



**HAL**  
open science

# Eye-Tracking Data Exploration Within Interactive Genetic Algorithms

Marcel Cremene, Ovidiu Sabou, Denis Pallez, Thierry Baccino

► **To cite this version:**

Marcel Cremene, Ovidiu Sabou, Denis Pallez, Thierry Baccino. Eye-Tracking Data Exploration Within Interactive Genetic Algorithms. 2010. <hal-00480933>

**HAL Id: hal-00480933**

**<https://hal.science/hal-00480933v1>**

Preprint submitted on 5 May 2010

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

## EYE-TRACKING DATA EXPLORATION WITHIN INTERACTIVE GENETIC ALGORITHMS

MARCEL CREMENE<sup>(1)</sup>, OVIDIU SABOU<sup>(2)</sup>, DENIS PALLEZ<sup>(3)</sup>,  
AND THIERRY BACCINO<sup>(4)</sup>

**ABSTRACT.** This research is part of a project that aim is to offer a general method for building artificial faces by using interactive genetic algorithm. Preliminary experiments are concerned with color discrimination and number comparison. The objective of this research is to compute the ranking for each individual (artificial face), based on the data collected from an eye-tracking system.

### 1. INTRODUCTION AND PROBLEM STATEMENT

The idea proposed in this paper is related to a general method [2] for building artificial faces (police portraits), based on human interaction. An interactive genetic algorithm [6] creates artificial faces and displays them simultaneously on the screen. The idea is to create a ranking between these individuals and use a rank-based fitness assignment for the genetic algorithm.

An eye-tracking system records continuously the user's gaze activity (while looking at the virtual faces) in order to replace the human explicit actions like using keyboard or mouse. The eye-tracking system offers an interface that is much more faster and easier to use than a keyboard/mouse based interface and avoid the user fatigue (for a high number of iterations).

Preliminary experiments are concerned with color discrimination and number comparison. Thus, the rank is known a priori and we can use supervised learning in order to classify the ocular (eye-tracking) data.

Each subject involved in the experiments is asked to identify the lightest color (experiment 1) or the highest number (experiment 2). The eye-tracking systems measures various parameters such as: the time the user has focused on a colored square, the pupil diameter and its relative rank, the relative time focused on screen,

---

2000 *Mathematics Subject Classification.* 68T10 Pattern recognition, speech recognition.

*Key words and phrases.* Eye-tracking, Interactive Genetic Algorithm, Ranking, Classification.

the maximum variation of the pupil diameter, etc. In our experiments, there are 16 parameters used as inputs for the rank classifier.

Based on this measurements, the system assigns a (subjective) fitness to each candidate solution shown on the screen. Eight colors/numbers are presented to the screen simultaneously. Colors/numbers are ranked according to the subjective fitness. Each epoch, the fittest individual in the population is selected. Using the fittest individual, the interactive genetic algorithm generates a new population. The process ends after a pre-established number of generations.

Our objective is to find a correlation between the ocular activity and the individual (color, number) rank. The aim is to train the system to discover the order the colors are presented in the experiment, based on the ocular data. Supervised learning technique is used. Data produced by the eye-tracking system are used as training data for a linear classifier in order to detect rules enabling to associate an unknown individual to a predefined class.

## 2. CONTEXT AND MOTIVATION

**2.1. Interactive Genetic Algorithms.** Interactive Evolutionary Computation (IEC) is an optimization technique based on evolutionary computation such as genetic algorithm, genetic programming, evolution strategy, or evolutionary programming. Evolutionary computation consider several candidate solutions to a problem called the population. Thanks to an iterative progress, this population is computationally evolved by using mechanisms inspired by biological evolution such as reproduction, mutation, recombination, natural selection or survival of the fittest [5] according to the Darwin's theory.

In classical evolutionary computation, a selection operator is often a program or a mathematical expression called the fitness function that expresses the quality of a candidate solution. Interactive Evolutionary Computation is used when it is hard or impossible to formalize efficiently this function where it is therefore replaced by a human user.

A large survey can be obtained in [6] but the generally accepted first work on IEC is Dawkins [7], who studied the evolution of creatures called "biomorphs" by selecting them manually. Another very good example to better understand the interest of IEC could be "photofit" building [8]. In this case, there is no mathematical function which could specify how much a photofit is interesting; only the witness can subjectively tell whether proposed photofits are similar or not to the person he had seen before.

As mentioned before, IEC is used when a fitness function is difficult and sometimes impossible to formalize. Human-Based Genetic Algorithms (HBGA) go further by allowing evolutionary computation where a good representation of individuals is hard or impossible to find Chen04, for instance they can be used in storytelling or in development of marketing slogans. To prove the usefulness of such techniques, the authors changed the classical One-Max optimization problem

into an interactive one by interpreting the individuals (strings of bits - 0 or 1) as colors to be interactively presented and manipulated.

Characteristics of IEC are inconsistencies of individuals fitness values given by the user, slowness of the evolutionary computation due to the interactivity, and fatigue of the user due to the obligation to evaluate manually all the individuals of each generation [6]. For instance, the user is often asked to give a mark to each individual or to select the most promising individuals: it still requires active time consuming participation during the interaction. The number of individuals of a classical IEC is about 20 (the maximum that can be represented on the screen), and about the same for the number of generations.

**2.2. Combining IEC with an eye-tracking system.** The paper [2] presents a new technique, totally domain independent, called E-TEA (Eye-Tracking Evolutionary Algorithm). The idea is to minimize the user fatigue by combining an IEC and an untraditional input device: an eye-tracking system. This device allows capturing user's gaze while the user is looking on a monitor.

This is possible by using eye-tracking systems such as "Tobii" which are totally non-intrusive for users. Thus, we ensure there is no need for explicit user action (choosing and clicking the most promising individual, valuating all the solutions etc.) during the evaluation process of the IEC. He just has to watch various solutions on the screen and to tell when he has finished evaluating/looking. The E-TEA algorithm then has to determine automatically which solution is better amongst presented solutions by combining gaze parameters obtained.

**2.3. Classification methods used.** Several classification methods were used in order to analyze the data obtained from the eye-tracking system. these methods are described bellow.

*Linear (Fisher's) classifier.* This method is related to the Fisher's linear discriminant. The idea is to find the linear combination of features, which separate two or more classes of objects. This method projects high-dimensional data onto a one dimensional space and performs classification in this one-dimensional space. The projection maximizes the distance between the means of the two classes while minimizing the variance within each class. Other details about this method may be found on Wikipedia and in [10].

*Neural networks (NN) classifier,* using variable number of hidden neurons (HidN). This method relates to the well known classical pattern recognition using neural networks [11], as we have tried in our first test also.

*Naive Bayesian (NB) classifier.* A naive Bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem with strong independence assumptions (the presence of a particular feature of a class is unrelated to the presence of any other feature). Usually, parameter estimation for naive Bayes models uses the method of maximum likelihood. Other details may be found in [12].

*Gaussian (ML) classifier.* Gaussian classification is a type of statistical classification. Statistical classification is a procedure in which individual items are placed into groups based on quantitative information on one or more characteristics inherent in the items. For other details see [Ger08].

*Direct Tree (DT) classifier.* A decision tree is a predictive model; that is, a mapping from observations about an item to conclusions about its target value. In these tree structures, leaves represent classifications and branches represent conjunctions of features that lead to those classifications [14].

*T-distribution, full covariance classifier.* The (Student) t-distribution is a probability distribution that arises in the problem of estimating the mean of a normally distributed population when the sample size is small and it is a special case of the generalized hyperbolic distribution. Other details may be found in [15].

### 3. PROPOSED SYSTEM

**3.1. System architecture.** The system based on interactive genetic algorithms and eye-tracker interface is depicted in figure 1. The *Genetic algorithm* generates the individuals, the *Classifier* computes the rank for each individual, the *Individual Strategy selection* bloc uses the ranks in order to select the individuals that will be used for creating the next generations.

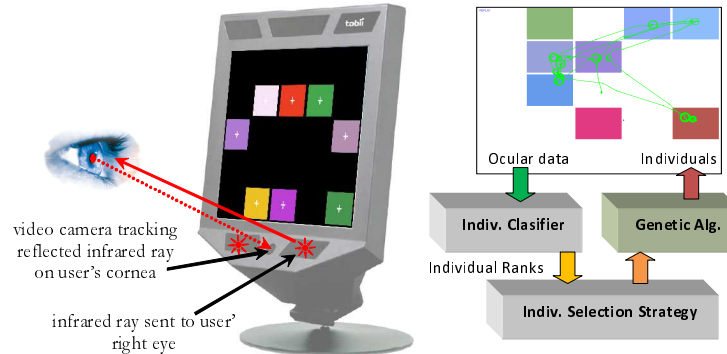


FIGURE 1. The system based on interactive genetic algorithms and eye-tracker user interface

**3.2. Ocular data.** Data obtained from the eye-tracker (Tobii 1750) each 20 milliseconds for each eye are the following: times-tamps of data in seconds and milliseconds, eye position (x and y) related the current calibration, eye position (x and y), distance between eye and camera of the eye-tracker, pupil size in millimeter, validity of eye (that is whether the eye was capture or not by the eye-tracker).

In order to simplify, we only consider the gaze position represented by center of gravity of both eyes and computed from eyes positions.

According to psychologists, a fixation last between 100 and 300 milliseconds. So fixations are computed from filtered raw data. For each fixation computed, we know the following: coordinates (x,y) of subject's gaze; duration in microsecond; colored square (or number) corresponding to the fixation. If no colored square is attached to a fixation, it is not consider as a fixation.

In raw data, the eye-tracker has given the pupil diameter and we know that it is correlated with the subject's concentration; however, we do not know how. That is why we have computed several data relating to this pupil diameter. As a fixation last at least 100 ms, a fixation is made of 5 measures at least; and we know for each of them the size of the pupil diameter.

The following data are stored for each fixation related to the size of the pupil diameter: the mean, the size at the beginning and at the end of the fixation, the value of the reference pupil that corresponds of the pupil diameter when focusing on the white cross and just before presenting the colored squares, the maximum variation of the size, the sum of variation of the size.

The experiment requires two phases. Phase 1, named 'cross fixation': a white cross is presented in the center of the screen. Fix this cross to go next screen (when correctly fixed a red rectangle will surround the cross). Phase 2 named 'evaluation': Several colored squares (or numbers) will be presented simultaneously. Detect color that seems to be lighter. Once you think you have finished, press the space bar without looking at it to go next screen (next try)." The experiment begins by the calibration of the device (the eye-tracker).

#### 4. NUMERICAL EXPERIMENTS

A set of about 54000 data vectors was recorded from the eye tracking system. Each vector corresponds to one individual. We make the hypothesis that there is a correlation between the ocular activity and the individual rank. The rest of this section presents a set of tests that were done in order to test our hypothesis.

**4.1. Test 1: MLP neural network classification.** In order to classify the ocular data in 8 classes, a first test was done using a MLP neural network (Multi-Layer Perceptron) with 16 inputs and 24 respectively 14 neurons on the hidden layers, using the back-propagation algorithm. The test results with 5000 training vectors offered very poor results. The network was able to correctly classify only about 14% of the data (not included in the training set).

**4.2. Test 2: multi-classifier toolbox.** Classifier performance depends very much on the characteristics of the analyzed data. There is no a best classifier that works on all possible problems. Knowing that, we have done also some additional data analysis using the toolbox "Matlab Classification Toolbox" from Meraka Institute.

We have analyzed our data using the following classification methods: Fisher's, Neural networks (NN), Naive Bayesian (NB), Gaussian (ML), Direct Tree (DT), and T-distribution with full covariance.

The classification errors obtained using the classification methods described previously are depicted in figure 2. A set of 10000 vectors was used. As we can observe, the best results were obtained using the Fishers linear classifier that it is also on of the fastest methods. On the second place we have the neural network classifier but the difference comparing to the Fisher's method is high. These errors were calculated for the training set of 10000 vectors.

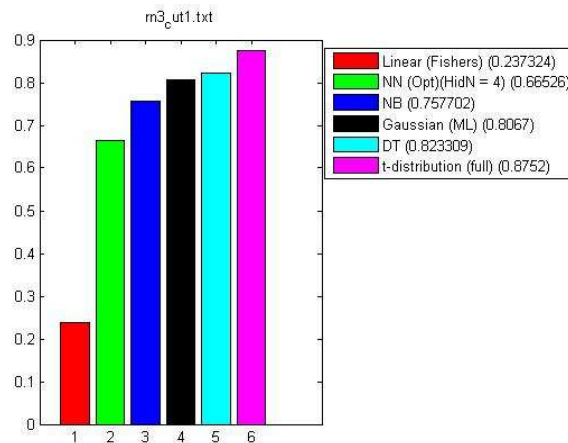


FIGURE 2. Data classification errors obtained with different classification algorithms

The toolbox includes some other algorithms such as k-Nearest-Neighbor and Gaussian Mixture Model but we were unable to use them on the full data set because they required too much resources (memory, time). However, we have tested these two algorithms on a reduced data set but their performances were poor.

**4.3. Test 3: SVM classification.** A multi-class implementation of SVM [4] was used to classify the dataset. The initial data was split in two parts, a training set of examples, comprising 5000 examples and the test set, contains the rest. The obtained accuracy was 32%.

After observing that the output of the classifier is not just a usual pattern, but a rank that forms an order relation with the rest of the classes, the accuracy analysis was further developed by taking into consideration the distance between the expected output and the one obtained from the SVM. An output result was considered accurate if it had an error less or equal than one. For example, if the

expected output was 4 and the actual output was 3,4 or 5, the output was considered correct. For ranks close to the boundary (1, 2, 7, 8) the new accuracy was 81%. The middle ranks were classified with 55% accuracy. A possible explanation is that individuals of strong interest or no interest stand out from the set.

A binary version of the dataset, with the first three ranks in one class and the rest in the other class was also tested with SVM, but the results were actually worse than the multi-class version. A possible explanation is that the SVM implementation used the one-versus-one strategy for multi-class classification which relied on multiple small machines with roughly equal number of training examples. The noise in the binary case contributed against the edge class, which had fewer training examples.

## 5. CONCLUSION AND FURTHER WORK

The research presented in this paper had as objective to automatically evaluate the user feedback (rank) for interactive genetic algorithms by analyzing the user's ocular activity (data collected from an eye-tracking system). The idea was to classify the individuals in 8 classes.

The fact that Fisher discriminant method gives about 75% correct recognition rate while the other methods give results near 30% correct recognition rate may be explained by the fact that data is very noisy and the ocular behavior is not very "stable". The ocular behavior might be influenced by some other factors that were more important than expected (peripheral view, memory, other brain activities, etc.). The experiments have shown that the classification error appear between neighbor classes (ex. classes 3,4, and 5 are easily confounded).

The fact that the color-based experiment and the number-based experiment give similar results might indicate the fact that the experiment setup should be also reconsidered. Another possible solution for increasing the precision will be to add new input parameters, for instance from additional bio-feedback sensors.

The results that have been obtained so far are not accurate, but they indicate that individuals of interest could be separated from the rest by using ocular data. A possible improvement to the current approach is to reduce the dimension of the data, in order to reduce the noise, and classify the new data with SVM.

## ACKNOWLEDGMENTS

This work was supported by the EcoNet project code 18826YM and the national project PNCIDI II code 1062.

## REFERENCES

- [1] Takagi, H., (2001). *Interactive Evolutionary Computation: Fusion of the Capacities of EC Optimization and Human Evaluation*. Proc. of the IEEE Vol. 89/9.

- [2] Pallez, D., P. Collard, et al. (2007). *Eye-Tracking Evolutionary Algorithm to minimize user fatigue in IEC applied to Interactive One-Max problem*. GECCO (Genetic and Evolutionary Computation Conference), University College, London.
- [3] G.J. McLachlan, *Discriminant Analysis and Statistical Pattern Recognition*. Wiley-Interscience; New Ed edition, August 4, 2004
- [4] Chih-Chung Chang and Chih-Jen Lin, *LIBSVM : a library for support vector machines*, 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [5] Wikipedia. (2007). *Evolutionary computation*. from [http://en.wikipedia.org/wiki/Evolutionary\\_computation](http://en.wikipedia.org/wiki/Evolutionary_computation).
- [6] Takagi, H. (2001). *Interactive Evolutionary Computation: Fusion of the Capacities of EC Optimization and Human Evaluation*. IEEE.
- [7] Dawkins, R. (1986). The Blind Watchmaker.
- [8] Takagi, H. and K. Kishi (1999). *On-line Knowledge Embedding for an Interactive EC-based Montage System*. Third International Conference on Knowledge-Based Intelligent Information Engineering Systems, Adelaide, Australia.
- [9] Cheng, C. D. and A. Kosorukoff (2004). *Interactive One-Max Problem Allows to Compare the Performance of Interactive and Human-Based Genetic Algorithms*. Genetic and Evolutionary Computation (GECCO), Seattle, Washington, USA.
- [10] G.J. McLachlan. (2004) *Discriminant Analysis and Statistical Pattern Recognition*. Wiley-Interscience; New Ed edition, August 4.
- [11] Bishop, C.M. (1995) *Neural Networks for Pattern Recognition*, Oxford: Oxford University Press. ISBN 0-19-853849-9.
- [12] Harry Zhang, (2004). *The Optimality of Naive Bayes*, AAAI Press, <http://www.cs.unb.ca/profs/hzhang/publications/FLAIRS04ZhangH.pdf>.
- [13] Robin Girard, (2008). *High dimensional gaussian classification*, <http://www.citebase.org/abstract?id=oai:arXiv.org:0806.0729>.
- [14] J. R. Quinlan, (1986). *Induction of Decision Trees*, <ftp://publications.ai.mit.edu/ai-publications/pdf/AIM-930.pdf>.
- [15] Wang, H.X., Zhang, Q.B., Luo, B., Wei, S., (2004). *Robust mixture modelling using multivariate t-distribution with missing information*, PRL(25), No. 6, 19, pp. 701-710, April.

<sup>(1)</sup> TECHNICAL UNIVERSITY OF CLUJ-NAPOCA, STR. BARITIU, NO. 26, PHONE: 0264-401813  
E-mail address: [cremene@com.utcluj.ro](mailto:cremene@com.utcluj.ro)

<sup>(2)</sup> BABES BOLYAI UNIVERSITY OF CLUJ-NAPOCA  
E-mail address: [ovidiu@sabou.org](mailto:ovidiu@sabou.org)

<sup>(3)</sup> UNIVERSITY OF NICE SOPHIA-ANTIPOLIS  
E-mail address: [denis.pallez@unice.fr](mailto:denis.pallez@unice.fr)

<sup>(4)</sup> UNIVERSITY OF NICE SOPHIA-ANTIPOLIS  
E-mail address: [thierry.baccino@unice.fr](mailto:thierry.baccino@unice.fr)