



Temas em Psicologia

ISSN: 1413-389X

comissaoeditorial@sbponline.org.br

Sociedade Brasileira de Psicologia
Brasil

dos Santos, Natanael Antonio; de Arruda Campos Neto, Armindo; Marinho de Sousa, Bruno; Cruz Pessoa, Ediko Dominike; Toscano Barreto Lyra Nogueira, Renata Maria
Matlab and Eye-Tracking: Applications in Psychophysics and Basic Psychological Processes

Temas em Psicologia, vol. 22, núm. 3, diciembre, 2014, pp. 589-601
Sociedade Brasileira de Psicologia
Ribeirão Preto, Brasil

Available in: <http://www.redalyc.org/articulo.oa?id=513751529006>

- How to cite
- Complete issue
- More information about this article
- Journal's homepage in redalyc.org

redalyc.org

Scientific Information System

Network of Scientific Journals from Latin America, the Caribbean, Spain and Portugal

Non-profit academic project, developed under the open access initiative

Matlab and Eye-Tracking: Applications in Psychophysics and Basic Psychological Processes

Natanael Antonio dos Santos¹

*Departamento de Psicologia da Universidade Federal da Paraíba, João Pessoa,
Paraíba, Brasil*

Armando de Arruda Campos Neto

*Instituto Federal de Educação, Ciência e Tecnologia de Mato Grosso, Cuiabá,
Mato Grosso, Brasil*

Bruno Marinho de Sousa

Universidade Federal de Goiás – Regional Catalão, Catalão, Goiás, Brasil

Ediko Dominike Cruz Pessoa

Centro Universitário de João Pessoa, João Pessoa, Paraíba, Brasil

Renata Maria Toscano Barreto Lyra Nogueira

*Departamento de Psicologia da Universidade Federal de Pernambuco, Recife,
Pernambuco, Brasil*

Abstract

This article aims to demonstrate the utilities of Matlab (MATrix LABoratory) and eye-tracker for research in psychophysics and basic processes. We first introduce its origin, functioning, and advantages over other programs, and then we review different applications of Matlab in psychophysical tests involving eye-trackers. As an example, we present the development of a psychophysical test involving eye-tracker and Matlab, showing its ability to communicate and interact with other devices and programming languages. Finally, we emphasize the advantages and difficulties of working with Matlab.

Keywords: Psychophysics, Matlab, Eye-tracker, technological innovation.

Matlab e Rastreamento Ocular: Aplicações em Psicofísica e Processos Psicológicos Básicos

Resumo

Esse artigo teve como objetivo demonstrar a funcionalidade e capacidade do Matlab (MATrix LABoratory) e rastreamento de movimentos oculares (*eye-tracker*) para pesquisas em psicofísica e processos básicos. Para isso é apresentado primeiramente os fundamentos da criação e funcionamento do Matlab e suas vantagens sobre outros *softwares*. Em seguida são apresentadas diferentes aplicações do Matlab em testes psicofísicos com rastreadores de movimentos oculares (*eyetracker*). Também é apresentada a criação de um teste psicofísico em Matlab para um *eyetracker* e sua capacidade de se

¹ Mailing address: Centro de Ciências Humanas Letras e Artes, Departamento de Psicologia, Laboratório de Percepção, Neurociências e Comportamento, Universidade Federal da Paraíba, Campus I, Cidade Universitária, João Pessoa, PB, Brasil 58051-900. E-mail: natanael_labv@yahoo.com.br, armindocampos@ibest.com.br, sousabm@hotmail.com, ediko.dominike@gmail.com and rm_toscano@yahoo.com.br

Acknowledgement: National Council for Scientific and Technological Development (CNPq).

comunicar/interagir com outras linguagens. Por fim, são enfatizadas as vantagens e dificuldades de se trabalhar com esse tipo de ferramenta.

Palavras-chave: Psicofísica, Matlab, *Eye-tracker*, inovação tecnológica.

Matlab y Rastreo Ocular: Aplicaciones en la Psicofísica y Procesos Psicológicos Básicos

Resumen

El objetivo principal de este trabajo fue exponer algunas de las aplicaciones que tiene el *software Matlab* (MATrix LABoratory) y *Eyetracker* en las investigaciones con psicofísica y procesos básicos. Para alcanzar este objetivo, el artículo se organiza en cuatro secciones. La primera aborda los aspectos básicos de la creación y gestión del *software*. La segunda sección expone sus diferentes aplicaciones en las pruebas psicofísicas. La tercera parte muestra un ejemplo de cómo crear una prueba psicofísica con *eye-tracker* programado por *Matlab*. Por último, se destacan las ventajas y dificultades de trabajar con este *software*.

Palabras clave: Psicofísica, Matlab, *Eye-tracker*, innovación tecnológica.

Neuroscience is currently among the scientific disciplines most esteemed and requested to address several questions regarding physical and mental aspects of human nature. This is mainly due to its reliance on new mathematical and computational tools. In the last decades, such tools have supported biological neuroscientific research. At first, they primarily had a methodological function in data analysis, for example. Later on, they were also responsible for conceptual advances that shaped modern neuroscience (Wallish et al., 2009).

Several quantitative methods applied in neuroscience were developed by pioneers from different academic fields such as physics, engineering, mathematics, statistics, and computer science. Although subject to criticisms, quantitative methods have positive effects on many branches such as molecular and cognitive neuroscience. Today, it is mandatory to know computational tools in order to understand and discuss state-of-the-art neuroscience (Wallish et al., 2009).

Psychobiology is a branch of neuroscience that frequently uses computational tools. In particular, psychophysics employs computer programs to control experimental procedures and stimuli characteristics in the study of sensation

and perception. Different psychophysical methods have been created to address several questions regarding stimuli magnitudes and sensorial thresholds. Research on visual perception, for example, is a leading area with numerous studies and advances in image processing technologies.

Visual information is quickly processed by the brain, and vision constitutes a robust component of the sensory system. In addition, the visual system is crucial to human perception and has an important role in information processing underlying cognitive processes (Alexandre & Tavares, 2007).

In the field of psychophysics, researchers develop tests of visual perception by using chromatic and achromatic images, eye-tracking and electrophysiological techniques, among others. Such tests are based on new technologies and resources that provide time precision of stimulus presentation and response registration, stimuli synchronization, and data analysis (Kingdom & Prins, 2010).

Several computer software programs have been used in psychophysical research, such as SuperLab (Abboud, Schultz, & Zeitlin, 2006) and E-Prime (Schneider, Eschman, & Zuccolotto, 2002). In the case of SuperLab, version 4.0 has a user-friendly interface divided into

lists of blocks, trials, and events that can be easily handled by the researcher. It supports different image formats (JPG, BMP, TIFF, PNG, and GIF), video (AVI and MPG), and audio files. In addition, SuperLab creates a data output file with several parameters regarding participant's responses, which can be further analyzed by another program.

E-Prime, on the other hand, has a more advanced platform than SuperLab, and it comprises five modules: E-Studio for experiment design and programming, E-Run to execute the experimental procedures, and other three modules dedicated to data management and recovery. In addition, E-Prime has an internal scripting language derived from Visual Basic, and it does not require advanced knowledge of programming. Inside the E-Studio environment, there is a toolbar with several drag-and-drop items, which are the building blocks of an experiment (e.g., image display, text display, sound presentation, parameters list, etc.).

Both SuperLab and E-Prime allow the use of external materials (e.g., sounds, images, and videos), and can control stimuli duration with millisecond precision, but E-Prime is more precise and accurate than SuperLab. However, both programs have limitations. For example, it is not possible to implement concurrent tasks (that is, two events running in parallel) or more sophisticated procedures, such as psychophysical adaptive methods, which require stimuli adjustments at runtime according to the participant's previous response. In order to overcome such limitations, it is preferable to use a flexible programming environment such as Matlab (Stahl, 2006), which is a powerful tool to implement sophisticated psychophysical tests, as well as perform advanced statistical analysis.

In this paper, our aim is to introduce some uses and applications of Matlab in psychophysics and research on basic psychological processes, such as perception, attention, and learning. As an example, we present a Matlab interface to control an eye-tracker device and visual stimuli presentation in order to show its potential applications in different studies in psychophysics.

Matlab

Matlab (MATrix LABoratory) is a high-performance, interactive software specialized in numerical computing, data visualization, and analysis. It is a suitable and robust tool for developing applications in several areas.

Matlab was initially developed by Cleve Moler in the 1970s to provide a simple and interactive way of programming – the codes do not require compilation and the instructions are executed at runtime. In addition, it allows operations such as variable values changing and saving into a file, providing a flexible computing environment (Wallish et al., 2009). Although the initial version was focused on calculus, several packages of functions and libraries have been developed, and today Matlab is a useful tool in practically any scientific field.

The strong points of Matlab include graphical resources, powerful programming tools, advanced algorithms, and the large number of internal functions and libraries (e.g., statistics, optimization, image processing, neural networks, etc.). Matlab can process large data files and perform fast linear algebra operations. Other useful resources include graphical functions to visualize several types of data and simulation results (Wallish et al., 2009). Furthermore, Matlab can integrate codes written in other programming languages such as C, Fortran, and Java.

In summary, Matlab has several sophisticated resources to address computational and analytical problems in a variety of domains, including psychophysics. However, Matlab is difficult for beginners and it is still rarely used outside the hard sciences and engineering. For example, a survey with experimental psychologists showed that Matlab was used by 2.7% of the sample, much less than DMDX (17.2%) and E-Prime (30.6%; Ferreira, n.d.).

Matlab Applications in Psychophysics

Matlab is a versatile tool for psychophysical research and can be applied to develop tests of auditory and visual perception, as well as to investigate other basic processes such as attention,

learning, and memory. For example, Matlab 7.0 was used in the Brazilian validation of the Dichotic Sentence Identification (DSI; Andrade, Gil, & Iório, 2010), a behavioral auditory test designed to assess central auditory function in individuals with peripheral hearing loss (Fifer, Jerger, Berlin, Tobey, & Campbell, 1983). In recent years, Matlab has been increasingly used in many studies of visual perception.

The human visual system is equipped with the most sophisticated circuits to interpret the visual world. For more than one century, scientists have been fascinated by questions regarding when, where, and how the human eye moves to gather information of the environment (Caldara & Miellet, 2011). One common procedure to address such interesting questions is the integration between Matlab and eye-tracker, which is an efficient device for studying eye movements.

In recent psychophysiological research, the eye-tracking technique has been widely used for collecting eye movement data to investigate the cognitive processes underling visual behavior (Berger, Winkels, Lischke, & Höppner, 2012), and Matlab has been widely used to create new tests in this field.

In order to understand psychophysical tests and the eye-tracking technique, it should be noted that eye movements have two basic components: fixations (brief stops) and saccades, that is, quick eye movements from one fixation point to another. The sensorial input of visual information occurs during eye fixations, whereas input is usually suppressed during saccadic movements (Martin, 1974).

Eye movements are strictly associated with visual attention, and many studies have been analyzing the target areas of saccadic movements to investigate covert attention. Some studies have shown that attention is always oriented towards the direction of saccadic movement (Berger et al., 2012). Thus, both fixations and saccades are crucial variables to be recorded, processed, and analyzed by programs written in Matlab.

In this field, van Beilen, Renken, Groenewold, and Cornelissen (2011) used Matlab as a tool for programming an eye-tracker to investigate attentional windows, that is, a specific re-

gion in the visual field to where we first direct our attention, and then our eyes. This study was based on the assumption that attention may be directed outwards the fixation point, a low-resolution area in peripheral vision, in order to support the planning of future saccadic movements towards another region or object (van Diepen & d'Ydewalle, 2003). This property is assumed to be inherent to human beings and crucial in dangerous situations (Graupner, Pannash, & Velichkovsky, 2011).

The test devised by van Beilen et al. (2011) was based on a psychometric function developed in Matlab by using the Eyelink Toolbox extension and the Eyelink 1000 eye-tracker (SR Research Ltd., Mississauga, Ontario, Canada). The participants observed a series of stimuli with different levels of congruence, and the test aimed at verifying possible changes in the attentional window (and eye saccades), as well as the influence of participants' expectations. The test revealed an association between accuracy in detecting stimuli (identification, reaction time, and saccadic latencies) and the size of the attentional window. In addition, the psychometric function developed in Matlab demonstrated that attentional windows can be modulated according to individual and voluntary behavioral goals.

Another study used Matlab and eye-tracker to investigate whether saccadic movements vary according to different regions of the visual field. Jóhannesson, Ásgeirsson and Kristjánsson (2012) conducted seven experiments using the Psychtoolbox (a Matlab extension used for stimuli presentation) and the Video Eyetracker Toolbox of the monocular 250 Hz CRS Eye-tracker (Cambridge Research System Ltd). The stimuli were presented at the left and at the right side of the screen with amplitudes from 5° to 10° of visual angle, at a distance of 53 cm from the participant's eye. The objective was to find significant differences in saccadic movements between the stimuli presented on nasal (5°) and temporal (10°) visual hemifields, but this study did not find asymmetries in saccadic latencies.

The integration between Matlab and eye-tracker supports the development of psychophysical tests that advance knowledge on visual

and auditory perception. As an example, we can mention the study by Leo, Romei, Freeman, Ladavas and Driver (2011) that used Gabor orientations at maximum contrast as visual stimuli, each one randomly presented for 250 ms at the left or right of the central fixation point. Concurrently to visual stimuli, static auditory stimuli were presented either in the same direction, or in the opposite direction of the visual stimuli. The eye movements were recorded by the CRS 250 Hz Eye-tracker controlled by Matlab and Psychtoolbox. In addition, a customized toolbox written for Matlab processed the data and computed the standard percentage of correct responses for each visual hemifield. The results revealed that eminent sounds may enhance visual processing.

The Matlab and eye-tracker integration may also bring important contributions in the field of learning, such as the study by Takeuchi, Puntous, Tuladhar, Yoshimoto and Shirama (2011) that estimated the mental effort by measuring pupil diameter changes. The authors used Matlab, Psychtoolbox and the EyeTracker ViewPoint 220 fps USB system (Arrington Research Inc.). A code written in Matlab generated the visual stimuli by combining spatial orientation and frequency defined by the Gabor function. During a visual learning task, the pupil behavior was registered. In brief, the results suggested that learning by visual perception requires mental effort, and the amount of effort is associated not only with behavioral performance, but also with autonomic responses such as pupil diameter.

Matlab also has an important role in studies of color and movement perception, which are topics of great importance to psychophysical research given that movement and color integration is required in many everyday life situations. The processing of colors and movements are functionally and anatomically dissociated: although both color and movement information processing begin in the retina, color information follows a ventral visual pathway (also called parvocellular pathway), whereas spatial and movement information follow a dorsal (or magnocellular) pathway (Tchernikov & Fallah, 2010).

In order to investigate the independence between color and movement processing, Tcher-

nikov and Fallah (2010) used Matlab and the eye-tracker Eyelink II 500 Hz (SR Research) to measure eye movements at a distance of 57 cm from the computer monitor. A photometer (SpectraScan PR 655, Optikon Corp.) measured the color coordinates of each color used in the test. The results showed that achromatic movement processing is intrinsically modulated by color.

The studies reviewed above indicate that Matlab is a versatile and robust tool in psychophysics and research on basic psychological processes, particularly for developing tests of visual and auditory perception. Although such tests were developed in laboratories, Matlab has been used even in surgical interventions. For example, Matlab was used to monitor an Electroencephalography (EEG) and ensure an ideal blood flow during the removal of an atherosclerosis plaque that was blocking the patient's artery (Accardo, Cusenza, & Monti, 2009).

Today, Matlab is still evolving due to the development and update of software packages (or toolboxes) to support data analysis and visualization. GazeAlyze is among the most used applications written for Matlab. The GazeAlyze software (Mathworks Inc., Natick, MA) was developed to generate and present static and dynamic stimuli, create databases, and make different types of analysis of eye movement patterns. For example, it has functions for pattern detection and filtering, event detection, selection of regions of interest, creation of new spreadsheets for statistical analyses, and a set of methods for data visualization and fixation heat maps. GazeAlyze has pre-processing and event detection functions that are based on ILAB, an application that uses Matlab functions to analyze eye-tracker data (Berger et al., 2012).

Berger et al. (2012) used GazeAlyze and Matlab to validate a face recognition test, which was designed to investigate the skills involved in recognizing emotional facial expressions. As stimuli, they used faces with negative and positive valences taken from the FACES database (Ebner, Riediger, & Lindenberger, 2010). The results showed that eye fixations were mainly directed towards the faces' eyes, in spite of emotional valence.

Another example of the evolution of Matlab applications is the i-Map software (available at www.unifr.ch/psycho/ibmlab/), which is an open and editable toolbox developed as an alternative method to compute fixation maps (Caldara & Miellet, 2011). Based on a technique derived from functional magnetic resonance, eye fixation data are transformed by Gaussian convolution kernel to generate tridimensional fixation maps. The study by Caldara and Miellet (2011) illustrates different steps of i-Map use and also presents some important features, such as analysis of real eye and face movement data, visual scene perception, and memory.

The Visual Maze Test

In this section, we present an eye-tracker interface using Matlab (see the Appendix for the complete source code), given that both SuperLab and E-Prime do not support our eye-tracker device. In the Laboratory of Perception, Neuroscience and Behavior (LPNeC) at the Federal University of Paraíba, we have developed a visual maze test to investigate the cognitive processes involved in the visual behavior of finding the way out of a maze. The task is relatively simple

and was devised to monitor the participant's eye movements, that is, eye fixations and saccades during the task. We developed this test because such measures seem more resistant to psychosocial factors and individual differences. For example, schooling is often a significant factor in neurocognitive tests that require recognition of numbers or letters, as in the case of the Trail Making Test (Reitan, 1958).

Method

The Visual Maze

We designed a rectangular visual maze with the start point 'A' at the screen center and four end points 'B', one at each corner, with distribution of symmetrical pairs (pair Type I and Type II) in order to reduce "chance" factors in the test (Figure 1). Symmetrical pairs are important when the participant fails in the first attempt and has to try again. That is, if the participant fails in a pathway Type I, then the experimenter may instruct the participant to try a pathway Type II. In addition, dashed lines indicate the pathways inside the maze to guide eye movements and reduce participants' anxiety. The maze was saved as an image file in the BMP 8 bits format.

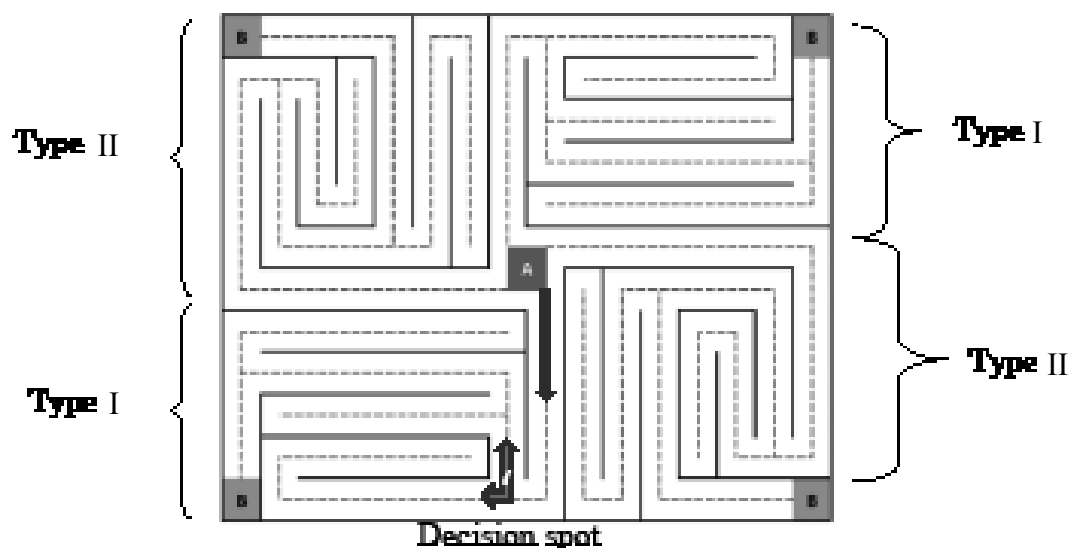


Figure 1. Maze created for the test with one spot for decision making.

It should be noted that in Figure 1, the maze has some decision spots, that is, the points where participants have to decide which route to fol-

low. The fixation time tends to be longer at such points, signaling the cognitive processes involved in decision making.

Equipment

We used the Cambridge Research System (CRS) Eye-tracker with sampling frequency set at 250 Hz, one personal computer, two computer monitors, and the CRS ViSaGe graphic system. The experimental procedures and the eye-tracker

interface were developed using Matlab version 7.2, and the programming was based on the important parameters described in Table 1. In addition, we developed the Saccades Eyetracker Translator, an auxiliary program written in Java to extract some measures of interest, such as fixation time (duration) and number of saccades.

Table 1
Initial Parameters and Specifications for Matlab Programming

Parameters	Specifications
Saccade	Fixation in a circular area with a ray of 20 mm in 100 milliseconds.
Display period	Open to terminate with a touch on any key on the keyboard.
Track saving period	50 seconds in the mimic screen (small screen for following the test).
Participant distance	570 mm matching the equivalent distance to a cycle per vision degree.

Procedure

The participant was instructed to depart from point A and arrive at point B as fast as possible, following the dashed line. In the case of following a wrong track (a dead end route), the participant had to follow the dashed lines back to the decision point and then follow the other way to arrive at point B.

The testing session began with the eye-tracker calibration, followed by stimulus presentation and task execution. In order to finish the testing session, the experimenter had to press any key as fast as possible when the participant reached point B.

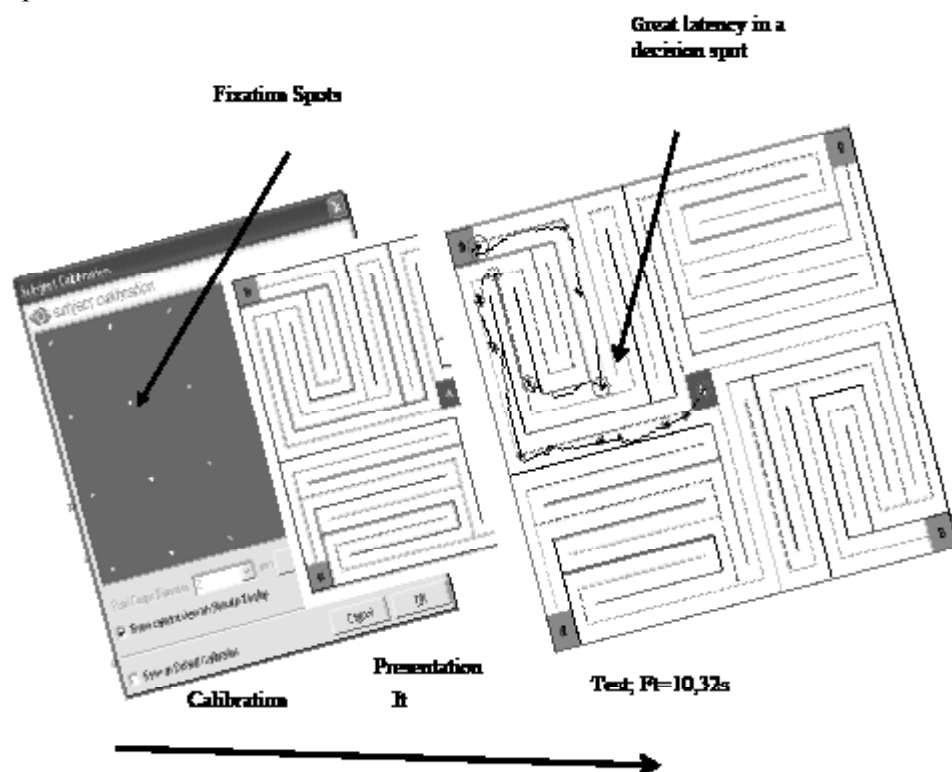


Figure 2. Maze test presentation sequence with 15 saccades and the test duration from It (initial time) until Ft (final time) of 10,32 seconds.

Calibration

Given that the eye-tracker is a device that follows the participant's eyes, it is necessary to synchronize the recording of participant's eye fixations with the stimuli positions in the screen. During the calibration process, a "mimic" screen is used and the participant is required to look at a series of random luminous points presented in the screen (Figure 2). The eye-tracker calibrates the participant's eye positions with the corresponding points, and after the last point, the experimenter may repeat some points that eventually did not get calibrated; these points are marked in green by the eye-tracker. Once all points are correctly calibrated (without green lines), a calibration test is performed by requesting that the participant look at other luminous points in the screen. Finally, the calibration settings are saved and the visual maze test begins.

Results

The program written in Matlab generates two outputs when the test finishes: an image combining the visual maze with the participant's eye fixations and saccades, and a text file called "myResults" containing large amounts of data (like machine code) recorded by the eye-tracker device. This file, however, is difficult to handle because it is very large and lacks adequate tabulation. Therefore, we used the Eyetracker Saccades Translator written in Java to extract the relevant data from the "myResults" file. The program source codes are available in the following link: <https://github.com/DominikeCruz/>. This program is easy to use: double-click on its icon and select the file "myResults" for the conversion process, as shown in Figure 3. Besides extracting the relevant data, the program organizes the results of saccades, fixations, and testing time, as shown in Table 2.

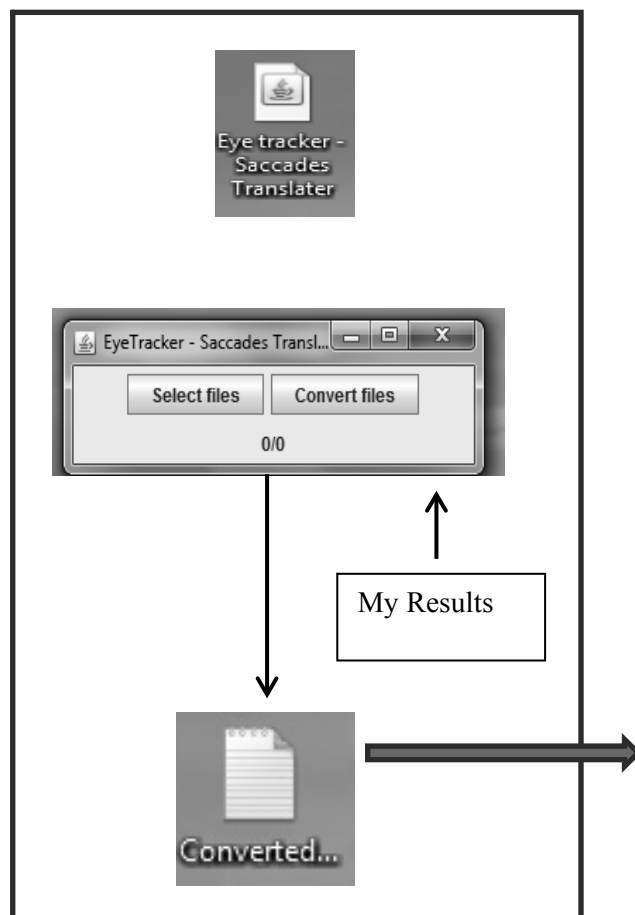


Figure 3. Flowchart of the Eyetracker Saccades Translator functioning.

Table 2

Saccades and Fixations

Saccades: time in milliseconds

Saccades 1: 864
 Saccades 2: 600
 Saccades 3: 488
 Saccades 4: 404
 Saccades 5: 224
 Saccades 6: 184
 Saccades 7: 584
 Saccades 8: 68
 Saccades 9: 704
 Saccades 10: 864
 Saccades 11: 820
 Saccades 12: **1156**
 Saccades 13: 120
 Saccades 14: 540
 Saccades 15: 1268

Total time: seconds/milliseconds

Total time: 10.32/10320

Figure 4 shows the relation between the number of saccades and the time to complete the maze (depicted in Figure 1) with the dominant eye. The number of saccades ranged from 11 to 42 ($M = 22.3$, $SD = 10.13$), and the time to complete the maze ranged from 3.8 to 21.0 s ($M = 10.3$, $SD = 5.13$). There is a strong correlation

between the number of saccades and the time to complete the maze with the dominant eye ($\rho = .99$, $\alpha < .001$), indicating that the longer the time to complete the maze, the greater the number of saccades. As expected, these results demonstrate that the application works with adult volunteers.

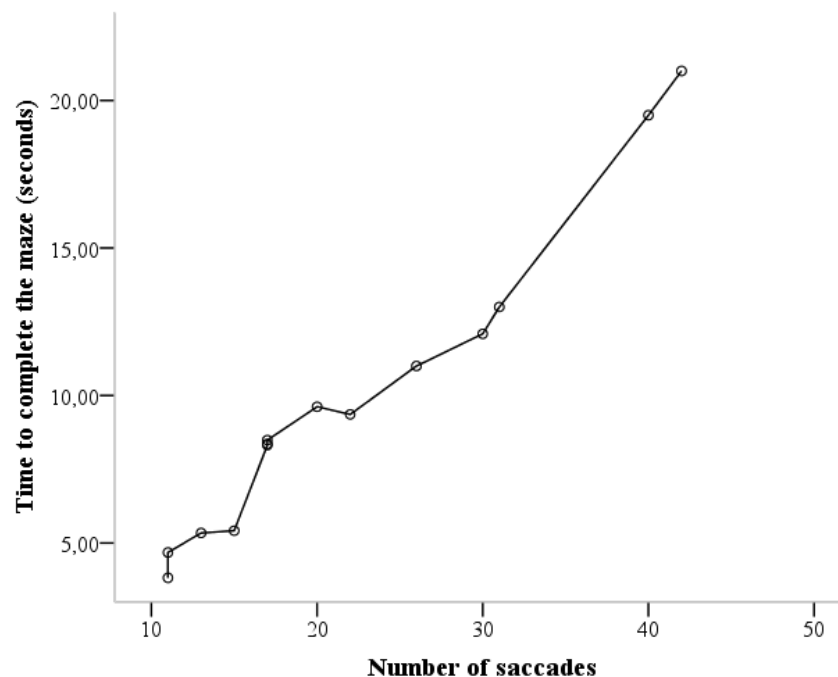


Figure 4. Time to complete the maze as a function of the number of saccades of the dominant eye of 14 volunteers aged 23 to 47 years ($M = 31.8$, $SD = 8.3$).

Note. Study approved by the Research Ethics Committee (CAAE subscription number 06746612.4.0000.5188).

Test Interpretation

As we have mentioned, there are two crucial parameters in eye-tracking experiments: fixations and saccadic movements (or saccades). The fixations are quick, momentary stops over a particular area, in general lasting less than 100 ms. On the other hand, saccadic eye movements occur between fixations, at a frequency of two to four per second (Cabestrero, Conde-Guzón, Crespo, Grzib, & Quirós, 2005; Riggs, Ratliff, Cornsweet, & Cornsweet, 1953).

Eye movements provide information about the participant's visual strategy, for example, the direction and duration of the eye gaze. A large number of saccades suggests a reduced control over the attentional window. In addition, it in-

dicates a narrow low-resolution, peripheral perception area outside the fixation point – that is, little attention devoted to plan the next saccades in the test. Therefore, in this case, the participant needs more saccades to cover a larger area of attention and planning. In other cases, longer fixation times may indicate slower cognitive processing during visual perception and may suggest congenital or acquired problems.

Decision points are usually followed by longer fixation times, when visual perception is suppressed in favor of cognitive processes (Martin, 1974). As shown in Table 2, the twelfth saccade clearly shows a longer latency over a decision point, which may indicate cortical processing in the area related to decision-making.

Final Considerations

The application presented in this paper shows that Matlab can interact with other resources, such as an eye-tracker and other programming languages (like Java), for the development and execution of psychophysical tests. In addition, Matlab is able to perform simple procedures like stimuli presentation and response registration, as well as complex procedures such as controlling an eye-tracker and generating any stimuli described by a mathematical function, which are useful resources for psychophysics and perception.

Many researchers consider Matlab as an efficient tool to investigate basic psychological processes, such as in studies of attention (van Beilen et al., 2011), learning (Takeuchi et al., 2011), movement perception (Tchernikov & Fallah, 2010), facial expressions (Berger et al., 2012), and so on.

In Brazil, there is an increasing tendency to use Matlab in neuroscientific research, as can be observed in the use of GazeAlyze and i-Map. In particular, Matlab may easily introduce the use of new technologies in the study of psychophysics and basic processes, as well as support the development of tools to instruct undergraduate and graduate students.

Finally, in order to support the development of new technologies in psychology and psychophysics, it would be positive to have introductory courses related to computer science at both undergraduate and graduate levels in the fields related to basic psychological processes. This is a necessary step towards instructing new generations of researchers adapted to develop and advance new technologies in this field. There are some textbooks that can guide readers in the adventure of learning Matlab (Borgo, Soranzo, & Massimo, 2012; Rosenbaum, 2011).

References

- Abboud, H., Schultz, W., & Zeitlin, V. (2006). SuperLab (Version 4.0) [Computer software]. San Pedro, CA: Cedrus Corporation.
- Accardo, A., Cusenza, M., & Monti, F. (2009). Linear and non-linear parameterization of EEG during monitoring of carotid endarterectomy. *Computers in Biology and Medicine*, 39(6), 512-518. doi:10.1016/j.combiomed.2009.03.003
- Alexandre, D. S., & Tavares, J. M. R. S. (2007). *Factores da percepção visual humana na visualização de dados*. Trabalho apresentado no Congresso de Métodos Numéricos em Engenharia, XXVIII Congresso Ibero Latino-Americano sobre Métodos Computacionais em Engenharia, Porto, Portugal.
- Andrade, A. N. de, Gil, D., & Iório, M. C. M. (2010). Elaboração da versão em Português Brasileiro do Teste de Identificação de Sentenças Dicóticas (DSI). *Revista da Sociedade Brasileira de Fonoaudiologia*, 15(4), 540-545. doi:http://dx.doi.org/10.1590/S1516-80342010000400011
- Berger, C., Winkels, M., Lischke, A., & Höppner, J. (2012). GazeAlyze: A MATLAB toolbox for the analysis of eye movement data. *Behavior Research Methods*, 44(2), 404-419. doi:10.3758/s13428-011-0149-x
- Borgo, M., Soranzo, A., & Massimo, G. (2012). *MATLAB for psychologists*. New York: Springer. doi:10.1007/978-1-4614-2197-9
- Cabestrero, R., Conde-Guzón, P. A., Crespo, A., Grzib, G., & Quirós, P. (2005). *Fundamentos psicológicos de la actividad cardiovascular y oculomotora*. Madrid, España: Universidad Nacional de Educación a Distancia Ediciones.
- Caldara, R., & Mielliet, S. (2011). iMap: A novel method for statistical fixation mapping of eye movement data. *Behavior Research Methods*, 43(3), 864-878. doi:10.3758/s13428-011-0092-x
- Ebner, N. C., Riediger, M., & Lindenberger, U. (2010). FACES - A database of facial expressions in young, middle-aged, and older women and men: Development and validation. *Behavior Research Methods*, 42(1), 351-362. doi:10.3758/BRM.42.1.351
- Ferreira, V. (Ed.). (n.d.). *Behavioural software survey*. Retrieved August 18, 2013, from <http://imaging.mrc-cbu.cam.ac.uk/imaging/BehavioralSoftwareSurvey>
- Fifer, R. C., Jerger, J. F., Berlin, C. I., Tobey, E. A., & Campbell, J. C. (1983). Development of a dichotic sentence identification test for hearing-impaired adults. *Ear and Hearing*, 4(6), 300-305.

- Graupner, S. T., Pannash, S., & Velichkovsky, B. M. (2011). Saccadic context indicates information processing within visual fixations: Evidence from event-related potentials and eye-movements analysis of the distractor effect. *International Journal of Psychophysiology*, 80(1), 54-62. doi:10.1016/j.ijpsycho.2011.01.013
- Jóhannesson, O. I., Ásgeirsson, A. G., & Kristjánsson, A. (2012). Saccade performance in the nasal and temporal hemifields. *Experimental Brain Research*, 219(1), 107-120. doi:10.1007/s00221-012-3071-2
- Kingdom, F. A. A., & Prins, N. (2010). *Psychophysics: A practical introduction*. New York: Elsevier Academic-Press.
- Leo, F., Romei, V., Freeman, E., Ladavas, E., & Driver, J. (2011). Looming sounds enhance orientation sensitivity for visual stimuli on the same side as such sounds. *Experimental Brain Research*, 213(2-3), 193-201. doi:10.1007/s00221-011-2742-8, 2011
- Martin, E. (1974). Saccadic suppression: A review and an analysis. *Psychological Bulletin*, 81(12), 899-917.
- Reitan, R. M. (1958). Validity of the Trail Making test as an indicator of organic brain damage. *Perceptual and Motor Skills*, 8(3), 271-276.
- Riggs, L. A., Ratliff, F., Cornsweet, J. C., & Cornsweet, T. N. (1953). The disappearance of steadily fixated visual test objects. *JOSA*, 43(6), 495-500.
- Rosenbaum, D. A. (2011). *MATLAB for behavioral scientists*. Mahwah, NJ: Psychology Press.
- Schneider, W., Eschman, A., & Zuccolotto, A. (2002). *E-Prime reference guide*. Pittsburgh, PA: Psychology Software Tools.
- Stahl, C. (2006). Software for generating psychological experiments. *Experimental Psychology*, 53(3), 218-232. doi:10.1027/1618-3169.53.3.218
- Takeuchi, T., Puntous, T., Tuladhar, A., Yoshimoto, S., & Shirama, A. (2011). Estimation of mental effort in learning visual search by measuring pupil response. *PLoS ONE*, 6, e21973. doi:10.1371/journal.pone.0021973
- Tchernikov, I., & Fallah, M. (2010). A color hierarchy for automatic target selection. *PLoS ONE*, 5, e9338. doi:10.1371/journal.pone.0009338, 2010
- Van Beilen, M., Renken, R., Groenewold, E. S., & Cornelissen, F. W. (2011). Attentional window set by expected relevance of environmental signals. *PLoS One*, 6, e21262. doi:10.1371/journal.pone.0021262
- Van Diepen, P. M., & d'Ydewalle, G. (2003). Early peripheral and foveal processing in fixations during scene perception. *Visual Cognition*, 10(1), 79-100. doi:10.1080/13506280143000023
- Wallish, P., Lusignan, M., Benayoun, M., Baker, T. I., Dickey, A. S., & Hatsopoulos, N. G. (2009). *MATLAB for neuroscientists: An introduction to scientific computing in MATLAB*. Waltham, MA: Academic Press.

Received: April, 02, 2013

1st revision: August, 13, 2013

2nd revision: October, 02, 013

Accepted: October, 14, 2013

Appendix

We developed a program in Matlab to control every phase of the testing session, such as: activation of the Cambridge library, ViSaGe integration, eye-tracker functioning, maze presentation, and data recording. The complete source code is presented below.

Cambridge Library Activation

```
global CRS;
crsLoadConstants;
CheckCard = crsGetSystemAttribute(CRS.DEVICECLASS);
if(CheckCard ~= 7)
error('Sorry, this demonstration requires a VSG ViSaGe.');
```

```
end;
```

Vet Configuration (VideoEyetracker)

```
vetSetStimulusDevice(CRS.deVSG);
errorCode = vetSelectVideoSource(CRS.vsUserSelect);
if(errorCode<0); error('Video Source not selected.');
```

```
end;
```

```
ImageFile = which('C:\Documents and Settings\Natanael Santos\Desktop\Armando DO\labyrintho_toxicologia.bmp');
```

```
vetClearAllRegions;
```

```
vetClearDataBuffer;
```

```
vetClearMimicscreenBitmap;
```

```
vetCreateCameraScreen;
```

```
vetCreateMimicScreen;
```

```
vetSetMimicScreenDimensions(300, 0, 300, 266);
```

```
vetSetMimicPersistence(50); (mimic screen activation time in seconds)
```

```
vetSetMimicPersistenceStyle(CRS.psConstant); vetSetMimicPersistenceType(CRS.ptMotionAndFixations); vetLoadBmpFileToMimicScreen(ImageFile,1);
```

```
vetSetFixationPeriod(100); (one fixation time setting)
```

```
vetSetFixationRange(20); (area size in mm for eye fixation during 100 ms)
```

Fixation Point

```
palette = zeros(3,256);
```

```
palette(:,254) = [0,0,1]';
```

```
palette(:,255) = [0,1,0]';
```

```
palette(:,256) = [1,0,0]';
```

```
palette(:,1) = [0.5,0.5,0.5]';
```

```
crsPaletteSet(palette);
```

```
Height = crsGetScreenHeightPixels;
```

```

Width = crsGetScreenWidthPixels;
crsSetDrawPage(1);
black = [0,0,0];
crsPaletteSetPixelLevel(256, black);
crsSetPen1(256);
crsDrawRect([0,0], [7,7]);
crsSetDisplayPage(1);
vetClearDataBuffer;
pause(2);

```

Stimuli Presentation and Data Recording

```

crsDrawImage(CRS.PALETTELOAD,[0,0],'C:\Documents and Settings\Natanael Santos\Desktop\Ar-
mino DO\labirinto_toxicologia.bmp');
crsSetDisplayPage(1);
vetStartTracking;
pause(); (the program has no time limit, it waits for any key press to finish testing)
vetStopTracking;
crsClearPage(1);
vetDestroyCameraScreen;
global CRS;
vetSaveMimicScreenBitmap('C:\Documents and Settings\Natanael Santos\Desktop\Armino DO\
Mimics.bmp');
if(ischar('C:\Documents and Settings\Natanael Santos\Desktop\Armino DO\Mimics.bmp')==0)
    error('filename must be a character array (MATLAB string).');
else
    ErrorCode = vetmex(CRS.VETX_GetMimicWindowBitmap,'C:\Documents and Settings\Natanael
    Santos\Desktop\Armino DO\Mimics.bmp');
end;
Remove = false;
DATA = vetGetBufferedEyePositions(Remove);
figure(2); cla; hold on;
plot(DATA.mmPositions(:,1),'b');
plot(DATA.mmPositions(:,2),'r');
grid on;

CurrentDirectory = cd;
tempfile = [CurrentDirectory,'\myResults.csv'
vetSaveResults(tempfile, CRS.ffCommaDelimitedNumeric);

```