) a negative function that summarizes effects leading to fatigue and decrease in performance:

$$p_t = p_0 + k_a \sum_{s=0}^{t-1} e^{-(t-s)/t_a} w_s - k_f \sum_{s=0}^{t-1} e^{-(t-s)/t_f} w_s \quad (1)$$

here p_t is the modelled performance at time t; p_0 is the initial performance; k_a and k_f are the fitness and fatigue magnitude factors; τ_a and τ_f are fitness and fatigue decay time constants; and w_s is the training load.

Calvert *et al.* [17] have proposed a model to quantify the training and performance relationship of a swimmer. The model was derived by fitting a mathematical equation and examining its parameters. A single training impulse elicited two fitness responses that increase performance, and a fatigue response that decreases performance as follows:

$$p(t) = \left[e^{\frac{-t}{\tau_1}} - e^{\frac{-t}{\tau_2}} - Ke^{\frac{-t}{\tau_3}} \right] w(t) \quad (2)$$

here $w(t)$ is the training impulse, $p(t)$ is performance, τ_1 and τ_2 are the time constants associated with the two fitness functions and τ_3 is the time constant associated with fatigue, and K is the fatigue coefficient specific to the individual, and t is the day of the training impulse.

Morton *et al.* [18] simplified the Calvert's model (Eq. 2) to two components (one for fitness and one for fatigue):

$$p(t) = \left[e^{\frac{-t}{\tau_1}} - Ke^{\frac{-t}{\tau_3}} \right] w(t) \quad (3)$$

Busso *et al.* [19] defined how fitness and fatigue are affected by a training input as follows:

$$g(t) = g(t-1)e^{\frac{-1}{\tau_1}} + w(t) \; ; \; h(t) = h(t-1)e^{\frac{-1}{\tau_2}} + w(t) \quad (4)$$

here $g(t)$ and $h(t)$ are arbitrary fitness and fatigue response levels at the end of day t, τ_1 and τ_2 are decay constants.

Eq. (4) were combined to form a simple linear equation

$$p(t) = k_1 g(t) - k_2 h(t) \quad (5)$$

here k_1 and k_2 are weighting factors for fitness and fatigue.

B. *Metrics of eye fatigue*

Eye movement refers to the voluntary or involuntary movement of the eyes, helping in acquiring, fixating and tracking visual stimuli. Humans use three types of voluntary eye movement to track objects of interest: smooth pursuit, vergence movements and saccades. McConkie [20] has demonstrated that the distributions of initial saccade landing sites are Gaussian in shape and that the centre of these distributions and their standard deviations are determined primarily by oculomotor factors. Variability in human saccades is caused by a combination of uncertainty in target localization and noise in movement planning and execution [21]. As mental as well as physical eye affects both target localization and movement execution, the onset of fatigue should lead to higher variability in saccadic movements and target fixation positions.

The evaluation of eye fatigue is usually performed using subjective evaluation questionnaires, e.g., Majaranta *et al.* [22] asked the participants how tired their eyes were before each test, and again after the test, on a scale of 1 to 7. The fatigue level was calculated by subtracting the first value from the last value. However, the authors have not managed to obtain any relationship of the level of fatigue vs. time or speed of work. It points to the unreliability of the method of evaluation used.

Dwell time is the duration a gaze fixation rests on a certain object. Dwell time helps to differentiate between accidental gazes, gazes during visual search and, e.g., intentional gazes during execution of tasks. The duration of a fixation correlates to the processing that is going on in the brain. It is expected that dwell time should increase due to fatigue.

Point of gaze (POG) accuracy. Given the target, we can compute the distance from the centre of the target during each fixation when the eyes are aligned with our target of visual attention. The fixated area is called the point of gaze. Due to fatigue, the POG accuracy should decrease.

Fatigue Threshold (TF) [23] is calculated using an empirical formula that depends on the average spatial accuracy in degrees, θ_{avg}, of the eye tracker so that the threshold can scale with noisier signals, A, difference in FQlS between the first fatigued group of data and the initial FQlS, and μ, the mean spatial accuracy of the data:

$$TF = A \times \theta_{avg} / \mu \tag{6}$$

Average spatial accuracy [23] is calculated during calibration by finding the mean gaze point, Gi, for each calibration point, Pi, and then calculating the average distance in degrees, θi, between each calibration point and gaze point:

$$\theta_{acc} = \frac{1}{n} \sum_{i=1}^{n} |P_i - G_i| \tag{7}$$

Fixation Qualitative Score (FQlS) is a metric that represents the distance between the fixation components of the eye movement signal and the stimulus [24].

$$FQlS = \frac{1}{n} \sum |p_i - g_i| \tag{8}$$

here p_i are stimuli points and g_i are the gaze points. FQlS should increase when a user becomes fatigued.

C. Gaze spelling related metrics

Metrics include types of metrics for evaluating typing characteristics (input accuracy, error rate, information transfer rate) [25] related to text entry task using gaze ('gaze spelling').

Typing speed is measured in words per minute (wpm), where a word is any sequence of five characters, including letters, spaces, punctuation, etc. [26]. Both corrected errors and errors left in the entered text are taken into account.

$$WPM = \frac{|T| - 1}{S} \times 60 \times \frac{1}{5} \tag{9}$$

Here, S is time in seconds measured from the first key press to the last, including backspaces and other edit and modifier keys. The constant 60 is the number of seconds per minute, and the factor of one fifth accounts for the average length of a word in characters including spaces, numbers, and other printable characters. Note that time is measured from the entry of the very first character to the last, which means that the entry of the first character is never timed. It is expected that due to fatigue typing speed should decrease.

Error Rate (ER) is calculated as the ratio of the total number of incorrect characters in the transcribed text to the length of the transcribed text:

$$ER = \frac{|E|}{|T|} \times 100\% \tag{10}$$

here E is the number of errors in the text T. This metric does not take into account corrected errors. It is expected that due to fatigue the error rate should increase.

Rate of Backspacing (BR) [24 .indicates how often the participants cancelled characters and correlates with errors to a degree. The rate of backspacing is calculated by dividing the total number of characters erased prior to the current position by the total number of characters typed:

$$BR = \frac{|B|}{|T|} \times 100\% \tag{11}$$

here B is the number of corrections in the text T. It is expected that due to fatigue BR should increase.

Total Error Rate (TER) combines the effect of accuracy during and after text entry:

$$TER = \frac{|E| + |B|}{|T| + |E| + |B|} \times 100\% \tag{12}$$

D. Summary

Summarizing, the fatigue models are based on the exponential decay function that is widely used to describe natural phenomena such as heat transfer between the object and its medium, rate of enzyme-catalysed chemical reactions, fluid dynamics, etc. These examples provide a logical foundation for application of exponential decay function for fatigue modelling. However the fatigue models also have been criticized for imprecision and low accuracy due to variability of their parameters [27]. Furthermore, the models also need verification in the context of PCS, where signals of human body are usually registered under normal conditions rather than considerable strain. Also the problem of mental fatigue is usually ignored, though over time it leads to decrease of performance in PCS.

III. Proposed Eye Fatigue Model

Let $\underline{X}$ be a time series consisting of spatial gaze landing positions $X = (x_1, x_2, ..., x_n)$ measured at time $T = (\mathrm{t}_1, \mathrm{t}_2, ..., \mathrm{t}_n)$ when performing the gaze fixation task. For simplicity, we consider only one dimensional (horizontal one), though the model can be applied to the vertical dimension as well. Given X and assuming normal distribution of landing site position, we can construct the probability density function (PDF) of X as $pdf(X)$.

Let us perform the segmentation of time-series using the sliding window with the length of the local sliding window *w* and the step between adjacent sliding windows *s*. This transformation transforms X into a sequence of vectors $\overline{X} = \left(\{x_1,...,x_W\},\{x_{S+1},...,x_{S+W}\},...,\{x_{kS+1},...,x_{kS+W}\}\right)$, here $k = \lfloor (n-w)/s \rfloor$.

Let M be a time series constructed from the maximal values of $pdf(X)$ meaning the largest probability value of landing: $M = (m_1, m_2, ..., m_k)$, where $M = \max\left(pdf\left(\overline{X}\right)\right)$. This value can be used to characterized the performance of a subject: the higher the value, the more accurate the subject is, the lower the value, the less accurate the subject is.

The probability distribution depends upon many different factors such as the skill of a subject in using the gaze-based interfaces as well as the complexity of the gaze tracking task. Therefore, we must normalize M to remove inter-subject and inter-task variability as follows. Let S be a set of subjects $S = (s_1, s_2, ..., s_P)$. Let M_S be a matrix constructed of M for subjects S. The normalization procedure consists of diving each element of matrix $\overline{M_S}$ by square root of a product of a mean value of each subject and the mean value of all subjects at each time step. The normalized matrix $\overline{M_{S,t}}$ is defined as:

$$\overline{M_{s,t}} = {M_{s,t}} \Big/ \sqrt{E\left(M_{\forall s,t}\right) \cdot E\left(M_{s,\forall t}\right)} \tag{13}$$

here $E(\cdot)$ is the mean (expectation) operator.

Let the grand mean of $\overline{M_{S,t}}$ be $\overline{\overline{M}} = E\left(\overline{M_{S,t}}\right)$. Finally, we perform the fitting of $\overline{\overline{M}}$ to a variant of Banister's model:

$$P(t) = k_1 \cdot e^{-t/\tau_1} - k_2 \cdot e^{-t/\tau_2} \tag{14}$$

here P(t) is performance, k_1 and k_2 are learning and fatigue parameters, t is time, τ_1 and τ_2 are decay parameters for learning and fatigue respectively.

The Banister model assumes that, in response to a training impulse, performance first decreases and then returns to the initial level after a time t_n and then peaks at a higher level after a time t_g [28]. Therefore, we calculate two additional t_n and t_g parameters to evaluate the time needed for subject to rest after the gaze tracking session as follows:

$$t_n = \frac{\tau 1\, \tau 2}{\tau 1 - \tau 2} \ln\left(\frac{k_2}{k_1}\right) \tag{15}$$

$$t_g = \frac{\tau 1\, \tau 2}{\tau 1 - \tau 2} \ln\left(\frac{\tau 1\, k2}{\tau 2\, k2}\right) \tag{16}$$

IV. Gaze Spelling System

A. Architecture

The architecture of the developed prototype gaze speller system is quite simple (see Fig. 1). It consists of the gaze tracking device (Eye Tribe), which is connected to a PC via USB 3.0 connection. On the PC, the core modules are responsible to calibration procedure and gaze feedback. A more detailed description of the architecture and implementation can be found in [29].

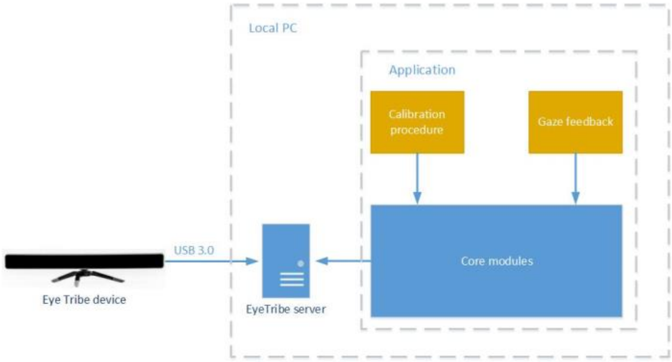

Fig 1.Architecture of the gaze speller prototype system

B. Interface

The primary motive for designing a user interface for a gaze speller is usability as good user experience would also enhance the user acceptance of the system. Our developed interface was inspired by Špakov *et al.* [30] and is based on the concept of “scrollable keyboard” (see Fig. 2).

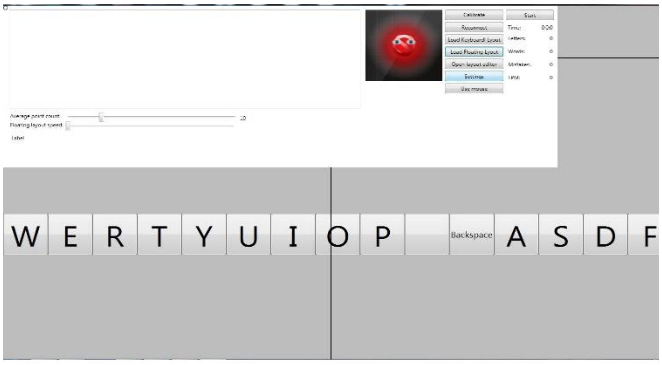

Fig. 2. Interface of the gaze speller

Current implementation uses two scrollable keyboard layouts: 1) a standard QWERTY layout mapped to a single scrollable line of letters, 2) and a letter bigram optimized layout described in subsection IV.C. Feedback is ensured by the black line which always stays on the center of the screen while the one-line keyboard moves underneath it depending on the horizontal position of the gaze. Letter selection for input is provided by eye dwelling. Additional menu buttons are provided for calibration, connection to the gaze tracking device, loading of alternative keyboard layouts, and setting program options. Layout editor has been implemented for designing other keyboard layouts.

C. *Letter bigram optimized layout*

We have developed an alternative layout for the letter scrollbar based on the frequency of bigrams in the language under consideration. First, we computed the number of occurrences of each bigram in text corpora (we used "Alice in Wonderland" from Project Gutenberg) as follows.

Let s_i and s_j be the symbols in alphabet L. Let the frequency of symbol occurrences is $f\left(s_i, s_j\right)$. Next we computer the distance from s_i to s_j, as inverse of frequency:

$$D\left(s_i, s_j\right) = 1/\left(1 + f\left(s_i, s_j\right)\right) \tag{13}$$

To make the matrix D a true distance matrix, the elements at the main diagonal of the matrix are assigned a zero, i.e. $D\left(s_i, s_j\right) = 0$, for all $i = j$.

Next, we consider the problem of optimally placing the letters on the 1D scrolling band as a separate case of the Travelling Salesman Problem (TSP). We can describe the letters as nodes of a graph, and the pairwise distance between nodes is inversely proportional to the frequency of occurrence of bigrams composed from the corresponding letters. A user has to visit all letters only once so that the travelled path is the shortest. The problem is the problem is computationally difficult, and can be solved exactly using brute force search or dynamic programming methods for a small number of nodes (usually <15). As we have 26 letters of English alphabet as well as a number of other symbols (delimiters, numbers, etc.), the brute force search approach is not feasible. We used a simple implementation of Ant Colony Optimization (ACO) to find a near optimal solution to this problem. ACO is a probabilistic technique for solving computational problems which can be reduced to finding good paths through graphs. As the ACO algorithm is probabilistic, each time a different solution is provided. We have repeated the algorithm 100 times and selected the solution with the shortest length of path. Finally, we shifted the solution so that the symbol with the largest frequency is placed in the middle of the scrolling bar. The solution for scrolling bar used in this paper is:

`YWVXQZGKJBURE STHANDICOFMPL`

D. *Adaptive dwell time and word autosuggestion feature*

Using gaze as an input method can be problematic, since the same modality is used for both perception and control. The system needs to be able to distinguish casual viewing of an object from intentional control. Eye movements are also largely unconscious and automatic. Gaze can be easily distracted by movement in the peripheral vision, resulting in unwanted glances away from the object of interest. When necessary, humans can control gaze at will, which makes voluntary eye control possible. For systems based solely on gaze-control, the most common method for preventing erroneous activations is to introduce a brief delay, a so-called ''dwell time'', to differentiate viewing and gaze-control. The duration should match the specific requirements of the task and the user. Expert eye typists require only a very short dwell time duration (e.g. 300 ms) while novices may prefer longer dwell time durations (e.g. 1000 ms) that give them more time to think, react and cancel the action. A long continuous dwelling (fixation) can be uncomfortable and tiring to the eyes. On the other hand, the possibility to adjust dwell time supports efficient learning of the gaze-controlled interface and increases user satisfaction [22].

When using dwell time, the user only initiates the action; the system executes it after a predefined interval. Appropriate feedback plays an essential role in gaze-based interfaces; the user must be given clear indication of the status of the system: if the user is entering text by gaze, he or she cannot see the text appear in the text input field while simultaneously selecting a letter on an on-screen keyboard. Proper feedback can significantly reduce errors and increase interaction speed.

The implemented gaze speller also has the adaptive dwell time and word autosuggestion feature implemented similarly to the ones implemented in EMG speller [31, 32]. The developed speller application is adaptive (input speed can be adapted dynamically in response to the user's state) and intelligent (uses word complete and word frequency features). A common way to implement word prediction is to present a list of predicted words for the user. The words are based on the letters the user has written so far and is sorted based on the frequency of words in the text corpora of the language. The list is dynamically adjusted as more letters are written and the number of possible continuations of the word decreases.

E. *Limitations*

The use of the developed gaze speller for disabled or impaired people may be limited. Some medical conditions cause involuntary head movements or eye tremor, preventing a good calibration or may even restrict eye movements to one direction (vertical) only. Our implementation uses a horizontal scrolling bar layout only. QWERTY and bigram frequency based layout may not be the best choice for people with disabilities who have no previous experience with the QWERTY layout and might thus find another kind of layout (for example, an alphabetically ordered layout) more familiar. The dwell time sets a limit to the maximum typing speed because the user has to wait for the dwell time to elapse before each selection. A long dwell time is good for preventing false selections, but a long fixation on the same target can be tiring to the eyes.

V. EXPERIMENT

A. *Apparatus*

The eyeTribe eye tracker (tracking range 45cm – 75cm, tracking area 40cm x 30cm at 65cm distance) was connected to a HP Ultrabook notebook running Microsoft 8 OS 64-bit with a Intel Core i5-4202Y 1.60 GHz CPU and 4 GB RAM. The application was displayed on a 14" flat LCD display with LED backlight and screen resolution of 1920x1080 pixels. The eyeTribe eye tracker communicates with notebook via USB 3.0 interface.

B. *Subjects*

In the experiment, 8 volunteers (aged 25-32 years, 7 male and 1 female) took part in the test. They were students or staff at the Kaunas University of Technology. All were fluent in English and has no known vision problems. All subjects have provided o written consent prior to the experiment. After experiment was performed, the initial screening of data has shown that the data for one subject was not recorded due to a software glitch. Therefore, only data from 7 subjects were used in further analysis.

C. *Datasets*

For the experiment, an easy-to-memorize phrase was chosen from a set of 500 phrases proposed in [26]. This phrase set is considered the *de facto* standard for text entry evaluations. Punctuation was removed, and the phrases were made case-insensitive.

D. *Procedure*

Prior to collecting data, the experimenter explained the task and demonstrated the software. The experiment was carried out with one disabled person, who could not control his legs and his hand movements are limited. The subject was instructed on the method of text entry, early word selection, error correction, and the audio feedback. He was instructed to enter the given phrases as quickly and accurately as possible and make corrections only if an error is detected in the current or previous word. The subject was allowed to enter a few trial phrases to become familiar with the gaze-controlled selection and correction methods. Then the subjects were instructed to eye type the phrases as rapidly and accurately as possible.

E. *Usage scenario*

Usually gaze-tracking interfaces are designed to imitate operation of a standard pointing device such as a mouse. The gaze tracker either head mounted or attached in front of the user then tracks the user's gaze and transforms it to the screen coordinates. During eye typing, the user first locates the letter on a virtual keyboard by moving his/her gaze to it. The gaze tracking device follows the user's point of gaze while software records and analyses the gaze behavior. For input, the user has to fix his/her gaze at the letter for a pre-defined time interval (dwell time). When the dwell time has passed, the letter is selected by the system and users can move on to gaze to the next letter. Feedback is shown both on focus and on selection.

VI. RESULTS

A. *Data analysis*

The experimental data collected (spatial positions of gaze landing sites at the central letter of the gaze speller interface) are presented in graphically in Fig. 3.

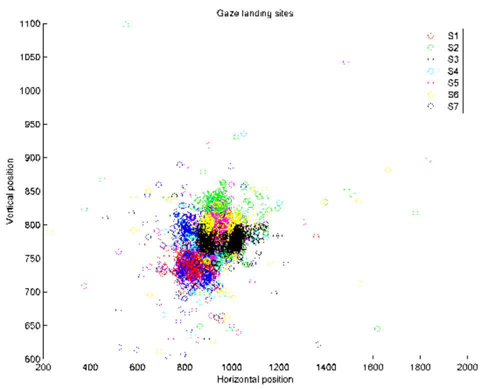

Fig. 3. Gaze landing site positions (all subjects)

To illustrate the effect of fatigue, we have constructed PDFs of first 100 gaze landing sites (horizontal position only) and PDFs of last 100 gaze landing sites. Only horizontal position was considered since the scrollable keyboard of the gaze speller we have used in our experiments requires that subjects use horizontal movements of gaze. The results are presented in Fig. 4 and Fig. 5, respectively.

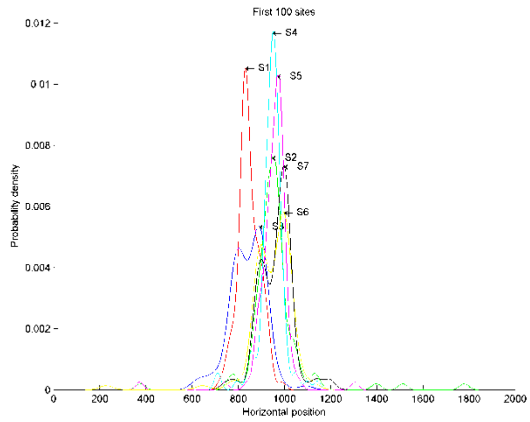

Fig. 4. Probability density functions of all subjects for horizontal gaze landing sites (first 100 sites)

Note that the shape of PDFs has changed: for most subjects the PDF has flattened, i.e., its maximum value has decreased and the spread of values has increased as is visible from the 'thicker' tails of the distributions. From the shape of PDFs we also can see the bimodality of distribution for most subjects indicating that two factors may be in effect.

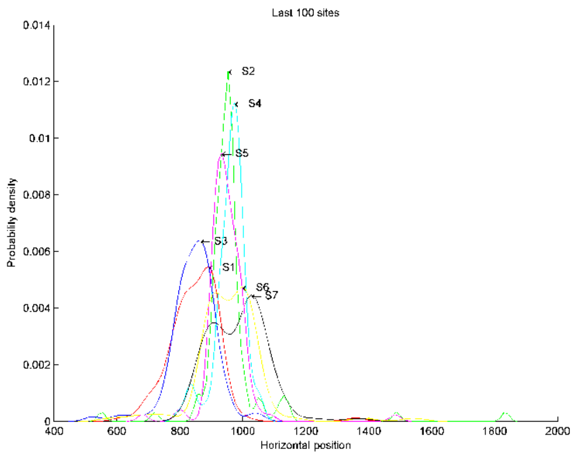

Fig. 5. Probability density functions of all subjects for horizontal gaze landing sites (last 100 sites)

To model user fatigue during the experiment, we assume that subject accuracy to land their gaze follows the classical exponential decay model with two components: the positive one is corresponding to learning (training), and the negative one is corresponding to user fatigue (as defined in the Banister *et al.* [16] model).

B. Model fitting

The model proposed in Section IV of this paper was used. The model parameters were estimated for each subject using the non-linear least squares iterative method, by minimizing the residual sum of quadratic differences between the real and the modelled performances (RSS) with a Trust-Region-Reflective Least Squares Algorithm. Computations were performed using Matlab 2013a (version 8.1, Mathworks). The 95% confidence bounds were calculated for the estimated parameters. The model fitting results are presented in Fig. 6.

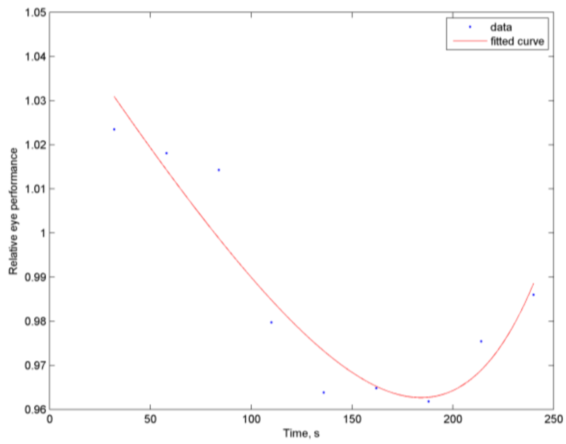

Fig. 6. Model fitting

The parameters of best fitted model (w = 62, s = 26) are presented in Table I.

MODEL PARAMETERS (MEAN AND CONFIDENCE BOUNDS)

Parameter	Mean	95% confidence bounds	
k1	1.051	1.021	1.081
t1	1471	-265	2677
k2	-0.0013	-0.012	0.015
t2	55.8	-15.8	127

To evaluate fitness of the model, the determination coefficient was calculated as: $R^2 = 1 - (RSS / TSS)$, where TSS is the total sum of squares, as well as the degree-of-freedom adjusted coefficient of determination. The sums of squares due to error (SSE) and root mean square error (RMSE) values were computed to evaluate difference between modelled and real data values (see Table II).

MODEL FITNESS CHARACTERISTICS

Characteristic	Value
SSE	0.0005
Degrees of freedom	5
R-square	0.9027
Adjusted R-square	0.8442
RMSE	0.0098

C. PCA analysis

For data analysis, Principal Component Analysis (PCA) was performed on the covariation matrix of the horizontal gaze landing position matrix. Factor 1 accounted for 73% of the total variance. This factor was highly negatively related to the performance and can be attributed to fatigue. Factor 2 accounted for 17% of the total variance. This factor was positively related to performance and could be attributed to learning (training).

D. Statistical validation

An important question is how many data points are needed per parameter to enable statistical analysis. For multiple linear regression, 15 observations per parameter is recommended. Since the Banister model is a non-linear one, more data points per parameter may be required. In our experiment we used 338 observations per 4 parameters, which should be enough for a non-linear model.

To analyse stability of this model, iterative computation was performed with the same data but minus one subject, chosen randomly following the methodology described by Hellard *et al.* [33]. A method is considered as unstable if small perturbations in the data can cause significant changes in the estimations. One hold-out 7-fold cross validation was done to evaluate the stability of results. In each fold, the data of one subject was removed, and the computation of the model repeated. Different models obtained during the cross-validation procedure are presented in Fig. 7.

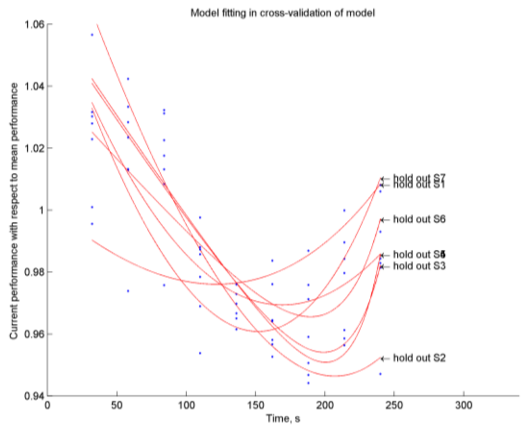

Fig 7. Models obtained using one hold out cross-validation

The model parameter results are summarized in Table III.

RESULTS OF CROSS-VALIDATION USING ONE HOLD-OUT

Model parameter	Mean value	SD of value
k1	0.974	0.136
k2	-0.085	0.135
t1	1131	654
t2	-149	147
tn	308	244
tg	177	31

Finally, the PDFs of the decay parameters of fatigue and learning obtained during the cross-validation procedure are shown in Fig. 8. We can see that the effect of fatigue is developed earlier and is stronger than the effect of learning.

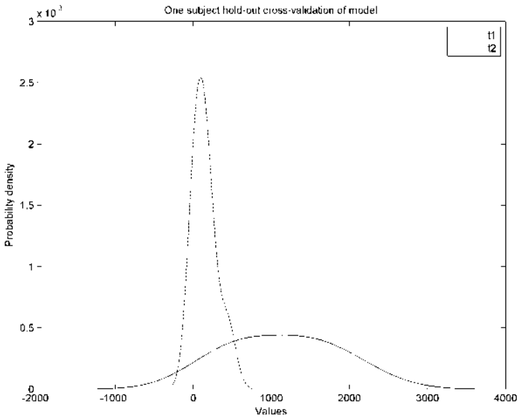

Fig. 8. Probability density of model's decay parameter values

VII. CONCLUSION

Our initial experiments with evaluating user fatigue when a subject is working with the gaze spelling system suggest that the classical model of Banister *et al.* proposed to evaluate performance of sports' athletes, is applicable to the domain of physiological computing as well. We have analysed the accuracy of gaze landing in performing text entry task. Experimental data was fitted to the proposed fatigue model adapted from Banister *et al.* The proposed model has been validated using the one hold out cross-validation procedure.

The variability between user performance is significant and is larger than intra-user variability due to learning and fatigue effects. PCA analysis shows that intra-user variability can be explained by two factors: fatigue (73% of variance) and learning (17% of variance). Both these factors explain 90% of intra-user variance. Since the learning acts slower than fatigue and has less effect on the results, time-to-peak value is smaller than time-to-initial performance (which is contrary to sports athletes performance). Therefore, it is advised to use the time-to-initial performance as an estimate of rest time. The model allows to calculate the time needed to rest after each session. The mean time to rest calculated from all user data is 308+-244 s (5+-4 min). The large SD of the total estimate is due to large variability of user performance. In fact, the analysis of data shows that the length of the test session was too short for two users, which does not allow for the fatigue effects to show up. The rest time can be calculated for each user individually. The prevalence of fatigue effect over learning effect means that the usability of the gaze tracking based interface is low due to negative reinforcement. Until the fatigue issue could be solved, the users are not likely to be using this kind of interfaces.

The results of the study presented in this paper confirm that the stimulus induced fatigue on users' eyes among the tasks conducted. The main findings of this study are that: 1) learning and fatigue effects are present in the gaze tracking data; 2) learning is slower process than fatigue; 3) the parameters of learning and fatigue can be evaluated using Banister model; 4) time required for eye rest break can be evaluated.

Further research is needed in analysing long-term effects of eye fatigue, which were not covered by this experiment. A larger (in terms of a number of subjects) and longer study is needed to validate the values of the model parameters.

REFERENCES

[1] A. Schroeder, and M. Wirsing, "Developing Physiological Computing Systems: Challenges and Solutions", Software Engineering 2012, pp. 21-36, 2012.

[2] L. J. Hettinger, P. Branco, L. M. Encarnaco, and P. Bonato, "Neuroadaptive technologies: Applying neuroergonomics to the design of advanced interfaces", Theoretical Issues in Ergonomic Science, vol. 4 (1–2), pp. 220–237, 2003.

[3] N. B. Serbedzija, and S. H. Fairclough, "Biocybernetic loop: From awareness to evolution", IEEE Congress on Evolutionary Computation, IEEE, pp. 2063-2069, 2009.

[4] J. Allanson, and S. H. Fairclough, "A research agenda for physiological computing", Interacting with Computers, 16(5), pp. 857-878, 2004.

[5] H. Huang, Y. Chen, W. Xu, H. Sundaram, L. Olson, T. Ingalls, T. Rikakis, and J. He, "Novel design of interactive multimodal biofeedback system for neurorehabilitation", Conf. Proc. IEEE Eng. Med. Biol. Soc.,1:4925-8, 2006.

[6] L. E. Nacke, M. Kalyn, C. Lough, and R. L. Mandryk, "Biofeedback game design: using direct and indirect physiological control to enhance game interaction", Proc. of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11), ACM, pp. 103-112, 2011.

[7] A. Câmara, "Natural User Interfaces," INTERACT 2011:1, 2011.

[8] S. Lanthier, E.F. Risko, D. Smilek, and A. Kingstone, "Measuring the separate effects of practice and fatigue on eye movements during visual search," CogSci 2013.

[9] J.A. Hawley, and T. Reilly, "Fatigue revisited", Journal of sports sciences, 15 (3), pp. 245–6, 1997.

[10] C. Blehm, S. Vishnu, A. Khattak, S. Mitra, and R.W. Yee, "Computer vision syndrome: a review," Survey of ophthalmology 50,3 (2005),253-262.

[11] S. Marcora, "Mental fatigue impairs physical performance in humans", Journal of Applied Physiology, 106 (3), pp. 857–864, 2009.

[12] M. K. Scheffers, D. G. Humphrey, R. R. Stanny, A. F. Kramer, and M. G. Coles, "Error-related processing during a period of extended wakefulness", Psychophysiology, vol. 36, pp. 149–57, 1999.

[13] E. Abdulin, and O. Komogortsev, "User Eye Fatigue Detection via Eye Movement Behavior," Proc. of the 33rd Annual ACM Conference on Human Factors in Computing Systems, 1265-1270, 2015.

[14] R. A. McKinley, L. K. McIntire, R. Schmidt, D. W. Repperger, and J. A. Caldwell, "Evaluation of Eye Metrics as a Detector of Fatigue," Human Factors (HF) 53(4):403-414, 2011.

[15] T. Taha, and S. G. Thomas, "Systems modelling of the relationship between training and performance", Sports Med., 33(14), 1061-73, 2003.

[16] E. W. Banister, T. W. Calvert, M. V. Savage, and A. Bach, "A system model of training for athletic performance", Australian Journal of Sports Medicine, 7, pp. 170-176, 1975.

[17] T. W. Calvert, E. W. Banister, and M. V. Savage, et al., "A systems model of the effects of training on physical performance", IEEE Trans Syst Man Cybern, 6 (2), pp. 94-102, 1976.

[18] R. H. Morton, J. R. Fitz-Clarke, and E. W. Banister, "Modeling human performance in running", J Appl Physiol, 69 (3), pp. 1171-7, 1990.

[19] T. Busso, H. Benoit, R. Bonnefoy, L. Feasson, and J. R. Lacour, "Effects of training frequency on the dynamics of performance response to a single training bout", Journal of Applied Physiology, 92, 572-580, 2002.

[20] G. W. McConkie, P. W. Kerr, M. D. Reddix, and D. Zola, "Eye movement control during reading: I. The location of initial eye fixations on words," Vision Research, 28, 1107–1118, 1988.

[21] R.J. van Beers, "The sources of variability in saccadic eye movements," J. Neurosci. 27, 8757–8770, 2007.

[22] P Majaranta, U.K. Ahola, and O. Špakov, "Fast gaze typing with an adjustable dwell time," Proc. of the 27th int. conference on human factors in computing systems, CHI 2009, pp. 357–360, 2009.

[23] D. J. Lohr, E. Abdulin, and O. V. Komogortsev, "Detecting the onset of eye fatigue in a live framework," Proc. of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications (ETRA '16), pp. 315-316, 2016.

[24] O. V. Komogortsev, D. V. Gobert, S. Jayarathna, D. H. Koh, and S. Gowda, "Standardization of Automated Analyses of Oculomotor

Fixation and Saccadic Behaviors," IEEE Transactions on Biomedical Engineering 57, 11, pp. 2635-2645, 2010.

[25] A.S. Arif, and W. Stuerzlinger, "Analysis of Text Entry Performance Metrics," Proc. IEEE TIC-STH 2009, 2009, pp. 100-105.

[26] I. S. MacKenzie, and R. W. Soukoreff, "Phrase sets for evaluating text entry techniques," Extended Abstracts on Human Factors in Computing Systems (CHI'03), pp. 754-755, 2003.

[27] K. Itoh, H. Aoki, and J. P. Hansen, "A comparative usability study of two Japanese gaze typing systems," Proc. of ETRA 2006, pp. 59-66, 2006.

[28] R Fitz-Clarke, R.H. Morton, and E.W. Banister. "Optimizing athletic performance by influence curves," Journal of Applied Physiology. 1991;71:1151–1158.

[29] M. Vasiljevas, J. Šalkevičius, T. Gedminas, and R. Damaševičius, "A prototype gaze-controlled speller for text entry," Proc. of the int. Symp. for young scientists in technology, engineering and mathematics (System 2015). CEUR Workshop Proc. vol. 1543, 2015, p. 79-83.

[30] O. Špakov, and P. Majaranta, "Scrollable Keyboards for Casual Eye Typing," PsychNology Journal 7(2), Special issue on "Gaze Control for Work and Play", pp. 159-173, 2009.

[31] M. Vasiljevas, R. Turčinas, and R. Damaševičius, "EMG speller with adaptive stimulus rate and dictionary support," Proc. of the 2014 federated conference on computer science and information systems, 2014, pp. 227-234.

[32] R. Damaševičius, M. Vasiljevas, and T. Šumskas,"Development of a concept-based EMG-based speller," DYNA. 2015, 82(193), pp. 170-179.

[33] P. Hellard, M. Avalos, L. Lacoste, F. Barale, J. C. Chatard, and G. P. Millet, "Assessing the limitations of the Banister model in monitoring training", J Sports Sci., 24(5), pp. 509-20, May 2006.