Patter Recognition Applied to Mouse Pointer Controlled by Ocular Movements

JOB RAMÓN DE LA O CHÁVEZ, CARLOS AVILÉS CRUZ
Signal Processing and Pattern Recognition
Universidad Autónoma Metropolitana – Unidad Azcapotzalco
San Pablo #180, Col. Reynosa Tamaulipas Delegación Azcapotzalco
México, Distrito Federal
yojobytu@hotmail.com

Abstract: - Researchers on *Brain-Computer Interface* (*BCI*) have tried to identify the origin of body movements in humans with limited success. This work looks at the problem using an ocular movement tracker based on ocular *artifacts* in *ElectroEncephalo-Graph* (EEG) readings, also called *Electro-Oculo-Gram* (EOG). The movements are reflected into the EEG signals, which are passed through a multiple classifier, composed of two statistical classic methods (*KNN and Bayesian-Gauss*) and a *Neural Network*. The voltage levels of EEG readings and their polarity provide the necessary information to track the focus of attention of the user in a computer screen. All of these artifacts have characteristic curves which can be classified. Focusing on the eye movements, we have developed an eye tracker to recover the point of attention of the user on a computer screen.

Key-Words: BCI, EOG, Artifacts, KNN, Bayesian, Neural Network, Eye Movement Classification.

1 Introduction

The usefulness and efficiency of computer interfaces like keyboard and mouse have stood the advance of time with little or no change whatsoever [1].

Many solutions have been proposed to improve the efficiency in communication between humans and computers. Those solutions offer different paths such as invasive methods, virtual environments [2] and *Brain-Computer Interfaces* [3][4][5][6], using Electro-Cortigram or Electro-Encephalogram (EEG) signals.

Other types of signals that can be used are the Electro-Oculograms (EOG), signals emitted by eye movement [7]. Those signals can be seen in EEG recordings. In other works, like the SIAMO project, EOG have been used to track the focus of attention of the user, using a face electrode montage to acquire signals [8].

Another way to track the focus of attention of the user is using video to follow eye movements and mapping such movement into a screen or image using video processing techniques [9].

The approach of this work involves EEG readings to extract artifacts of ocular movements (EOG). An artifact in EEG is that part of a signal that doesn't belong to neural activity, but a signal that is induced by external sources like corporal changes such as hand movements, leg movements, muscular tension or eye movements [7].

All of these artifacts have characteristic curves which can be classified. Focusing on the eye movements, we have developed an eye tracker to recover the point of attention of the user on a computer screen.

2 Experimental Model

In figure 1 we show a diagram that was from the design of the experiment, from infrastructure required (biomedical equipment) to the applications for training, passing by the design of the training software and the training protocols for the individuals.

An important thing about the successful of this research was the appropriate design of the acquisition protocol for the correct extraction of the desired features.

To achieve this point we design different protocols of acquisition, in first instance based on the comfort and the natural behavior of the user, and in second instance on the best signal acquisition in quality terms.

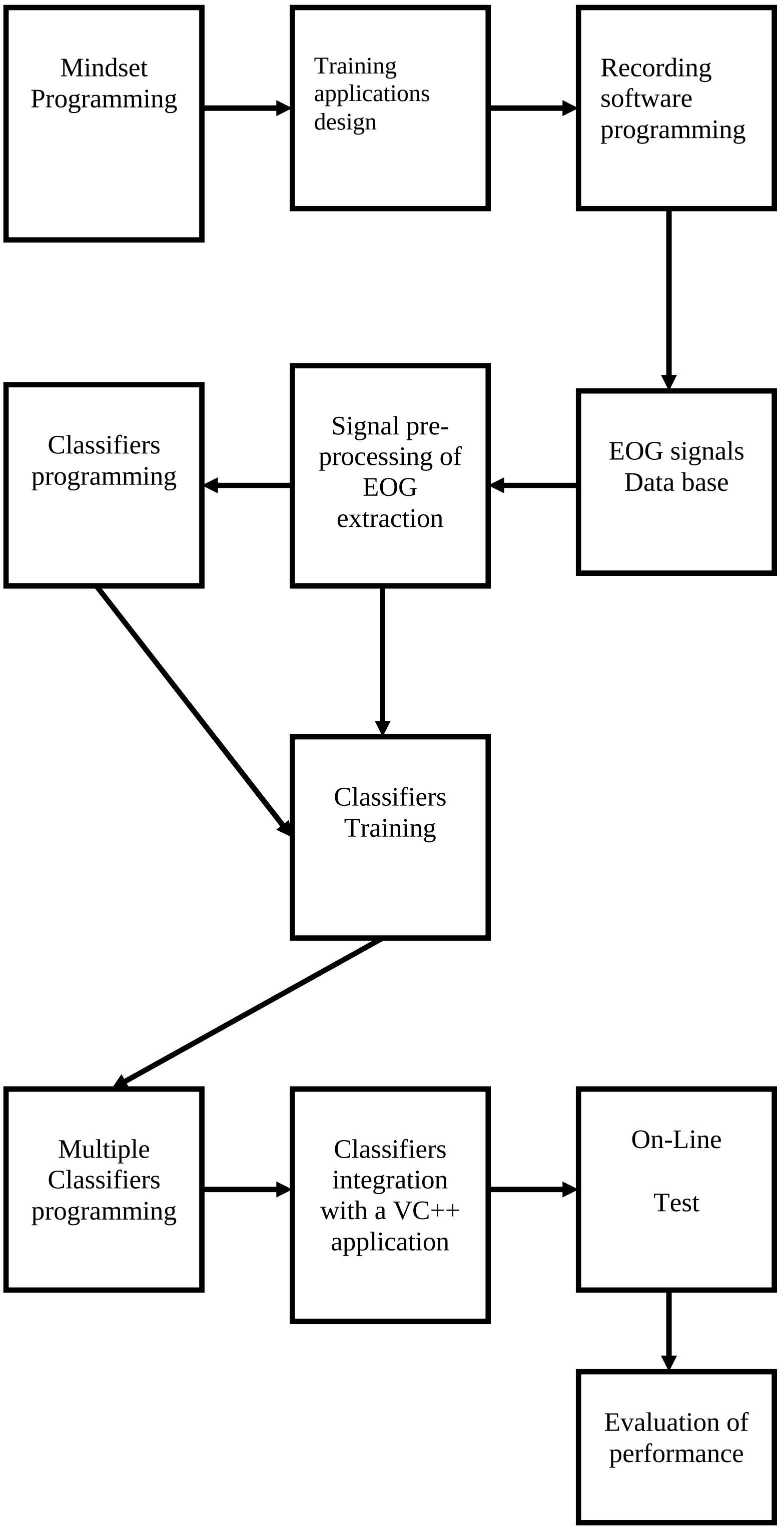

Figure 1: Experimental Model used in this research.

3 Feature Extraction

The features used in this work are artifacts induced by the ocular movement. These artifacts are caused by a potential emitted by the dipole effect between the charge in the cornea and the retina, the former being positive with respect to the latter.

This difference in potential is about 100 mV [7], and when the eye moves, a positive potential and a negative potential corresponding to the direction of the movement. The readings corresponding to the eye movement are shown in Figure 2.

Eye movements can be detected by the EEG 10-20 international system [10]. However, this acquisition lacks a good reference for vertical movements, so the potentials caused by the up and down movements show the same waveform. Another way to get EOG signals is adhering electrodes to the face; where 2 electrodes are placed about 1 cm away from the outer canthus of each eye for horizontal movements, and 2 electrodes are placed immediately above and below the eyes [7], and another one in the center of the forehead for reference. This face montage can be seen in figure 3.

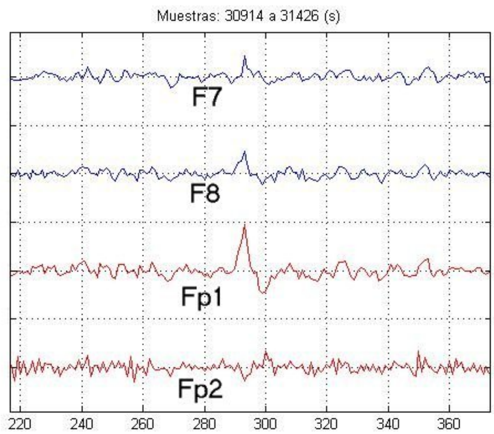

Figure 2: Sample of one eye movement recorded by the EEG system, electrodes Fp1, Fp2, F7 and F8. The movement in this figure belongs to an upward movement.

3.1 Data Acquisition

The data acquisition system used in this work is an amplifier Mindset MS-1000 from Nolan Computer System's with 16 channels in a mono-polar configuration with an ear-link reference. The 10-20 international system is used for the electrode distribution, which is provided by an Electro-Cap and a second montage with a cup electrode set (see Figure 5) placed on the face as shown in Figure 3. Additionally, we use Electro-gel to obtain a clean acquisition. The EEG acquisition system is shown in Figure 4.

For the training stage we have implemented the two different montages mentioned above, the 10-20 electrode montage and the face electrode montage. Because of the lack of reference of the 10-20 system to get good signals for vertical movements, it is not possible to have good separability, and that was the reason no to include it in this paper. The second montage, using the cup electrodes on the face (face montage) provides the necessary information to get four directions clearly and it is the montage used to obtain the results reported in this paper.

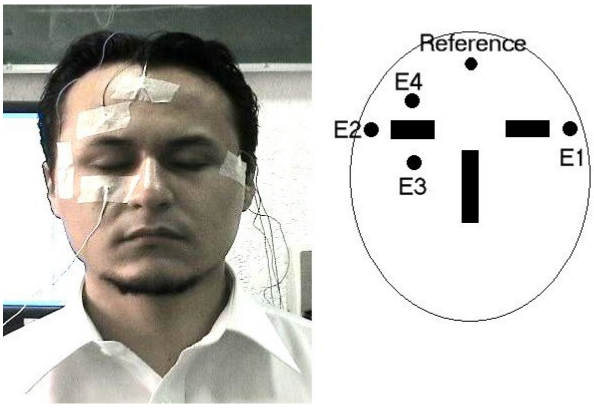

Figure 3: Face electrode montage.

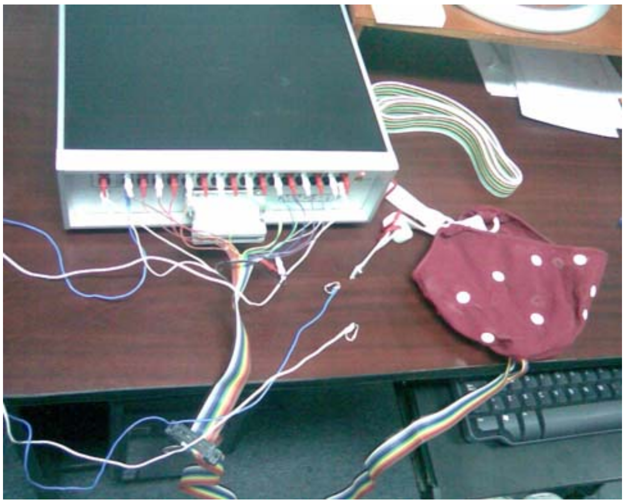

Figure 4: EEG acquisition system.

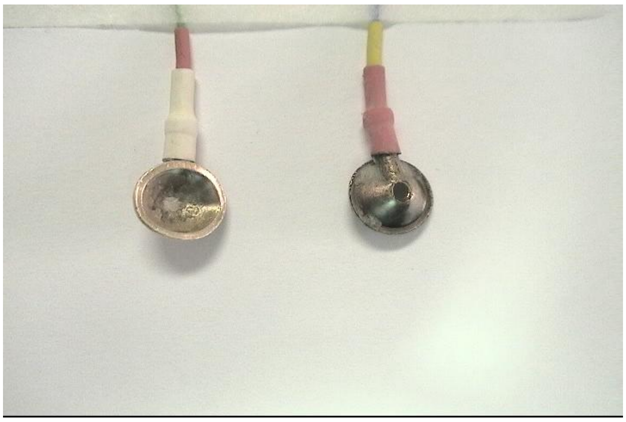

Figure 5: Cup electrode.

3.1.1 Methodology

In order to get the EOG readings, we have designed an experimental protocol for the training recordings. This protocol is implemented on a software application and consists of a session of 400 cycles, each 1.7 seconds long. When the recording starts, the first second presents to the user a blank screen where the subject has the restriction to keep the eyes still. After that, a circle appears on the screen for 500 milliseconds and then the circle moves to the right and stays there for another 500 milliseconds and then the screen goes blank again for 700 milliseconds. The above sequence defines one cycle.

There are 100 cycles for each one of the four directions: right, up, left and down; following that order. So the screen shows a circle in the middle of the screen for 500 milliseconds, then the circle changes to the right, stays there for another 500 milliseconds and afterwards disappears for 700 milliseconds, and this cycle repeats 100 times, from center to right. After 100 cycles have been executed, the circle starts again in the center to be displaced up for another 100 times. When this ends, the circle again starts in the middle to be displaced to left, and finally down.

The subject is instructed to follow the circle that appears on the screen and to keep his attention focused on the circle until it disappears. The recording starts when the circle appears in the middle of the screen, keeps recording during the transition and stops when the circle (in it's new position) disappears. The above is to get just one eye movement, when the eye goes from the center to the right, up, left or down. The system doesn't record the signal when the eye returns to the center, waiting for the next movement.

The goal is to get approximately 100 movements for each direction in a single file, in a consecutive way; that means, to have a file with 100 movements to the right, later on 100 movements up, left and down respectively. The sampling rate for the recordings was set at 64 samples per second.

We have used five volunteers for these sessions, four healthy males between 24 and 28 years old and a 21 years old healthy female.

In the record file we have four columns corresponding to four channels and a 5^{th} column to mark the original direction of the movement in that cycle.

We use this file to train the classifiers, which were created in Matlab and exported to Visual C++. This training produces a ".mat" file, which contains all the statistical parameters used by the classifiers.

3.1.2 Artifact Extraction

Feature extraction consists in extracting the ocular artifacts which have a particular amplitude and waveform. The training file obtained, shows graphically in Figure 6 the waveforms for the face electrode configuration, and in Figure 7 for the Electro-Cap 10-20 system configuration.

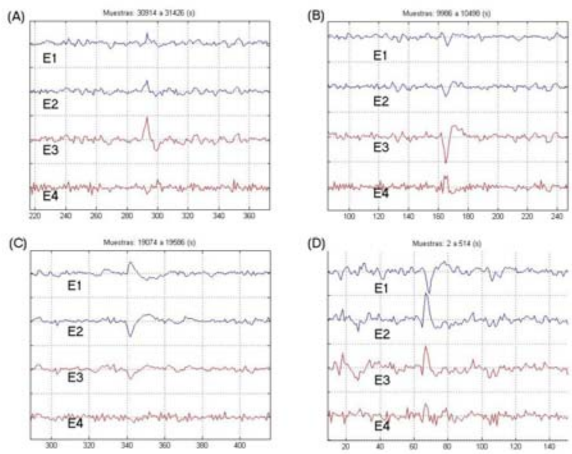

Figure 6: Signal readings for the face electrode configuration. (A) artifact for an up movement, (B) artifact for a down movement, (C) artifact for a left movement and (D) artifact for a right movement.

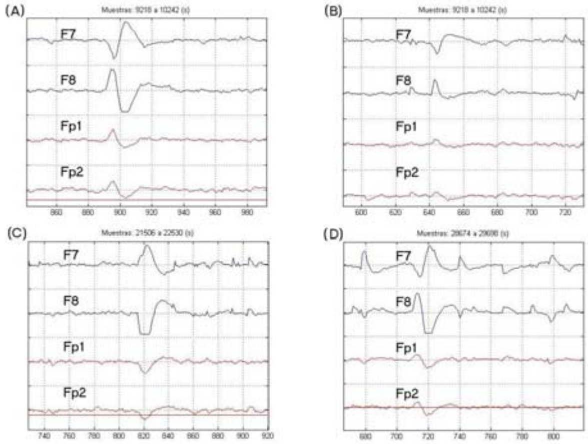

Figure 7: Signal readings for the 10-20 system configuration. (A) artifact for an up movement, (B) artifact for a down movement, (C) artifact for a left movement and (D) artifact for a right movement.

The above signals have been pre-processed with a band-pass filter from 0.1 to 32 hertz to eliminate the 60 hertz frequency, just for the case of the first electrode configuration (with the Electro-Cap). For the second electrode configuration it is not necessary this filtering, because the 60 hertz frequency is not induced, because of the lower number of electrodes connected to the system.

We can appreciate that signals can be classified in terms of their polarity and voltage level. To extract these features we construct a shared library from Matlab which analyzes the training file taking the maximum and minimum amplitude voltages of the 4 channels (Fp1, Fp2, F7 and F8 for the Electro-Cap montage, and F1, F2, F3 and F4 for the face montage) with the following criteria:

First we look for a high level voltage with respect to the average, and we take the maximum absolute value over a 0.39 seconds segment of the F1 channel (if it is a horizontal movement) or F3 (if it is a vertical movement), and we take the voltage values in the same time position for all channels

With those parameters we can graphically see its separability in four classes, shown in Figure 8, corresponding only to the face electrode montage, because the signals on the Electro-Cap montage don't have good separability.

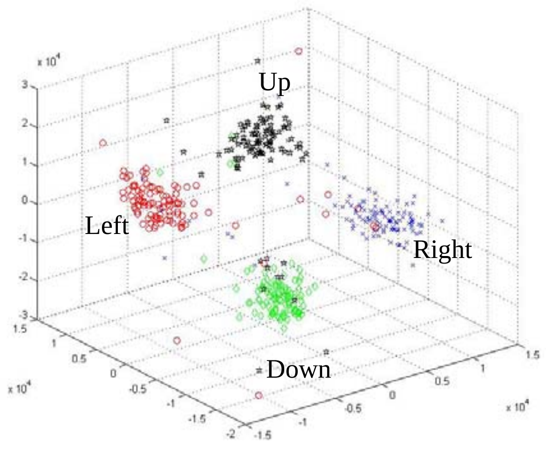

Figure 8: Plot of samples of channels 1 vs 2 vs 3 after segmentation with the face electrode montage.

We have used five classes to describe the eye movement as follows:

Class 1 Movement to the right
Class 2 Movement upwards
Class 3 Movement to the left
Class 4 Movement downwards
Class 5 No Movement: Given by the voltage below the average (in the center of the graph), or far away from the mean value.

These classes represent the training templates for the classifiers, and we can get their mean and standard deviation values.

4 Multiple Classifier

To get a more robust classification we used a multiple classifier for the eye movements. Three different classifiers were used in two configurations. The classifiers are:

a) KNN
b) Bayesian - Gauss
c) Artificial Neural Network

And the configurations are:

- Majority Vote
- Unanimity

We design a model to implement the classifiers to the same evaluation system using two different rules to give a valid result. We can see the general model of the multiple classifier in figure 9.

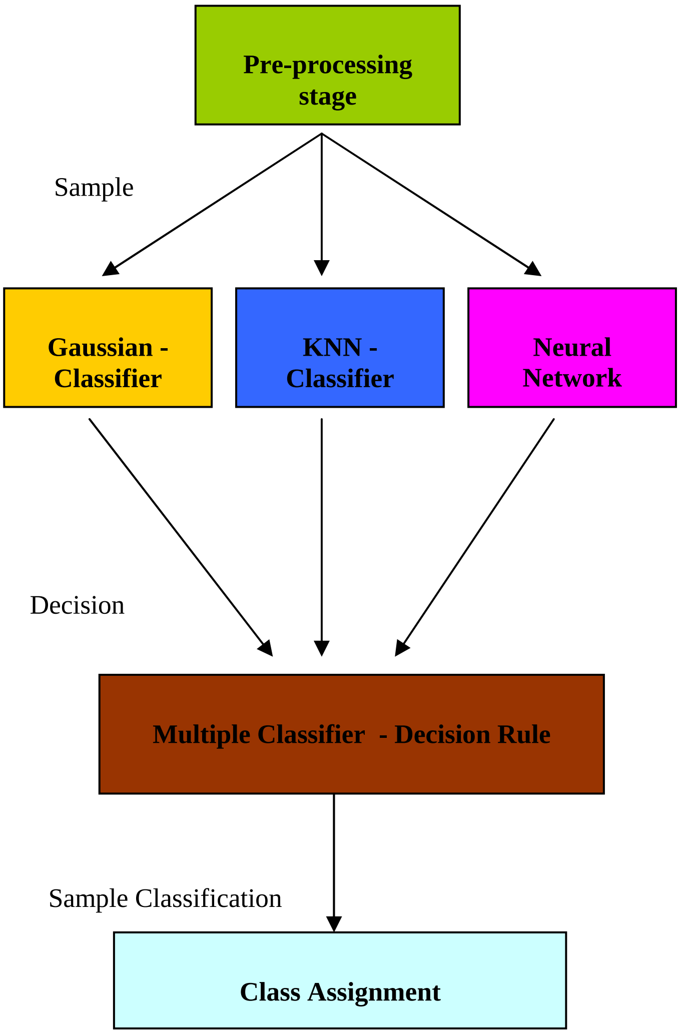

Figure 9: Design of the multiple classifier with freedom to change the rule of decision.

In Figure 10, we can see the architecture used in order to get the fusion decision of the three classifiers. This configuration is called “majority vote”. The classifier’s result will be a number that represents the ownership class, or another special value representing non ownership to any class.

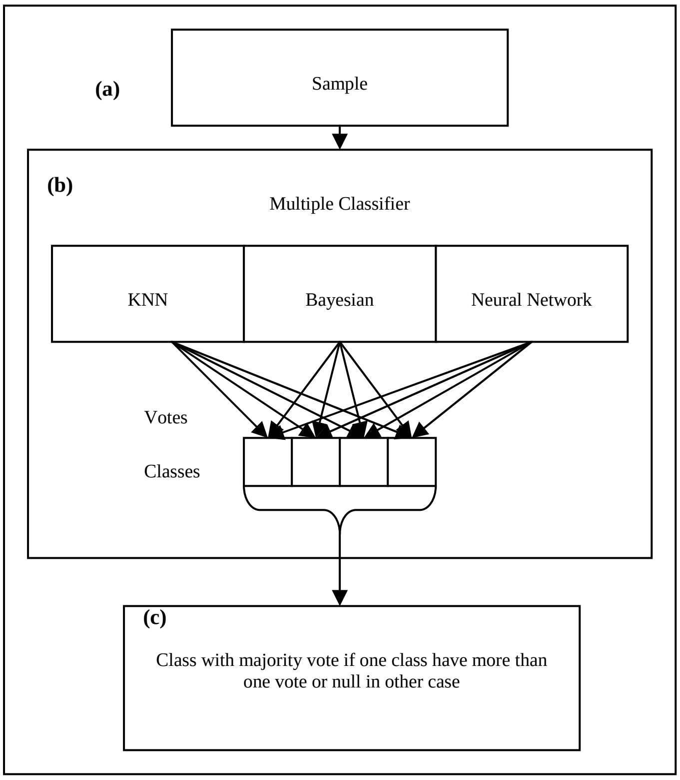

Figure 10: Majority Vote configuration diagram. (a) The sample to be classified, in this case is a vector with signal values corresponding to each channel. (b) The multiple classifier that internally contains the three classifiers and a repository where each classifier places its vote. (c) The classifier's result is a number representative of the ownership class.

The other configuration of the multiple classifier is the "Unanimity" mode (Figure 11). This configuration gives a valid result only if the three classifiers show the same class; if not, the answer will be a special value of no ownership to any class.

In both cases, the classifiers were implemented to discriminate samples by distance (KNN and Bayessian-Gauss). The statistical classifiers always give a answer in the context of the known classes with certain amount of certainty (probability), but when the samples evaluated are so far from the clusters of classes, the classifier even give an answer but with a very low probability or certainty. Is for this reason that we have added a distance discriminator.

For the case of the Neural Network Classifier the distance discriminator equivalent went added like another class.

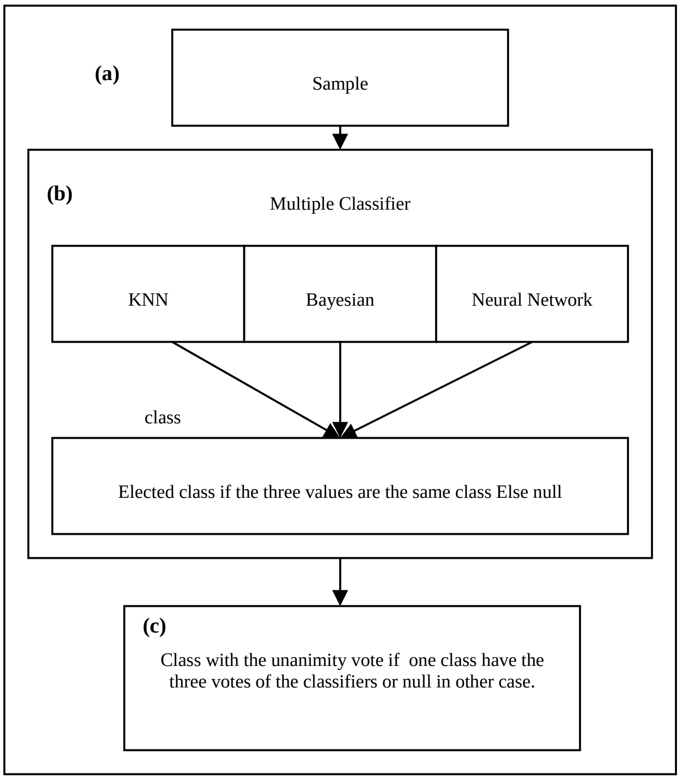

Figure 11:"Unanimity" configuration diagram. (a) The sample to be classified, composed by a vector of 4 numbers representing the values of the signal in each channel at the same time. (b) The multiple classifier that internally contains the three classifiers and an algorithm which gives a valid output if the three classifiers give the same answer. (c) The ownership class or the special value of no ownership to any class.

The Artificial Neural Network Classifier was configured with 4 inputs to the input layer, 4 neurons to the hide layer and 5 neurons to the output layer. The training method for this classifier was back-propagation using Batch Gradient Descent with Momentum implemented in Matlab with the parameter *traingdm*.

In addition we have implemented a threshold discriminator to the KNN classifier to give more stability to the results, the threshold was set up at 75% of certainty, if the classifiers don't have the 75% of certainty, sample are discriminated.

4.1 Tests

At this point we show how the sessions have been recorded. We construct an application in Visual C++ which shows a blank screen for five seconds when the training starts, followed by 400 cycles, while a circle appears and changes its position every cycle only in four directions. Figure 12 show this screen.

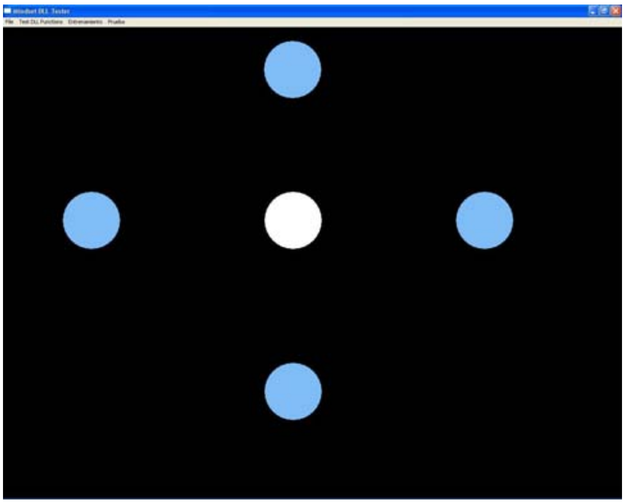

Figure 12: Training screen. The circle appears in the center for 500 milliseconds and changes its position to one of four directions, first right, afterwards up and left and down successively, and then stays there for 500 milliseconds and later disappears for 700 milliseconds, to start again in the middle of the screen.

During the 400 seconds of the duration of the session, a file is recorded with the 4 channels placed in the face as shown in Figure 6, and a 5th channel is generated by the application program and appended to the file.

For the Electro-Cap montage we use Fp1, Fp2, F7 and F8 channels of the 10-20 international system.

This file is used to train the classifiers off-line and later on, we use Matlab to reconstruct the movements and estimate the error for each classifier, working alone and in the multiple classifier configuration with the two different montages. The file format used is showed in figure 13.

Channel

1	2	3	4	5
x	y	z	w	Direction
x	y	z	w	Direction
x	y	z	w	Direction
x	y	z	w	Direction
x	y	z	w	Direction
x	y	z	w	Direction

Figure 13: This figure shows the file format used for recordings.

4.2 Results

For the classifiers tested separately, after 3 sessions of the above-mentioned recording protocol we can reconstruct the movements with the accuracy showed in Table 1, using the cross-validation method at 50%. The first classifier, KNN, was tested with different k values.

In Table 2 we show the results for the same test, but adding a discriminator threshold of distance.

In the Table 3, we can see the results for the Bayessian-Gauss Classifier, and in Table 4 we have the results for the Neural Network Classifier.

k value	Right	Up	Left	Down	Total
3	86%	96%	100%	100%	95.50%
7	92%	96%	100%	94%	95.50%
15	92%	96%	100%	92%	95%

Table 1: Efficiency of KNN classifier using different values for the k parameter.

K value	Right	Up	Left	Down	Total
3	70%	86%	74%	84%	78.50%
7	80%	94%	92%	88%	88.50%
15	92%	94%	94%	88%	89.5%

Table 2: Efficiency of KNN using different values for k and a threshold discriminator.

Right	Up	Left	Down	Total
88%	96%	100%	96%	95.00%

Table 3: Results for the Gaussian Classifier

Right	Up	Left	Down	Total
76%	96%	96%	92%	90.00%

Table 4: Neural Network classifier results

In the Tables 5 and 6 we can see the results obtained from the "Unanimity" and "Majority Vote" configuration classifiers respectively.

Right	Up	Left	Down	Total
72%	86%	98%	92%	87.00%

Table 5: Results for the Unanimity configuration using the three classifiers.

Right	Up	Left	Down	Total
92%	96%	100%	92%	95.00%

Table 6: Results for the Majority Vote configuration using the three classifiers.

Additionally to these results, we have made a simulated reconstruction of the ocular movements from the same readings used to get the results above. The graphs resulting of the reconstructions are showed in the Figures 11 to 16, corresponding to the classifiers KNN (k=3), KNN (k=15), Bayesian-Gauss, Neural Network, "Unanimity" and "Majority Vote" respectively.

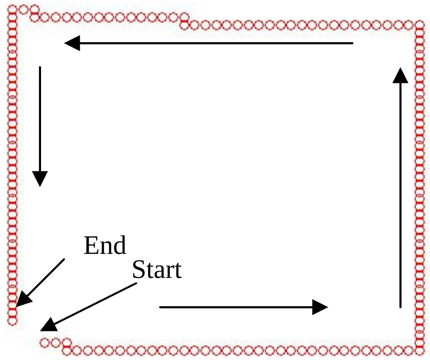

Figure 14: Reconstruction using KNN with k = 3 and with the threshold discriminator at 75%.

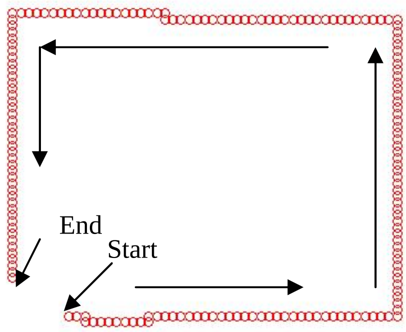

Figure 15: Reconstruction using KNN with k = 15 and the threshold discriminator at 75%.

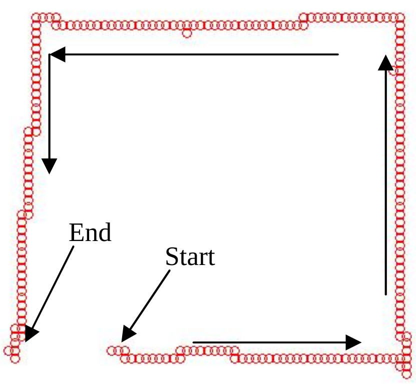

Figure 16: Reconstruction using Bayesian-Gauss classifier

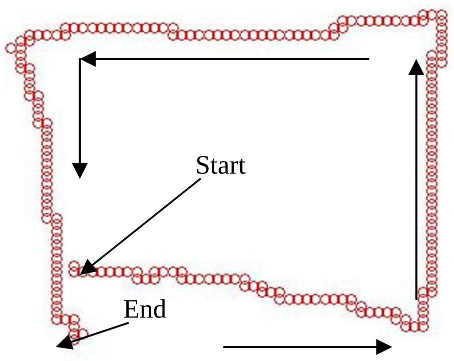

Figure 17: Reconstruction using the Neural Network classifier.

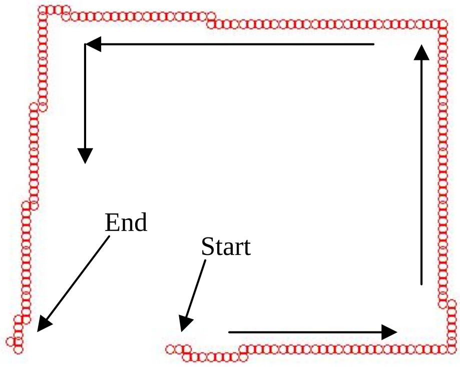

Figure 18: Reconstruction using the “Unanimity” configuration.

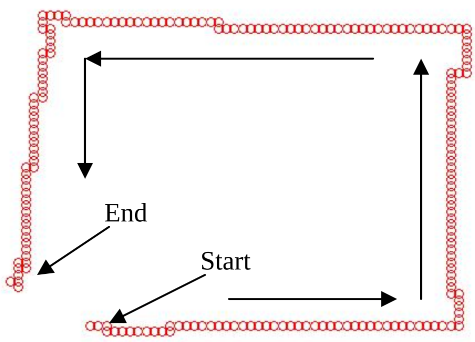

Figure 19: Reconstruction using the “Majority Vote” configuration.

The number of error’s showed in the graphs are reduced because of the threshold discriminator.

Making a review of the results showed above, we can see good results from the “Unanimity” configuration, but the best performance was obtained with the “Majority Vote” configuration, reaching efficiency up to 95%.

From the results we can observe that the classifiers tested individually offer better performance but show a decrement in their robustness. In the other hand, we can see lower performance when we use a threshold discriminator, but we can see a clean reconstruction with fewer movements; so the best configuration is the one that provides efficiency or robustness and this can be decided in terms of the specific application of system.

5 Conclusion

A new methodology was proposed and tested for ocular movements classification, using two multiple classifiers composed by Gaussian classifier, KNN classifier and Artificial Neural Network based classifier.

Using the classifiers mentioned above, we can reach 95.5% of accuracy in ocular movement detection.

Unanimity vote configuration of the multiple classifier deliver a lower efficiency than Majority vote configuration, but its more easy to have control position of the cursor at the screen.

Other configuration, deliver more efficiency but is a little harder to control it, so this is one question to be resolved by the application and the final user of this kind of interfaces.

References:

[1] Ken Hinkley, Randy Pausch, Dennis Proffitt and Neal F. Kassell. Two-Handed Virtual Manipulation. *ACM Transactions on Computer-Human Interaction*, Vol.5, No.3, 1998, pp. 260-302.

[2] Jessica D. Bayliss, Danna H. Ballard. A Virtual Reality Testbed for Brain – Computer Interface Research. *IEEE Transactions on Rehabilitation Engineering*, Vol. 8, No. 2, 2000, pp 188-190.

[3] Damien Coyle, Girijesh Prasad, Thomas Martin McGinnity. A Time-Series Prediction Approach for Feature Extraction in a Brain-Computer Interface. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, Vol. 13, No. 4, 2005, pp 461-467.

[4] Jonathan R., Wolpaw and Dennis J. McFarland. Control of a two-dimensional movement signal by a noninvasive brain-computer interface in

humans. *Proceedings of the National Academy of Sciences*, Vol 101, No. 51 2004, pp 17849-17854.

[5] Jonathan R. Wolpaw, Niels Birbaumer, William J. Heetderks, Dennis J. McFarland, P. Hunter Peckham, Gerwin Schalk, Emanuel Donchin, Louis A. Wuatrano, Charles J. Robinson, Theresa M. Vaughan. Brain-Computer Interface Technology: A review of the First International Meeting. *IEEE Transactions on Rehabilitation Engineering*, Vol. 8, No. 2, 2000, pp 164-173.

[6] Chihi Hung, Po-lei Lee, Yu-te Wu, Li-Fen Chen, Tzu-Chen Yeh, Jen-Chuen Hsieh. Recognition of Motor Imagery Electroencephalography Using Independent Component Analysis and Machine Classifiers. *Annals of Biomedical Engineering*, Vol. 33, No. 8, 2005, pp 1053-1070.

[7] EEG Technology. R. Cooper, J. W. Osselton, J. C. Shaw. *EEG Technology,* Butterworths, 1974.

[8] Mazo M., Garcia J. C., Rodriguez F. J., Urena J., Lazaro J. L., Espinoza F. Experiences in Assisted Mobility: *The SIAMO Project. Proceedings of the 2002 International Conference on Control Aplications*, Vol. 2, 2002, pp 766-771.

[9] Constantin A. Rothkopf, Jeff B. Pelz. Head Movement Estimation for Wearable Eye Tracker. *ACM Proceedings of the 2004 symposium on Eye Tracker research & Applications*, *San Antonio* , *Texas*, pp 123-130.

[10] H. Jasper. The ten-twenty electrode system of the International Federation, *Electroencephalography Clinical Neurophysiology*, Vol. 10, 1958, pp 371-375