

Finding an Efficient Threshold for Fixation Detection in Eye Gaze Tracking

Sudarat Tangnimitchok¹✉, Nonnarit O-Iarnnithipong¹, Armando Barreto¹,
Francisco R. Ortega², and Naphtali D. Rische²

¹ Electrical and Computer Engineering Department, Florida International University,
Miami, FL, USA

{stang018, nolar002, barretoa}@fiu.edu

² School of Computer and Information Sciences, Florida International University,
Miami, FL, USA

fortega@cs.fiu.edu, ndr@acm.org

Abstract. We propose a combined analytical/statistical method to determine an efficient threshold on the dispersion of estimates of the point of gaze (POG) to indicate a user fixation. The experimental data for this study was obtained with an EyeTech TM3 eye gaze tracker (EGT). The experimental protocol to make the user fixate on pre-determined visual targets was implemented using the C language and OpenCV. Subjects first used the system in a training mode, from which an individualized dispersion threshold was obtained. Our approach was verified by applying the individualized threshold to POG data from a second run, in testing mode, with encouraging results.

Keywords: Fixation identification · Eye gaze tracking · Data analysis algorithm

1 Introduction

Eye Gaze Tracking has been used for many years in the Human-Computer Interaction field. As the eyes are another gateway to express human emotions and thoughts, one can identify a person's object of interest by determining where the person is directing his/her gaze. Trying to assign meaning to a person's gaze patterns is not an easy task. Nonetheless, one specific gesture of eye movement that we can extract from eye tracking is fixation. Our point of gaze (POG) tends to stop at a screen location when we are interested in something since our brain needs time to analyze and make sense of what we are looking at. We call this gesture 'fixation', and the method to identify the gesture 'fixation identification' or 'fixation detection'.

The key to identify the fixation gesture is to observe the dispersion of POGs (Points of Gaze), i.e., the distribution of spatial coordinates indicating the gazing point in the display. In an ideal case, the POGs during a fixation should stop at one point for a time interval (i.e., X- and Y- coordinates of the POGs should remain constant for a period of time). In practice, however, even when a person fixes his/her eyes at one point, there will be slight movements from the eye balls. This phenomenon ("microsaccades")

has an important role in our visual perception but it prevents us from obtaining the ideal POG constant coordinates we would expect during a fixation.

Several algorithms have been introduced to detect an EGT fixation despite the microsaccades, with most of them detecting a reduction in the spatial dispersion of consecutive POGs. These algorithms have been described in the EGT analysis general literature, e.g., [2–4], and in papers specifically devoted to the investigation of EGT fixation detection algorithms, e.g., [8–10]. Some other researchers have proposed modifications to the basic method [5, 11] aimed at improving the fixation detection performance. A few algorithms also use some temporal constraints in deciding whether or not a fixation has occurred.

However, many of the algorithms use thresholds for dispersion that have been developed as custom, ad hoc solutions to specific implementations and may be expressed in a variety of units (pixels, mm, etc.). This work aims at identifying a recommendable threshold for the POG dispersion that is likely to be effective, and that is not dependent on the specific units of spatial displacement used. Thus the resulting threshold obtained from the study can be applied to any system regardless of the devices and units it may use. We also propose that each EGT system user should have his/her own efficient individualized threshold determined because the specific behavior of the eye gaze during fixations may vary from person to person. So, each user should go through the process of finding his/her best individualized threshold (during a training stage) and apply it for system use afterwards (testing).

2 Methodology

2.1 EyeTech TM3 Eye Gaze Tracker (EGT)

The EyeTech TM3 is a compact and portable eye tracker from EyeTech Digital Systems, Inc. The system consists of a high definition camera, infrared sources, and its software environment. It is capable of tracking with one or both eyes in real time (providing POG estimates every 26 ms, in our experiment) and it can be used with any Windows-based communication software. In this study, we used their provided libraries (OpenCV) [1, 7] to build the visual-based interactive software in the designed experiment. The model specifications are listed in Table 1.

Table 1. Technical specification of EyeTech TM3

Methods	Video, dark pupil, infrared illumination
Spatial resolution	1 degree (approximate)
Temporal resolution	Adjustable, 15–55 samples per second
Freedom of head movements	25 X 16 X19 cm
Temporal resolution	Adjustable, 15-55 samples per second

2.2 Experimental Design

For the purpose of finding an efficient threshold that does not depend on any specific units, we created an interactive program based on the EyeTech TM3 eye gaze tracker (EGT) to record the POGs continuously throughout the experimental session.

We involved 22 participants in the implementation of our approach. After performing the calibration suggested by the eye gaze tracker manufacturer, each subject was instructed to complete 2 experimental stages: Training and Testing. The X- and Y- screen coordinates of the POG were recorded throughout both complete stages. In each stage the protocol presents 5 visual targets (pink circles), located randomly in sequence as shown in Fig. 1. Prior to the beginning of the experiment, the subject is instructed to fixate his/her gaze only on those visual targets, when they appear. To prevent unintended fixations in the intervals between target presentations, a yellow circle, completely different in appearance from the targets, is shown moving around the screen, as a distractor.

Our approach continuously stores the current X coordinate and the previous 49 X coordinates recorded in a 50-point First-In-First-Out (FIFO) buffer and uses them to calculate a standard deviation of the X coordinates every sampling instant. The same process is followed for the Y coordinates of the point of gaze. The standard deviations for both the X and Y axes will be calculated iteratively. Therefore, in our approach we have current estimates ($N = 50$) of the standard deviation of the POG coordinates (i.e., σ_x and σ_y), at every sampling instant.



Fig. 1. Interactive program used in the experiment with the EyeTech TM3system. The pink circle at the lower bottom of the screen is a fixation target.

2.3 Statistical Approach for Fixation Identification

We propose to use the statistical dispersion of the X and Y POG coordinates as the criteria to determine when a fixation has occurred. In particular, we seek to identify a constant threshold, K , for the standard deviations of the POG coordinates in X and in Y

(σ_x and σ_y) to determine the occurrence of EGT fixations, as we expect marked and simultaneous decreases in σ_x and σ_y during fixations (as shown in Fig. 2). Note that our proposed threshold K will, therefore, be unaffected by the type of units in which the EGT system reports coordinates or distances.

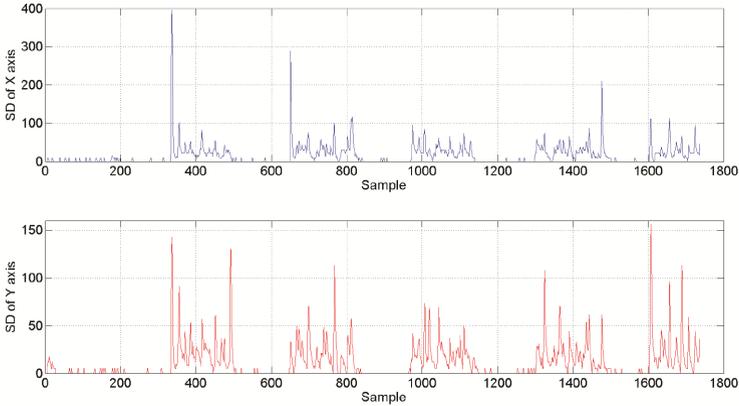


Fig. 2. Plots of the standard deviations of the POG coordinate in X and in Y

Smoothing the Standard Deviation Signals using a one-pole Filter. Due to the abrupt, short-term drops that occur in σ_x and σ_y when a fixation is not taking place (Fig. 2), the system may report brief erroneous fixation detections. To circumvent this problem we apply a one-pole filter of the type used in Gamma memories [6] to σ_x and σ_y . The filter will give the current output sample based on the input sample and the previous output sample. We can adjust the μ parameter ($0 < \mu < 1$) to set whether the output should depend more on the current input or on the previous output. As a result, the filter will smooth the signal that it processes, while retaining the envelope shape of the original signal. Its purpose is similar to that of a running average filter, except that it has faster performance and is much easier to implement, compared to the normal average filter. Figure 4 shows the signal before and after applying the filter (Fig. 3).

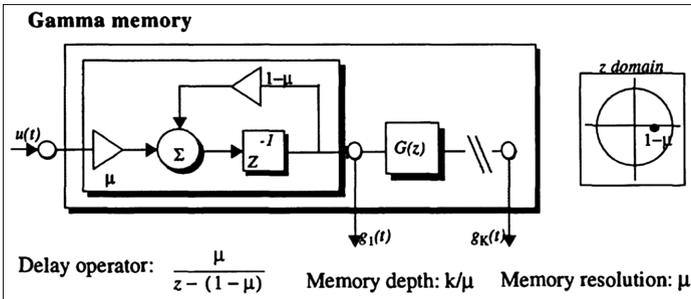


Fig. 3. One-pole filter as used in a Gamma Memory (From [6])

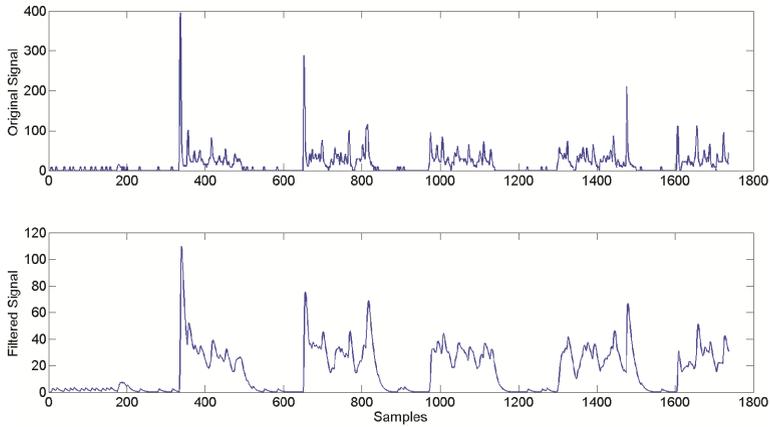


Fig. 4. Standard Deviation of X signal, (σ_x), before and after applying a one-pole filter

Sweeping Threshold and ROC Curve. To identify the most effective K value, the pre-recorded POG files from the training stage of each participant was processed by an algorithm that indicates a fixation only if $\sigma_x < K$ AND $\sigma_y < K$, where the standard deviations are calculated on the basis of the present POG and the immediately previous 49 POGs (i.e., $N = 50$). The result (Fixation OR No-Fixation) is assigned to the present temporal sample and the analysis is repeated throughout the complete POG file. Since the timing of appearance and disappearance of the 5 actual targets through the experiment is known and recorded, we are able to assess how many of the fixation indications from the system are correct (“True positives”) and how many are incorrect (“False Positives”). This process is repeated for increasing values of K (starting at 0), until every

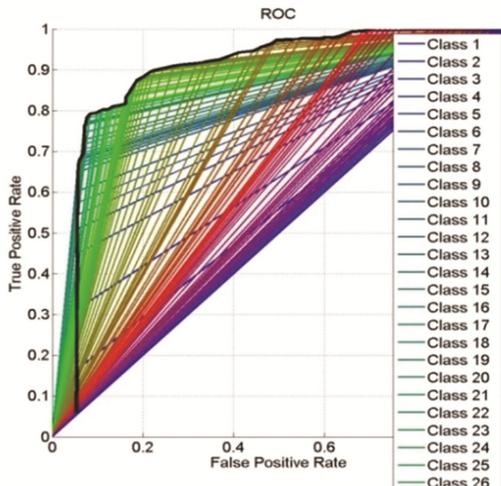


Fig. 5. Receiver Operating Characteristic (ROC) curve of 100 thresholds

POG is reported as a fixation by the system. For each threshold, K , tried the “True positive rate” and “False Positive rate” enables us to calculate and draw one point of the Receiver Operating Characteristic (ROC) curve for the fixation detection process (Fig. 5) and, from the whole curve, we are able to select the best K value (the K value that defines the closest point in the ROC curve to the coordinates “False Positive Rate” = 0 and “True Positive Rate” = 1, located on the top-left corner of the graph). This will be considered the best individualized K value for that specific participant.

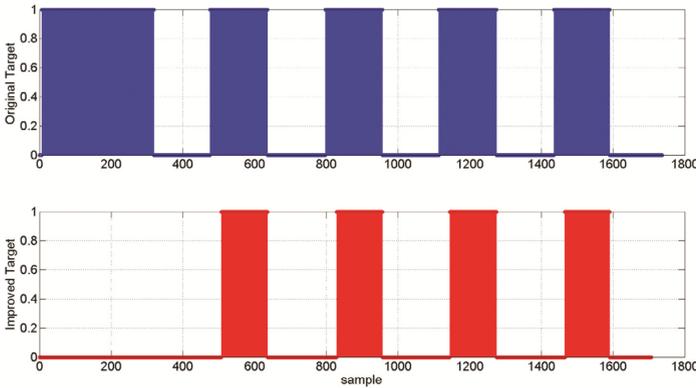


Fig. 6. Original target indications and the adjusted target indications (including only 4 targets)

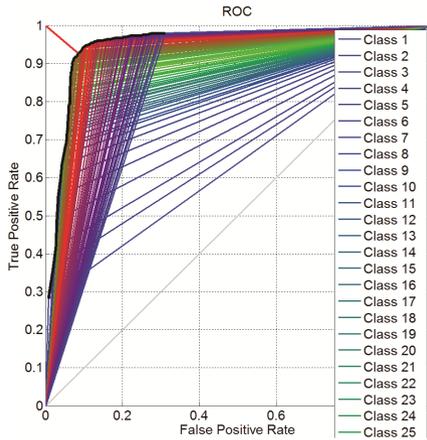


Fig. 7. Improved ROC curve after disregarding the first target presentation

Improved Result from ROC Curve. In processing the POG data from the training stage from all participants we noticed that the evolutions of σ_x and σ_y prior and during the presentation of the first target were extremely inconsistent, seemingly due to the lack of familiarity of the test subjects with the system at the beginning experiment. This confounding effect does not persist after the second target presentation. Accordingly,

we decided to perform our analysis only considering the POG evolution during and after the second target presentation (“adjusted target indications”, Fig. 6). As expected, this resulted in higher, more consistent levels of accuracy in fixation detection (e.g., Fig. 7).

3 Results

3.1 Diversity of Individualized Thresholds Found

After the individualized thresholds for all the participants were found, in the way described above, the histogram of these thresholds was constructed, as shown in Fig. 8. We observe that the thresholds found to provide the best performance for each of the participants are not all equal, and, in fact are significantly dispersed around their mean value. This provides some level of verification of our expectations and further confirms that an individualized threshold should be obtained for each EGT user through training for efficient operation of the fixation detection process.

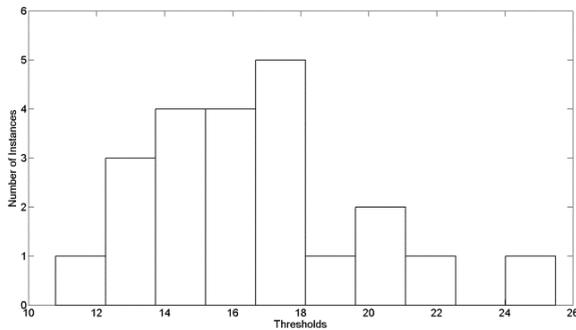


Fig. 8. Histogram of individualized best thresholds found for the participants

3.2 Testing the Individualized Thresholds

For the purpose of testing the performance of the individualized thresholds, we asked the test subjects to go through the experiment a second time, in the testing stage, and recorded the POG data (sequence of X- and Y-coordinates) for the second time. In this second stage, however, the individualized threshold found for the specific participant is applied in real-time to the σ_x and σ_y calculated continuously as the experiment takes place. Therefore, this time there will be a small red dot that appears in the display to indicate the test subject’s POGs. The dot’s color will remain red if the system detects no fixation, and turn to yellow when a fixation is detected.

The instructions to the subject in the testing stage are the same as during the training stage: Whenever the visual target appears (Target = 1), we ask the subject to fix his/her gaze on it so the algorithm is supposed to indicate a fixation (Result = 1). On the contrary, whenever the target disappears (Target = 0), we ask the subject to follow the distractor (moving yellow circle) in the display to prevent the occurrence of an unintended fixation.

Thus the result is presumed to be non-fixation ($\text{Result} = 0$). This setup allows us to also calculate the correct and incorrect results provided by the system and, therefore, evaluate the system error rate ($1 - \text{accuracy}$).

After we apply the individualized best threshold obtained from the training stage to the testing stage recorded data, we obtained the error rate results as shown in Fig. 9 (bottom trace). Please note that it takes some time after the target appears before the test subjects can actually locate it and move their gaze to the visual target. As a result of that, we set the value in the target vector in those transition intervals to zero and call the resulting target indicators ‘Improved target’ indicators (middle trace of Fig. 9). This adjustment allows the calculation of a more realistic error rate, which we obtained for all subjects, for both their training POG data and their testing POG data. These error rates are shown and compared Fig. 10. The graph in the continuous line is the error rate from the training stage, while the graph in the dashed line is the error rate from the testing stage. We can appreciate that, for the vast majority of subjects, the testing and training performances are not significantly different.

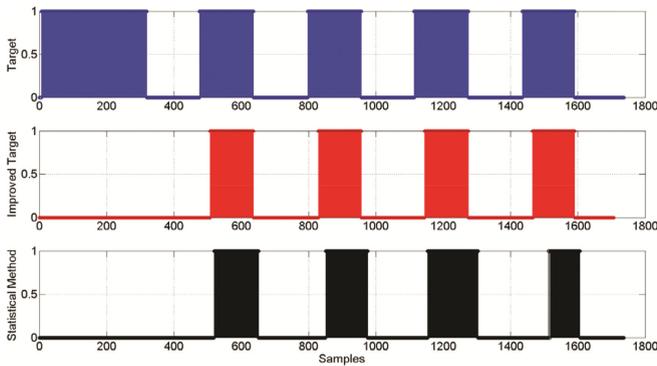


Fig. 9. Target indicators and resulting vector calculated by the system (example). On the top shows the original target vector. In the middle shows the improved target vector after trimming the first target and on the bottom shows the resulting vector from our approach.

To verify the hypothesis that there are no statistically significant differences in mean between the two error rates (from the training stage and the testing stage) of each participant, we ran a Paired T-test [12] using the R-software as shown in Fig. 11. We set the null hypothesis to be that the difference between the mean of two data sets is zero and set the level of significance at 0.05. The p-value resulting from the test was 0.098 which is greater than 0.05, so we do not reject the null hypothesis and conclude that the means are not different at 5 % level of significance. This further supports our observation on the similarity of both traces in Fig. 10.

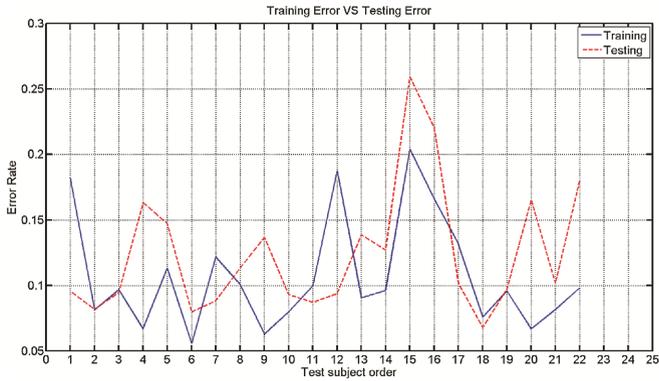


Fig. 10. Plots of error rate from the training and testing stages

```

> # Input Data
> Train = c(0.18,0.08,0.09,0.06,0.11,0.05,0.12,0.10,0.06,0.07,0.09
+ ,0.18,0.09,0.09,0.20,0.16,0.13,0.07,0.09,0.06,0.081,0.09);
> Test = c(0.09,0.08,0.09,0.16,0.14,0.08,0.09,0.11,0.13
+ ,0.09,0.08,0.09,0.14,0.12,0.25,0.22,0.10,0.068,0.10,0.16,0.10,0.18);
> # Test the equality of variance
> var.test(Train,Test);

      F test to compare two variances

data:  Train and Test
F = 0.79683, num df = 21, denom df = 21, p-value = 0.6075
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.330827 1.919228
sample estimates:
ratio of variances
 0.7968263

> # T-test of the difference in means (pair-Testing)
> t.test(Train,Test,var.equal=T,paired=T);

      Paired t-test

data:  Train and Test
t = -1.73, df = 21, p-value = 0.09831
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.041739879 0.003830788
sample estimates:
mean of the differences
 -0.01895455

```

Fig. 11. Paired T-test result of the difference in mean of the error rate between training and testing stages using R software

This means that the benefit of having found the best individualized threshold for each participant through the ROC analysis of training stage POG data is kept even in subsequent uses of the EGT system. In our experiment, for example, the system was almost as accurate in detecting fixations in the testing stage as it was during the training stage.

3.3 Adjustable Characteristics of the Algorithm Proposed

Another important aspect of the proposed algorithm is that by adjusting the μ parameter used in the one-pole filter employed for the smoothing of the standard deviation sequences the performance characteristics of the algorithm can be altered. μ has a range from 0 to 1; a small μ value will increase the detection accuracy while a large μ may cause more false fixation detections. Conversely, a small μ tends to promote a slower response, compared to a large μ . The reason behind this is that a smoother standard deviation sequence will have slower level changes and, therefore, may take additional time to drop below the K threshold to indicate a fixation. On the other hand, if the strength of the smoothing effect is lessened, the filtered standard deviation signal is more like the unfiltered version of the signal, still displaying fast transitions, but also containing spurious drops, even when no fixation is taking place.

4 Conclusion

In this work, we aimed at identifying a recommendable threshold for the POG dispersion that is likely to be effective, and that is not directly affected by the specific units of spatial displacement (pixels, mm, etc.) used in any particular device. Further, we proposed that each EGT system user may need a different threshold in the fixation detection algorithm. We showed that such individualized threshold can be obtained from the data gathered during a short training stage, by means of ROC curve analysis.

The histogram of individualized best thresholds found does show diversity for the different participants, indicated by a noticeable dispersion around the mean. Using the individualized threshold in the analysis of a subsequently recorded testing stage proved that the high performance shown in the training stage is kept. Both these observations seem to confirm that it is useful to determine an efficient threshold for the fixation algorithm for each user during the brief training stage.

Moreover, we can adjust the performance balance between accuracy and response time using the μ parameter of the smoothing filter to fit the demands of a specific fixation detection application. By adjusting this parameter, the response time of the fixation detector could be shortened, at the expense of detection accuracy. Conversely, a higher accuracy may require a μ value that might make the detector somewhat slower to respond.

Acknowledgements. This material is based in part upon work supported by the National Science Foundation under Grant Nos. I/UCRC IIP-1338922, AIR IIP-1237818, SBIR IIP-1330943, III-Large IIS-1213026, MRI CNS-1532061, OISE 1541472, MRI CNS-1532061, MRI CNS-1429345, MRI CNS-0821345, MRI CNS-1126619, CREST HRD-0833093, I/UCRC IIP-0829576, MRI CNS-0959985, RAPID CNS-1507611.

References

1. Bradski, G., Kaehler, A.: *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly, Sebastopol, CA (2008)
2. Duchowski, A.: *Eye tracking methodology: Theory and practice*, 2nd edn. Springer, Heidelberg (2009)
3. Jacob, R.J.K.: What you look at is what you get: Eye movement based interaction techniques. *Proceedings ACMCHI'90 Human Factors in Computing Systems*, pp. 11–18. ACM Press, New York (1990)
4. Jacob, R.J.K.: Eye movement-based human-computer interaction techniques: Toward noncommand interfaces. In: Hartson, H.R., Hix, D. (eds.) *Advances in human-computer interaction*, vol. 4, pp. 151–190. Ablex, Norwood, NJ (1993)
5. Kumar, M., Klingner, J., Puranik, R., Winograd, T., Paepcke, A.: Improving the accuracy of gaze input for interaction. *Proceedings of the 2008 Symposium on Eye Tracking Research and Applications*, pp. 65–68. ACM Press, New York (2008)
6. Principe, J.C., Hsu, H.H., Kuo, J.M.: Analysis of short term memories for neural networks. In: *NIPS*, pp. 1011–1018 (1993)
7. Quicklink2 Library. (n.d.). Retrieved November 5, 2015. https://gitlab.eyetechds.com/windows_public/ql2matlabwrapper/raw/2608b0c438382b32a2d063e89aa42632f806be8b/QL2MatlabWrapper/QL2MatlabWrapper.cpp
8. Salvucci, D.D., Goldberg, J.H.: Identifying fixations and saccades in eye-tracking protocols. *Proceedings of the 2000 Symposium on Eye Tracking Research and Applications*, pp. 71–78. ACM Press, New York (2000)
9. Shic, F., Chawarska, K., Scassellati, B.: The incomplete fixation measure. *Proceedings of the 2008 Symposium on Eye Tracking Research and Applications*, pp. 111–114. ACM Press, New York (2008)
10. Spakov, O., Minitas, D.: Application of clustering algorithms in eye gaze visualizations. *Inf. Technol. Control* **36**, 213–216 (2007)
11. Urruty, T., Lew, S., Ihadaddene, N., Simovici, D.A.: Detecting eye fixations by projection clustering. *ACM Trans. Multimedia Comput. Commun. Appl.* **3**, 23:1–23:20 (2007)
12. Yates, R., Goodman, D.: *Probability and stochastic processes: A friendly introduction for electrical & computer engineers*. John Wiley, New York (1999)