# Pupillography as indicator of programmers' mental effort and cognitive overload


Ricardo Couceiro
*CISUC, University of Coimbra*
Coimbra, Portugal
rcouceir@dei.uc.pt

Gonçalo Duarte
*CISUC, University of Coimbra*
Coimbra, Portugal
duarte.1995@live.com.pt

João Durães
*CISUC, Polytechnic Institute of Coimbra*
Coimbra, Portugal
jduraes@isec.pt

João Castelhano
*ICNAS, University of Coimbra*
Coimbra, Portugal
joaocastelhano@uc.pt

Catarina Duarte
*ICNAS, University of Coimbra*
Coimbra, Portugal
catarinaduarte86@gmail.com

Cesar Teixeira
*CISUC, University of Coimbra*
Coimbra, Portugal
cteixei@dei.uc.pt

Miguel Castelo Branco
*ICNAS/CIBIT, University of Coimbra*
Coimbra, Portugal
mcbranco@fmed.uc.pt

Paulo de Carvalho
*CISUC, University of Coimbra*
Coimbra, Portugal
carvalho@dei.uc.pt

Henrique Madeira
*CISUC, University of Coimbra*
Coimbra, Portugal
henrique@dei.uc.pt



***Abstract*—Our research explores a recent paradigm called Biofeedback Augmented Software Engineering (BASE) that introduces a strong new element in the software development process: the programmers' biofeedback. In this Practical Experience Report we present the results of an experiment to evaluate the possibility of using pupillography to gather biofeedback from the programmers. The idea is to use pupillography to get meta information about the programmers' cognitive and emotional states (stress, attention, mental effort level, cognitive overload,...) during code development to identify conditions that may precipitate programmers making bugs or bugs escaping human attention, and tag the corresponding code locations in the software under development to provide online warnings to the programmer or identify code snippets that will need more intensive testing. The experiments evaluate the use of pupillography as cognitive load predictor, compare the results with the mental effort perceived by programmers using NASA-TLX, and discuss different possibilities for the use of pupillography as biofeedback sensor in real software development scenarios.**



***Keywords— software faults, pupillography, programmers' biofeedback, mental effort, cognitive overload, human error.***


## I. INTRODUCTION

Software development is (still) a human intensive task. Although there are examples of automatic generation of executable code from high-level specifications (however, faults may lie in the specification as well), most of the software produced today results from an intensive human made process. And humans may fail while doing abstract and complex tasks such as requirement elicitation, functional specification, software architecture design, and in particular code development and testing. Decades of advances in software engineering methods have mitigated the problem of residual software faults (bugs), but the reality is that even when software is developed using highly demanding and mature software development processes, the deployed code still has a quite high density of bugs, ranging from 2 to 5 bugs per KLoC [1, 2], while the "industry average is about 15 to 50 errors per 1000 lines of delivered code" [3]. Knowing the current trend for the overinflated size of software (e.g., the first version of Linux had 17 KLoCs while the current version has more the 20 millions of LoC, a modern high end car has 100 millions of LoC, etc), we may say that bugs are there to stay, if not claiming that things are getting worse.

Field studies characterizing in detail real bugs found in deployed software are quite rare in the literature. But the few available studies show an impressive similarity of bug types classification (according to the Orthogonal Defect Classification), even when software products use different software development methodologies, different programming languages and resulted from different technical cultures [4,5]. This suggests that software developers tend to err in similar ways and originate a limited set of bug types. This observation concurs with recent cognitive taxonomy on human error causes for software defects [6, 7], which show that a relatively small number of cognitive human error modes can be traced as the primary cause of software defects [6].

If bugs have a causal link with a limited number of error-prone contexts in software development, as suggested by previous studies [4, 5, 6, 7], it should be possible to detect such error prone scenarios and use such information to improve software quality. This is precisely the idea of the Biofeedback Augmented Software Engineering (BASE) approach, which introduces a strong new element in the software development process: the programmers' biofeedback captured by wearable and non-intrusive sensors that monitor Autonomic Nervous System (ANS) physiologic manifestations. The goal is to detect signals that can be linked to error prone scenarios related to programmers' cognitive and emotional states (stress, attention, mental effort level, cognitive overload,...), as established by cognitive human error modes [6].

This paper presents experimental results evaluating the possibility of using pupillography (i.e., the rapid changes of pupil size) as an indicator of programmers' mental effort and cognitive overload, which is related to some prevalent cognitive human error modes. The big advantage of pupillography is that

it is a non-intrusive method that is fully compatible with traditional software development environments. Thus, it can be quickly adopted as one of the sensors for concrete Biofeedback Augmented Software Engineering implementations. The experimental results show a clear mapping between the mental effort measured using pupillography and both the software complexity metrics of the different code excerpts used in the experiments and the subjective mental effort perceived by the programmers using NASA-TLX (Task Load Index)[1].

The next section presents the background of our research and related work, Section III describes the elements of the experiment and protocol, Section IV presents the methodology used in the data analysis, Section V presents and discusses the results, Section VI discusses future work and utilization perspectives and Section VII concludes the paper.

## II. Background and Related Work

The emerging research area of software fault defense based on cognitive human error models and mechanisms has gained ground in recent years. It is rooted in human error theories and models [8] and generally adapts human error models and taxonomies to software development process [6, 7], with the goal of defining defensive strategies or to improve specific steps such as requirements elicitation [9].

A different research path that has also emerged in recent years is the study of the human brain mechanisms behind software code comprehension using neuroscience approaches, particularly using heavy equipment such as functional magnetic resonance imaging (fMRI), near field infrared spectroscopy (fNIRS), and electroencephalography (EEG). In [10], working memory, attention, and language processing have been identified as brain regions involved in code comprehension and in the identification of syntax errors. Another study [11] compared code and natural-language text comprehension to identify the brain mechanisms involved in each activity and [12] studied the mental execution of source code by programmers.

Software bugs have been studied for the very first time in a recent neuroscience study [13] that reported a causal connectivity brain pattern associated to the "eureka" moment of bug intuition. This study also identified the distinct role for the insula in software bug monitoring and detection, as the insula activity levels were critically related to the quality of bug detection, showing that the activity in this salience network region evoked by bug suspicion was predictive of bug detection accuracy. This was the first time a brain signal could be related with software skills on bug detection [13].

Obviously, brain signals predicting accurate bug detection, such as identified in [13], were obtained using fMRI scanners and cannot be used in practical software development environments. But these findings show a possible pathway for future establishment of reliable connections between software errors at the fundamental brain level and physiological responses driven by the autonomic nervous system (ANS) that can be monitored by low intrusive sensors compatible with programmers' code development activities.

In fact, there are many commercial wearable devices that can monitor ANS driven response such as heart rate variability (HRV), breathing rhythm and electrodermal activity (EDA), eye tracking with pupillography, and even wearable versions of electroencephalography (EEG). However, physiological sensors are very sensitive to many other causes of physical body response, totally unrelated to the software development activities, and should be used with care as a single source of programmers' biofeedback, at least while possible connections with deep brain mechanisms such as the ones reported in [13] are not available to be used as filter to remove ANS response not related to human errors

A very recent study has shown that HRV can be used as indicator of programmers' mental effort and cognitive load [14]. HRV was also proposed to predict code quality and help guiding the testing effort [15]. Although HRV can be captured by low intrusive sensors, pupillography seems to be even better as it does not require any physical contact with the programmer or any change in the programming environment. This is exactly the target of our study, to evaluate whether pupillography can be used as indicator of programmers' mental effort and cognitive load.

The pupil is the opening in the center of the iris, a pigmented structure that contains two antagonistic muscle groups: the sphincter and the dilator muscles [16]. Receiving input from both parasympathetic and sympathetic components of the ANS, these two muscles group have an extensive control over the pupil size (diameter), reflecting a dynamic equilibrium of the opposing sympathetic and parasympathetic activations of the ANS. The parasympathetic excitation (and/or sympathetic inhibition) results in the dilation of the pupil, while sympathetic excitation (and/or parasympathetic inhibition) has the opposite effect, i.e. the constriction of the pupil [17].

The study of the relationship between the pupil activity, attentional effort and cognitive processing started almost 6 decades ago with the works from Hess and Polt [18] and Kahneman and Beatty [19]. More recently, several studies have showed an association between the increase of the pupil size and the increase of mental activity [17, 21] to various sources of psychological stress (e.g. [20]).

The analysis of the spectral content of the pupil diameter (PD) signal and its association with the increase of cognitive demanding tasks also drew researchers attention since Lüdtke and its colleagues [22] quantified, in the frequency domain, pupillary fatigue waves (below 0.8 Hz) neglecting fast pupillary changes (> 1.5625 Hz), to measure sleepiness. Since then, spectral analysis of PD data is commonly applied to the analysis of pupillary diameter fluctuations. More particularly, Nakayama and Shimizu [23] found an increase of the spectral density of PD signals within frequency bands of 0.1–0.5 Hz and 1.6–3.5 Hz as a function of cognitive task difficulty, during calculation tasks.

Other interesting research focused on the ratio between the low (LF) and high frequency (HF) bands of the PD signal spectra. Murata and Iwase [24] associated increase of LF/HF ratio with mental workload (in mental arithmetic and Sternberg short-term memory tasks). Here, the LF band was defined from 0.05 and 0.15 Hz, while the HF band between 0.3 and 0.5 Hz.

[1] NASA-TLX site: https://humansystems.arc.nasa.gov/groups/TLX/

Peysakhovich et al. [25] showed evidence that the LF/HF ratio (LF - 0–1.6 Hz – and HF - 1.6–4 Hz) of the PD power spectral densities are sensitive to the cognitive load but not to luminance changes.

## III. Experiment design and protocol

The goal of our experiments is to prove that the distinct levels of mental effort experienced in code comprehension (e.g., during software inspections) can be captured by manifestations of the autonomic nervous system (ANS) activity. In order to assess sympathetic and parasympathetic activity imbalances of the ANS induced by mental effort during reading and understanding of code of distinct complexity we have implemented an observational study using different technologies of low intrusive/wearable sensors that could potentially be applied in producing contexts of software code inspections. The set of sensors selected were the ECG, EDA, and eye tracking with pupillography. ECG and EDA signals were collected using BiosignalsPlux from Plux and the pupil diameter (among other variables) was collected using an SMI eye tracker. These were captured using a common time base in order to allow cross analysis as well as to explore complementary manifestations of the ANS. In this paper we will focus only on the analysis of the variability of the pupil diameter.

The observational study included 30 experienced programmers (male: 24, female: 6, age: 24.4 ± 6.18 yrs) in Java programming language. The volunteers were selected using an interview-based screening process, which enabled us to categorize each participant as Intermediate (12 participants), Advanced (14 participants) or Expert (4 participants) in Java programming. Participants have been informed about the purpose of the experiment. In particular, they have been told that they are not under evaluation in any way, to reduce the effect of being "watched". In any case, the way pupilography features are extracted is essentially differential (i.e. in comparison with baseline activities such as reading a text or doing nothing during 30 seconds), so the effect of being "watched" is not significant.

The data collection campaign received ethical clearance by the University's Ethical Commission, in accordance with the Declaration of Helsinki, and all participants provided written informed consent. An open data philosophy is followed in this project. Hence, the anonymized data from this research will be available upon request to the authors.

The designed protocol involved code inspection of 3 small Java programs, herein identified by C1, C2 and C3, with different complexities. Special care was taken during the program development phase in order to maintain consistency in the programming style and to avoid added difficulty, such as math or algorithm, not directly attributable to the code complexity. In order to achieve these goals, in C1 a simple program was implemented that counts the number of values existing in a given array that fall within a given interval using a straightforward loop. Program C2 translates the multiplication of two numbers using the basic algorithm for arithmetic multiplication. First the program converts the string input parameters into byte arrays. Next, a straightforward multiplication is implemented where every digit from one number is multiplied by every digit from the other number, from right to left. Finally, for program C3 a search problem was implemented where the largest occurrence of an integer cubic array inside a larger cubic array is searched. This problem exhibits many nested loops and, therefore, has a high cyclomatic complexity. Table I summarizes the complexity metrics of the three programs. It should be mentioned that in C1 and C3 the algorithm is coded in one function, whereas C2 is spread across two functions. In spite of C2 and C3 having a similar number of code lines, C2 might be easier to read and understand than C3.

TABLE I. Programs C1, C2 and C3 used in the experiments

| Prog. | Lines of code | Nested Block Depth | No. params. | Cyclomatic complexity |
|---|---|---|---|---|
| C1 | 13 | 2 | 3 | 3 |
| C2 | 42 (12+30) | 3 | 3 | 4 |
| C3 | 49 | 5 | 4 | 15 |

Throughout the experiment similar condition were applied in all data collection sessions using a controlled environment without distractions, noise or the presence of people unrelated to the experiments. The main steps of the protocol, performed on the screen of a laptop, are:

1. A baseline was captured by exhibiting an empty grey screen with a black cross in its center for 30 seconds.
2. A simple reference activity for the purpose of data analysis was captured using a text in natural language to be read by the participant (60 seconds max.).
3. Empty grey screen with a black cross for 30 seconds.
4. Screen displays the code in Java of the program to be analyzed for code comprehension (C1, C2, C3). This step last 10 minutes maximum for each program.
5. Empty grey screen with a black cross for 30 seconds.
6. Survey 1: NASA-TLX to assess the subjective mental effort perceived by each participant in the code comprehension.
7. Survey 2: understanding of the program.

This protocol is repeated 3 times for each participant. In each iteration, a different program (C1, C2, C3) is used in step 4.

## IV. Methods and analysis procedure

In the present study we aim to evaluate the changes of the pupil diameter (PD) during the analysis of code using different frequency domains. It has been reported that the power spectrum density of pupil signals increases within certain band intervals as a function of cognitive task difficulty [23]. Therefore, since these frequency bands are not well established in the literature, our study also investigates how the spectral components change when performing tasks (i.e., analyzing code snippets) with different complexities.

To perform this analysis, the PD signal from the left eye was first pre-processed (removal of blink related artifacts and outliers, resampling and reconstructing intervals of missing data) and estimation of the spectral content of the PD in a time-variant fashion.

### A. Pre-processing

First, all intervals labeled by the eye tracker as having invalid pupil diameter (PD) values ($\Delta_{inv}$), due to blink events or other external factors were considered as inaccurate. Additionally, it was observed that in the vicinity of these intervals $\Delta_{inv}$ the PD values were questionably larger. In order to exclude these artifacts, the intervals adjacent to $\Delta_{inv}$, 100 ms before $\Delta_{inv}$ onset and 100 ms after $\Delta_{inv}$ offset were also considered as inaccurate (i.e.$PD\big([\Delta_{inv} - 0.1, \Delta_{inv} + 0.1]\big) \in \mathcal{G}_{OUT}$, being $\mathcal{G}_{OUT}$ the set of excluded PD readings.

Second, spurious values in the pupil diameter signal that do not reflect the underlying physiological process (e.g. disproportionately large dilation speeds and abnormally large deviations from the trend line), were also considered as inaccurate values. To detect these, an outlier detection algorithm based on the boxplot analysis [26] was adopted. The pupil diameter signal was differentiated ($PD'$) and the lower quartile (Q1: 25$^{th}$ percentile), the upper quartile (Q3: 75$^{th}$ percentile) and the interquartile range ($IQR = Q3 - Q1$) were identified. The PD value at the instant $t$ is considered as outliers if:

$$PD'(t) < Q1 - 1.5\,IQR \vee PD'(t) > Q3 + 1.5\,IQR \quad (1)$$

All the identified outliers are considered as abnormal values and therefore belonging to $\mathcal{G}_{OUT}$.

Finally, the identified inaccurate values were interpolated using a shape-preserving piecewise cubic interpolation and the resulting PD signal was down sampled to 20 Hz, reducing the data size considerably while preserving the frequency contents to be studied, from 0 to 10 Hz.

#### 1) Reconstruction of the PD signal

Artifacts related to eye blinks and other external factors are well known to have a great impact in the analysis of the PD signals, both in the time and frequency domains [23]. In order to reduce the influence of these factors in the current analysis, we used an algorithm for filling in missing data based on Singular Spectrum Analysis (iterative SSA) [27, 28, 29]. The basis of this algorithm is to decompose the original time series into a sum of components with meaningful interpretation (trends, oscillatory modes or noise) and reconstruct the intervals with missing data using an arbitrary number of components in an iterative fashion. Using only temporal correlations present in the time series, the reconstructed time series in the intervals corresponding to missing data present produce better estimates of the missing data when compared to an average or an interpolation, an essential element when considering analysis in the frequency domain.

#### 2) High pass filtering

Prior to the analysis in the frequency domain, the PD time series was high-pass filtered with a very low cutoff frequency ($4 \times 10^{-4}$ Hz) in order to minimize the effects of medium-term nonstationary within the time interval under analysis [30].

### B. Time-variant frequency analysis

To investigate the influence of the sympathetic and parasympathetic activity in the pupil diameter we used an approach similar to the classic heart rate variability (HRV) analysis [31], as we are interested in the variability of the pupil.

A sliding window of 30 sec shifted by increments of 1 sec was used to decompose in consecutive subsets of the time series. For each subset, the estimation of the spectral contents was performed using a parametric spectral analysis via the estimation of the autoregressive power spectral density using Burg's method [32]. For each estimated spectrum 6 frequency bands were analyzed. Firstly, the classic HRV frequency bands were analyzed: 1) aVLF - area of the spectra in the very low frequency (VLF) band (0-0.04 Hz); 2) aLF - area of the spectra in the low frequency (LF) band (0.04-0.15 Hz) and; 3) aHF – area of the spectra in the high frequency (HF) band (0.15-0.4 Hz). Secondly, aiming to investigate the influence in higher frequency bands, the area of the spectra was also accessed in: 1) 0.4-1.6 Hz frequency band ($aVHF_{[0.4\text{-}1.6]}$); 2) 1.6-5 Hz frequency band ($aVHF_{[1.6\text{-}5]}$) and; 5-10 Hz frequency band ($aVHF_{[5\text{-}10]}$). While, in the classic HRV analysis the HF component is commonly accepted as a marker of parasympathetic activity, the LF component is considered as a primary indicator of sympathetic modulation [31], in the analysis of PD data the choice of the frequency bands and their physiological interpretation are still vague**.**

#### 1) Data normalization

Due to the wide inter-subject changes in the frequency contributions observed in previous studies [33], the PD signal was normalized in each trial according to the median (or mean) of the respective index in the resting phase (step 2 of the protocol)). The choice of the statistical operator during this phase was performed according to the result of the one-sample Kolmogorov-Smirnov test.

TABLE II. DATA LABELS, DESCRIPTION AND STEPS IN PROTOCOL

| Data Label | Description | Step in protocol |
|---|---|---|
| REST | Reading of text in natural language before CODE 1, CODE 2 and CODE 3 | 2 |
| CODE | Comprehension of code – CODE 1, CODE 2 and CODE 3 | 4 |
| REST1 | Reading of text in natural language before CODE 1 | 2 |
| CODE1 | Comprehension of code – CODE 1 | 4 |
| REST2 | Reading of text in natural language before CODE 2 | 2 |
| CODE2 | Comprehension of code – CODE 2 | 4 |
| REST3 | Reading of text in natural language before CODE 3 | 2 |
| CODE3 | Comprehension of code – CODE 3 | 4 |

## V. RESULTS AND DISCUSSION

The spectral components of the signals of 30 volunteers were extracted and analyzed according to the methodology proposed in steps 2 and 4 of the previously defined protocol, leading to the definition of 8 major groups of data shown in Table II.

In order to visualize the distribution of data values within each defined group, to compare it with the adjacent groups and with the remaining analyzed frequency bands and improve the interpretation of the extracted data, an exploratory data analysis was conducted resorting to box plot analysis. Using this technique it is possible to access the amount of dispersion, and its asymmetry as well as the outliers.

Figure 1 presents the boxplots for the analyzed frequency bands in each of the predefined data groups representing the global view (in a range of 0 to 25 n.u.), while Figure 2 shows the boxplots in a closer view in the range [0 to 2.75] n.u.

In Figure 2 there is a clear distinction between REST group and CODE groups, having the CODE groups median values of the normalized PSD area below 0.923, within the frequencies from 0 to 0.4 Hz. From the analysis of the REST group in each of the frequency bands, it is possible to observe that the largest variance is presented in the VLF band (0 to 0.04 Hz). It is also possible to observe that the variance decreases from VLF band ($IQR \approx 1 \text{n.u.}$) to $VHF_{[5\text{-}10]}$ ($IQR \approx 0.25$ n.u.) and a similar trend is also observed for the outliers and whiskers.

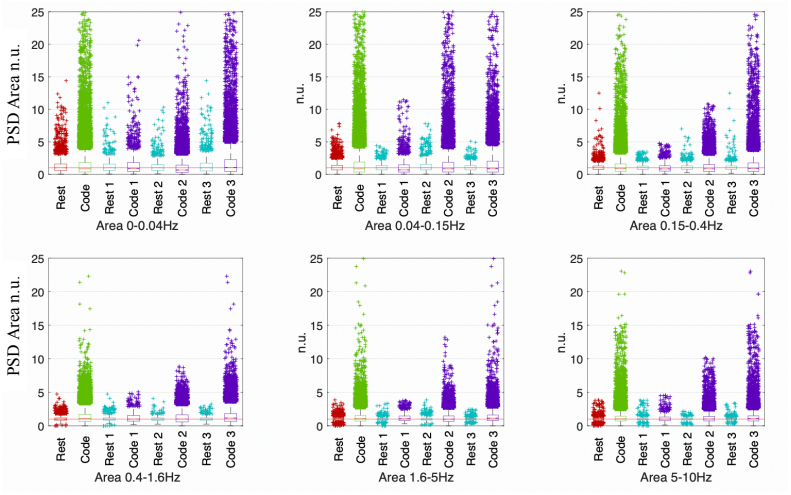


Figure 1 - Boxplot analysis of the frequency bands from 0 to 10 Hz

The analysis of the CODE group shows a similar dispersion in the VLF and LF bands, while the median value remains below the baseline from VLF to HF bands. From $VHF_{[0.4\text{-}1.6]}$ to $VHF_{[5\text{-}10]}$ we can see the median value remain almost unchanged above the baseline, followed by a decrease in the dispersion of the values. The outliers and whiskers follow a similar trend, decreasing from LF to $VHF_{[5\text{-}10]}$.

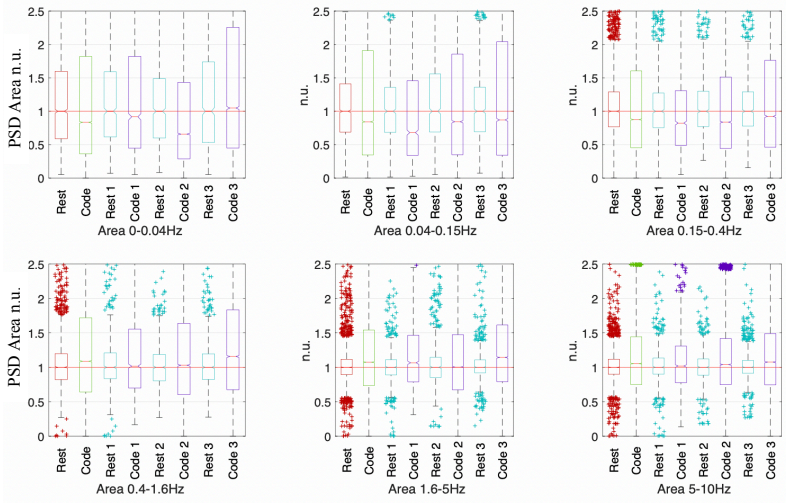


Figure 2 – Closer view of the boxplot analysis in in the range [0 to 2.75] n.u.

From the analysis of the symmetry of the distributions corresponding to the REST and CODE data groups, it is possible to observe that while the values in the REST group remain symmetric around the median value, in the CODE group there is a positive skewness, presenting a tail on the right side of the distributions, denotating the presence of more values largely above the median.

This separation is not so clear in the frequencies above 0.4 Hz, although it is more evident the increase of the median values of the normalized PSD area with the increase of the code complexity, especially in the 5-10 Hz frequency range

It is also possible to observe a larger dispersion of the values in CODE groups (represented by larger IQRs and the number of outliers above the baseline) when compared to the REST groups (small IQRs and low number of outliers). This observations becomes even more evident from HF to $VHF_{[1.6\text{-}5]}$ where there is an increase in the gap between these groups considering these characteristics, which suggests a higher (although sporadic) activation of the parasympathetic pathways during the analysis of the code snippets. This is a clear indication of the mental load of participants in the different steps, showing a clear difference between CODE and REST and among the different codes.

Comparing the evolution of the distributions within the CODE groups in Figure 2, it is possible to perceive an increase in the median values of the LF to $VHF_{[5\text{-}10]}$, from CODE 1 to CODE 3. This trend is more noticeable in the frequency band $VHF_{[0.4\text{-}1.6]}$. While in the HF and HF bands the median values remain below the baseline, in the $VHF_{[0.4\text{-}1.6]}$, $VHF_{[1.6\text{-}5]}$ and $VHF_{[5\text{-}10]}$ the median values exceed the baseline. It is also visible that the outliers concentration and range increase from CODE 1 to CODE 3 in all the frequency bands. The values of the upper whiskers follow the same trend all the frequency bands, with exception to the VLF band. The increase in the variability of the values from CODE 1 to CODE 3 is also marked by the increase of the IQRs in the LF, HF and $VHF_{[0.4\text{-}1.6]}$ bands. This suggests that the increase in the cognitive effort is translated in an increase of the dispersion of the pupil diameter frequency bands content and in the increase of the spectral power in these bands.

From these results, it is clear to conclude that the extracted indexes reflect the influence caused by the cognitive effort in the ANS. This is markedly clear in the higher frequency bands, where the separability between the REST and CODE groups is more visible, especially when considering the lack of overlap between the distributions corresponding to the REST group and the values above the 75$^{th}$ percentile corresponding to the CODE groups.

It is also possible to conclude that the increase in the level of cognitive effort caused the comprehension of CODE, whose complexity increases from CODE 1 to CODE 3 (despite this not being consensual among the subjects that analyzed the code), is translated in an increase in the maximum whiskers, in the outliers concentration/range in almost all the frequency bands. This observations are supported by the fact that the captured processes evolve over time and are not stationary.

The increase of the median values (see Figure 2) in these groups was also markedly visible in the HF and $VHF_{[0.4\text{-}1.6]}$ bands, suggesting the activity of the ANS can be better captured in the higher frequency bands, rather that the classic HRV ones,

when analyzing PD changes in the context of the present study. These findings are in agreement with the literature (on pupillography use not related to software), where it was reported an increase of the power spectra in the high frequency bands as a function of cognitive task difficulty [22], which are mostly associated with the activation of the parasympathetic afferent.

Globally, the pupillography results in the different frequency bands show a clear difference between the REST activities and the CODE activities, as well as distinct pupillography data between CODE 1 and the other two codes, and also some difference (clear in some bands) between CODE2 and CODE 3.

An interesting aspect is to compare pupillography results with the effort perceived by participants using an adapted version of the NASA-TLX survey, as shown in Table III.

TABLE III. EFFORT AND LOAD MEASURED USING NASA-TLX

| Prog | OE | | ME | | TP | | DI | |
|---|---|---|---|---|---|---|---|---|
| | Avg | Std | Avg | Std | Avg | Std | Avg | Std |
| C1 | 0,96 | 0,11 | 0,29 | 0,11 | 0,3 | 0,15 | 0,27 | 0,12 |
| C2 | 0,43 | 0,27 | 0,78 | 0,17 | 0,65 | 0,25 | 0,68 | 0,2 |
| C3 | 0,48 | 0,33 | 0,81 | 0,17 | 0,66 | 0,25 | 0,61 | 0,26 |

In the survey (step 6 of the protocol) each participant was first asked to describe the algorithm of each code C1 to C3, to assess how well the participant understood each program. The next questions of the survey are an adaptation of the NASA Task Load Index (TLX) to assess the participant perceived effort in the tasks. This data helps us to understand the subjective participant-perceived effort and relate that effort with the physiological response and with the objective success in understanding the programs. Table III summarizes the data collected from these surveys. Objective evaluation OE captures the average correctness of the answers related to the understanding of the programs (0 means complete failure in understanding the code and 1 a complete success). Mental effort (ME), Time Pressure (TP) and Discomfort (DI) were collected with a NASA-TLX enquiry (second survey). All values are from 0 to 1. Included are the average and standard deviation values.

The results of pupillography are consistent with the NASA-TLX results, suggesting that pupillography can be used to provide a real time measurement of programmers' effort and cognitive load during software development activities. The subjective effort measured by NASA-TLX, as well as the biofeedback pupillography results, show that the mental effort does not correlate well with complexity metrics. For example, the cyclomatic complexity of program C1 is higher than C2 (and C1), but both the pupillography and NASA-TLX show that participants see C2 as almost as complex as C3. This was somewhat expected, as it is well-known that complexity metrics cannot be used as predictor of programmers' effort in many scenarios.

## VI. UTILIZATION PERSPECTIVES AND FUTURE WORK

Our study shows that pupillography is a very promising biofeedback mechanism, providing accurate indications of programmers' mental effort and cognitive load. As explained, the experiments performed also included also eye tracking data (among other sensors), which allow us to identify the exact lines of code where the participant was looking at at any moment of the experiment. Thus, the immediate next step is to analyze the pupillography data at a low level of code granularity, combining the variations in the pupil size with the code lines where the participant was looking at, which are identified by the eye tracking. It is worth noting that the eye tracking device is actually the same device that provided the pupil size, so adding eye tracking data does not change the highly relevant non-intrusiveness of the method.

Forthcoming use of pupillography, combined with eye tracking, will be a key element in measuring programmers' cognitive stress and mental effort in real time, while producing code or reading code for software inspections. The meta-data about the programmers' cognitive state while dealing with a given code unit can be recorded and linked to the related program code lines, which will allow new visionary features such as:

- **Online advice to programmers and testers**, ranging from simple warning of software code areas that may need a second look at (to remove possible bugs) to more sophisticated programmers' support scenarios that consider the biofeedback metadata in conjunction with the complexity of the code handled by the programmer and her/his history of previous bugs (we call this **alter-pair programming**, as an analogy to the agile pair programming approach, but without the need of the second programmer of the pair).
- **Biofeedback improved models of bug density estimation and software risk analysis** through the use of code complexity metrics enhanced with programmer's biofeedback metadata that shows how complexity is really perceived by each programmer.
- **New biofeedback driven testing approaches** using the meta information about the programmers' cognitive and emotional states while working on the different portions of the code, combined with traditional complexity metrics, to guide the testing effort to the code units representing higher bug risk.
- **Programmers' friendly integrated development environments** with automatic warning/enforcement of programmers' resting moments, when accumulated signs of fatigue and mental strain show that not only the code quality is doubtful but, above all, programmers' mental well-being must be protected.
- **Biofeedback optimized training needs** through the creation of individual programmer's profiles to help define training plans based on the biofeedback metadata associated to individual programmers.

It is clear that pupillography should be complemented with other types of low intrusive sensors, such as HRV. A deeper knowledge on the error mechanisms at a neuroscience level is also quite important to remove noise and increase accuracy. In any case, the present results on the use of pupillography to measure programmers' mental effort and cognitive load suggest promising near future utilization of pupillography in software

development and testing. Low-cost eye tracking devices currently available (used by the videogame industry) opens new possibilities for the rapid adoption of pupillography in software engineering.

## VII. Conclusions

This practical experiment report presents new evidence showing that pupillography is an effective approach to measure programmers' mental effort and cognitive load in code comprehension tasks such as in code inspection scenarios, as well as during general code development activities. The fact that pupillography is non-intrusive from a physical point of view and can be easily adapted to current software development environments suggest that pupillography could be adopted as a biofeedback mechanism in the near future.

The results reported from the observational study including 30 experienced programmers shows that programmers' mental effort and cognitive load measured using pupillography is consistent with the subjective perception of complexity and load recorded by the programmers using NASA-TLX task load index. As somewhat expected, the analysis based on complexity metrics deviates from both results obtained from pupillography and NASA-TLX.

The paper ends with the discussion of future perspectives for the utilization of pupillography in the broader context of Biofeedback Augmented Software Engineering (BASE) approach.

## Acknowledgment

The authors would like to thank to the volunteers that participated in the experiments. This work was partially funded by the BASE project, POCI - 01-0145 - FEDER- 031581.

## References

[1] S. Shah, M. Morisio, M. Torchiano "The Impact of Process Maturity on Defect Density", ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, 2012.

[2] N. Honda, S. Yamada, "Empirical Analysis for High Quality SW Development", American Journal Op. Research, 2012.

[3] Steve McConnell, "Code Complete: A Practical Handbook of Software Construction", Microsoft Press, 2004.

[4] J. Durães and H. Madeira "Emulation of SW Faults: A Field Data Study and a Practical Approach", IEEE Transactions on SW Engineering, vol. 32, no. 11, pp. 849-867, November 2006.

[5] J. Christmansson and R. Chillarege, "Generation of an Error Set that Emulates SW Faults", Proc. of the 26th International Fault Tolerant Computing Symposium, FTCS-26, Sendai, Japan, 1996

[6] A. Fuqun Huang, B. Bin Liu, and C. Bing Huang, "A Taxonomy System to Identify Human Error Causes for Software Defects", 18th ISSAT International Conference on Reliability and Quality in Design, 2012

[7] Fuqun Huang, "Human Error Analysis in Software Engineering", chapter of the book "Theory and Application on Cognitive Factors and Risk Management", F. Felice and A. Petrillo Editors, IntechOpen, 2017.

[8] James Reason, "Human Error", Cambridge University Press, 1990.

[9] V. Anu, et. Al, "Using A Cognitive Psychology Perspective on Errors to Improve Requirements Quality: An Empirical Investigation", IEEE 27th International Symposium on Software Reliability Engineering, 2016.

[10] N. Peitek, J. Siegmund, et al., "A Look into Programmers' Heads", IEEE Transactions on Software Engineering, August, 2018.

[11] B. Floyd, T. Santander, and W. Weimer, "Decoding the Representation of Code in the Brain: An fMRI Study of Code Review and Expertise", ICSE 2017, pp 175–186, Piscataway, NJ, USA, 2017.

[12] T. Nakagawa, Y. Kamei, et al., "Quantifying Programmers' Mental Workload During Program Comprehension Based on Cerebral Blood Flow Measurement: A Controlled Experiment", Proc. of ICSE 2014.

[13] J. Castelhano, I. C. Duarte, C. Ferreira, J. Durães, H. Madeira, and M. Castelo-Branco, "The Role of the Insula in Intuitive Expert Bug Detection in Computer Code: An fMRI Study", Brain Imaging and Behavior, 2018.

[14] R. Couceiro, G. Duarte, J. Durães, J. Castelhano, C. Duarte, C. Teixeira, Miguel C. Branco, P. Carvalho, H. Madeira, "Biofeedback augmented software engineering: monitoring of programmers' mental effort", International Conference on Software Engineering, New Ideas and Emerging Results, ICSE 2019 (accepted)

[15] S. C. Müller and T. Fritz, "Using (Bio)Metrics to Predict Code Quality Online", Department of Informatics, University of Zurich, Switzerland, Proc. of 38th IEEE ICSE, 2016.

[16] S. Sirois and J. Brisson, "Pupillometry," Wiley Interdisciplinary Reviews: Cognitive Science, vol. 5, pp. 679-692, 2014.

[17] J. Beatty and B. Lucero-Wagoner, "The pupillary system," Handbook of psychophysiology, vol. 2, 2000.

[18] E. H. Hess and J. M. Polt, "Pupil size in relation to mental activity during simple problem-solving," Science, vol. 143, pp. 1190-1192, 1964.

[19] D. Kahneman and J. Beatty, "Pupil diameter and load on memory," Science, vol. 154, pp. 1583-1585, 1966.

[20] M. L. H. Võ, et al., "The coupling of emotion and cognition in the eye: Introducing the pupil old/new effect," Psychophysiology, vol. 45, pp. 130-140, 2008.

[21] J. Beatty, "Task-evoked pupillary responses, processing load, and the structure of processing resources," Psychological bulletin, vol. 91, 1982.

[22] H. Lüdtke, B. Wilhelm, M. Adler, F. Schaeffel, and H. Wilhelm, "Mathematical procedures in data recording and processing of pupillary fatigue waves," Vision research, vol. 38, pp. 2889-2896, 1998.

[23] M. Nakayama and Y. Shimizu, "Frequency analysis of task evoked pupillary response and eye-movement," in Proceedings of the 2004 symposium on Eye tracking research & applications, pp. 71-76, 2004.

[24] A. Murata and H. Iwase, "Evaluation of mental workload by fluctuation analysis of pupil area," in Engineering in Medicine and Biology Society, 1998. Proceedings of the 20th Annual International Conference of the IEEE, pp. 3094-3097, 1998.

[25] V. Peysakhovich, M. Causse, S. Scannella, and F. Dehais, "Frequency analysis of a task-evoked pupillary response: Luminance-independent measure of mental effort," International Journal of Psychophysiology, vol. 97, pp. 30-37, 2015.

[26] O. Salem, L. Yaning, and A. Mehaoua, "A lightweight anomaly detection framework for medical wireless sensor networks," in Wireless Comm. and Networking Conference (WCNC), 2013 IEEE pp. 4358-4363, 2013.

[27] D. Kondrashov and M. Ghil, "Spatio-temporal filling of missing points in geophysical data sets," Nonlinear Processes in Geophysics, vol. 13, pp. 151-159, 2006.

[28] R. Sassi, V. D. Corino, and L. T. Mainardi, "Analysis of surface atrial signals: time series with missing data?," Annals of biomedical engineering, vol. 37, pp. 2082-2092, 2009.

[29] F. Onorati, M. Mauri, V. Russo, and L. Mainardi, "Reconstruction of pupil dilation signal during eye blinking events," in Proceeding of the 7th International Workshop on Biosignal Interpretation, pp. 117-120, 2012.

[30] A. Eleuteri, A. C. Fisher, D. Groves, and C. J. Dewhurst, "An efficient time-varying filter for detrending and bandwidth limiting the heart rate variability tachogram without resampling: MATLAB open-source code and internet web-based implementation," Computational and mathematical methods in medicine, vol. 2012, 2012.

[31] M. Malik, "Heart rate variability: Standards of measurement, physiological interpretation, and clinical use," European Heart Journal, vol. 17, pp. 354–381, 1996.

[32] Burg, J.P. "Maximum Entropy Spectral Analysis", Proceedings of the 37th Meeting of the Society of Exploration Geophysicists, 1967.

[33] F. M. Villalobos-Castaldi, J. Ruiz-Pinales, N. C. K. Valverde, and M. Flores, "Time-frequency analysis of spontaneous pupillary oscillation signals using the Hilbert-Huang transform", Biomedical Signal Processing and Control, vol. 30, pp. 106-116, 2016.