Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

EEG – Eye Tracking: A Wavelet Packets Approach

Semester Thesis

Wayne Zeng

zengw@ethz.ch

Distributed Computing Group
Computer Engineering and Networks Laboratory
ETH Zürich

Supervisors:
Ard Kastrati, Anh Duong Vo, Martyna Plomecka
Prof. Dr. Roger Wattenhofer
Prof. Dr. Peter Bühlmann

July 18, 2023

Acknowledgements

I would like to thank Ard, Duong and Martyna for their generous help and unparalleled level of support over the span of this project. Their time and patience over weekly thorough and detailed meetings is greatly appreciated.

Abstract

In this project, we investigate a classical approach of an Electroencephalography (EEG) based gaze estimation. EEG data is often noisy and is affected by baseline drift, and needs to be preprocessed before modelling. We propose a wavelet packet based approach to remove baseline drift from the EEG data. We then use a regression model to map the EEG data to the eye movement data and compare results to basic linear regression.

Contents

Acknowledgements **i**

Abstract **ii**

1 Introduction **1**
- 1.1 Types of eye movements . . . 1
- 1.2 Dataset . . . 2

2 Theory **3**
- 2.1 Preprocessing: Kalman Filters and Median Filters . . . 3
  - 2.1.1 Kalman Filters and Sensor Fusion . . . 3
  - 2.1.2 Median Filters . . . 4
- 2.2 Baseline Drift removal and Wavelet Transforms . . . 4
  - 2.2.1 Wavelet Transforms . . . 5
  - 2.2.2 Daubechies Wavelets . . . 8
  - 2.2.3 Saccade Detection . . . 8

3 Experiments **9**
- 3.1 Preprocessing . . . 9
- 3.2 Wavelet Transform . . . 10
- 3.3 Linear Regression . . . 10
  - 3.3.1 Saccade Detection . . . 10
- 3.4 Results and Plots . . . 11
  - 3.4.1 Results . . . 11
  - 3.4.2 Plots . . . 11
- 3.5 Discussion and Outlook . . . 15

Bibliography **16**

Chapter 1

Introduction

Electroencephalography (EEG) is a non-invasive and convenient method of measuring electrical brain activity through electrodes placed on the scalp. It is recorded as a series of signals representing the brain activity over time, which can then be analysed to detect any abnormalities and patterns in brain activity. In medical applications, EEGs are often used to detect neurological and psychiatric conditions such as epilepsy [1, 2].

In works by Kastrati et al. [3, 4], state-of-the-art deep learning models have been deployed to investigate the possibility of using EEG signals containing Electrooculography (EOG) electrodes to predict eye movements recorded by an eye tracker. However, these models are not easily explainable, and the reliability of the predictions is not guaranteed. In this project, we investigate the possibility of using a more traditional approach to perform the same task, by using a wavelet packet and regression-based model to map signal to signal.

The ideas in this project closely follow the work of [5, 6]

1.1 Types of eye movements

We aim to extract three different eye movement features from our model: saccades, fixations and blinks [7].

- *Saccades* are fast eye movements that rapidly move the gaze from one location to another, causing an instant change in gaze position. These fast, coordinated movements are used primarily to shift the direction of gaze toward an object of interest. Saccades can be executed voluntarily (e.g., when reading) or involuntarily in response to a reflex or stimulus.
- *Fixations* are defined as time periods without saccades.
- *Blinks* are short periods of time where the eyes are closed and are a special case of a fixation.

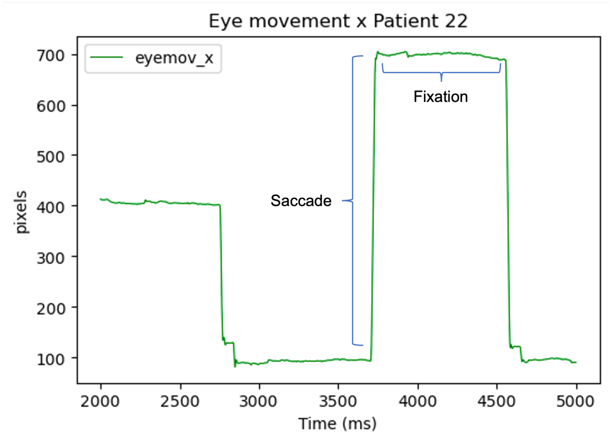

Figure 1.1: Eye Movement with saccade and fixation marked

1.2 Dataset

Data from EEGEyeNet (http://www.eegeye.net) [3] was used. Specifically, the dataset "path task with dots" was used, consisting of a 128-channel EEG signal time series as well as eye movement data in a single stream. The "Large Grid" experimental paradigm was used, where participants were asked to fixate on a series of dots that are sequentially presented each at one of 25 different screen positions.

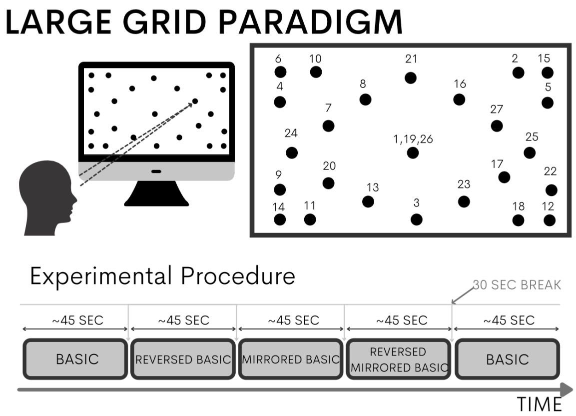

Figure 1.2: Large Grid: Experimental setup

Chapter 2

Theory

2.1 Preprocessing: Kalman Filters and Median Filters

Initially, a Kalman filter was used to preprocess the EEG data. The motivation behind this was that fusing information from multiple data streams provides the ability to combine several inaccurate and noisy sensors into a combined unit with increased performance. Since the EEG data was very noisy, that fusing data from multiple electrodes would provide us with a smoothening effect on the EEG signal, producing better eye tracking predictions.

2.1.1 Kalman Filters and Sensor Fusion

This section follows [8].

Kalman filtering is an algorithm used to fuse information from a series of measurements and sensors to produce a more accurate estimate. It consists of two main steps: the prediction step and the update step. In the prediction step, the filter predicts the joint probability distribution over the variables of current state of the system based on the previous state estimate and its dynamics. The prediction also provides an estimate of the uncertainty associated with the predicted state. In the update step, the filter updates the state estimate and the associated uncertainty based on the new measurement and the measurement model.

Consider a recursive linear dynamical system Kalman filter:

$$x_{k+1} = F_k x_k + B u_k + w_k \tag{2.1}$$

$$y_k = H_k x_k + e_k \tag{2.2}$$

$$E(x_1) = \hat{x}_{1|0}, \tag{2.3}$$

where we assume that at time k, we would like to estimate the parameters of teh pdf of the state x_k given information of the measurements $y_1, \ldots, y_k$, and where $\hat{x}_{k|j}$ denotes the estimate of x_k given information of the measurements $y_1, \ldots, y_j$.

Additionally, we can use Kalman filters as a signal smoothening technique. By delaying the signal by n samples, we can essentially create a moving average filter (for further reading, see [9]. The smoothening can be seen below:

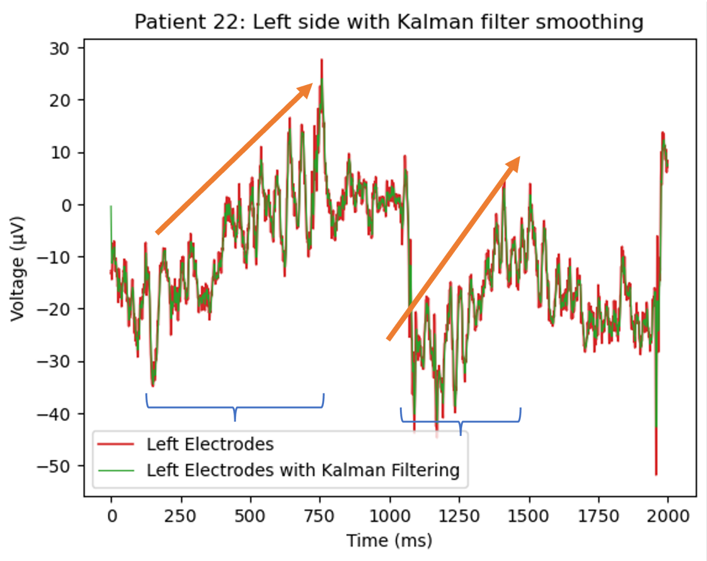

Figure 2.1: Kalman filter applied to left side EEG. Orange arrows indicate baseline drift

During the smoothening process, we also noticed baseline drift (see Figure 2.1 above). The orange arrows indicated a slow signal change in the EEG signal, which the Kalman filter alone was not able to remove. In addition, in order for the Kalman filter to work most effectively, we would need to have a model of the dynamics of the system, which are not available in the case of a 128-channel EEG signal. Therefore, we decided to use a different preprocessing technique, the 1D median filter, to simplify the smoothening process, and used a wavelet packet transform to remove baseline drift.

2.1.2 Median Filters

A median filter is a non-linear digital filtering technique, often used to remove noise from an image or signal. It replaces each pixel with the median of its neighboring pixels. This reduces the effects of impulse noise in an image or signal. The median filter is a non-linear filter, meaning that the output is not a simple function of the input.

2.2 Baseline Drift removal and Wavelet Transforms

Baseline drift is a defined as a slow signal change in the EEG signal. Baseline drift does not affect saccade detection, but it does affect detection of fixations and blinks. To perform baseline drift removal, we follow an approach based on electrocardiography (ECG) signal processing [6]. The proposed algorithm performs a multilevel 1D wavelet decomposition at level 9 using Daubechies wavelets on the left, right and front channels of the EEG signal. The reconstructed decomposition coefficients are removed, and the remaining coefficients are used to estimate the baseline drift. The baseline drift is then subtracted from the original EEG signal, yielding the corrected signals with reduced baseline drift.

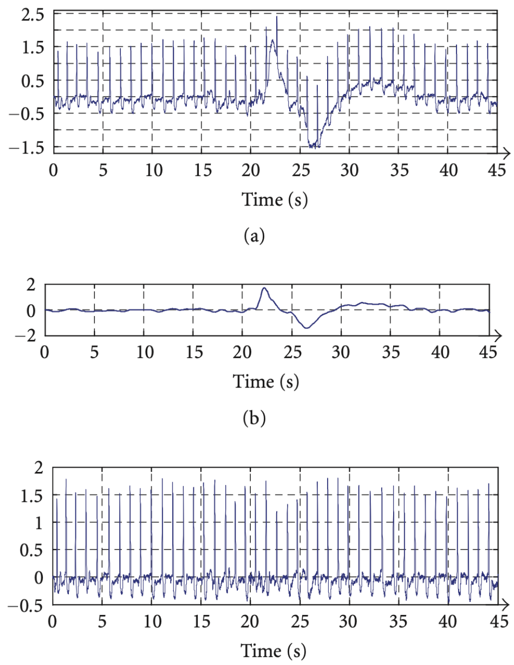

Figure 2.2: ECG Signals with estimated baseline drift, as well as its removal. From [6, Figure 6]

2.2.1 Wavelet Transforms

Wavelet transforms are signal processing methods used to decompose signals into a linear combination of simpler functions called wavelet functions. They are related to Fourier transforms: whereas a Fourier transform creates a representation of the signal in the frequency domain, the wavelet transform creates a representation of the signal in both the time *and* frequency domain. This allows us more efficient access of localized information about the signal [10].

A wavelet system is a two-dimensional system used to construct a signal. We would like to decompose the signal $f(t)$ into basis functions of a mother wavelet $\psi(t)$, which is a function of time. The wavelet decomposition of our signal $f(t)$

is defined by the following equation:

$$f(t) = \sum_{k=-\infty}^{\infty} c_k \varphi(t-k) + \sum_{k=-\infty}^{\infty} \sum_{j=0}^{\infty} d_{jk} \psi\left(2^j t - k\right) \tag{2.4}$$

where c_k are the scaling or coarse coefficients, d_{jk} are the wavelet or detail level coefficients, and $\varphi(t)$ is the father wavelet.

The mother and father wavelets are related by the following recursion:

$$\psi(t) = \sum_{k=\infty} \sqrt{2} h_1(k) \varphi(2t-k) \tag{2.5}$$

$$\varphi(t) = \sum_{k=\infty} \sqrt{2} h_0(k) \varphi(2t-k), \tag{2.6}$$

where h_0 and h_1 are the are low-pass and high-pass filters respectively. The benefit of this recursive nature is that, once one of the c_ks is known, the rest of the coefficients can be simply calculated with a linear transform.

We then follow the following schematic for baseline drift removal:

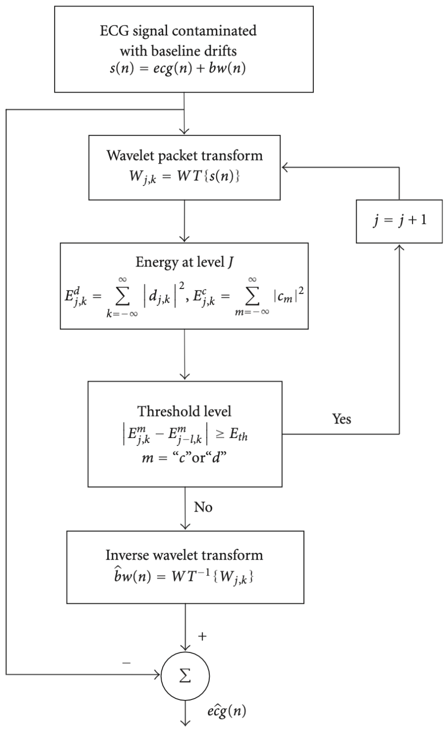

Figure 2.3: Baseline Drift Removal Flowchart, from [6]

We cannot use simple digital filtering to remove baseline drift, since it may also remove fixations. Thus we assume that the EEG signal and baseline drift are a linear combination of two independent signals,

$$f(t) = EEG(t) + BL(t)$$

Firstly, the wavelet transform of the signal is computed using a Daubechies-4 mother wavelet. Since high-frequency components are mostly focused on low-level scales, so it is expected to observe the baseline drift in larger scales. [6] then proposes that in each scale using the wavelet coefficients, the energies of the signal for both the coarse and detail levels are calculated. These energies represent the energy of the decomposed signal in assumed scales as

$$E^c_{j,k} = \sum_{k=-\infty}^{\infty} |c_k|^2, \qquad E^d_{j,k} = \sum_{k=-\infty}^{\infty} |d_{j,k}|^2.$$

In the above equations, $E^c_{j,k}$ is the energy of the signal in the coarse level of scale j (low-pass filtering branch), and $E^d_{j,k}$ is the energy in the detail level of the signal at scale j (high-pass filtering branch).

The following step in the algorithm involves comparing energy levels and selecting the binary tree branch with the highest energy. This process determines the best basis functions for the decomposition. The algorithm follows the path of the higher energy branches until the energy difference exceeds a predetermined threshold level, E_{th}. At this point, the binary tree search ends and the baseline drift signal is obtained by performing the inverse wavelet transform of the wavelet packet coefficients of the last scale. To eliminate the baseline drift, the estimated baseline wander is subtracted from the original data record, resulting in a baseline drift-removed EEG signal.

We will now explain the different types of wavelets in more mathematical detail.

2.2.2 Daubechies Wavelets

The Daubechies wavelets are a family of orthogonal wavelets with compact support, meaning that their wavelet coefficients are nonzero for only a finite number of scales. They also have a high degree of regularity, which makes them computationally efficient [11].

The Daubechies wavelets are identified by a number, such as "Daubechies-4" or "db4", which indicates the number of vanishing moments. The vanishing moments are a measure of the smoothness of the wavelet function, and the higher the number of vanishing moments, the smoother the wavelet function.

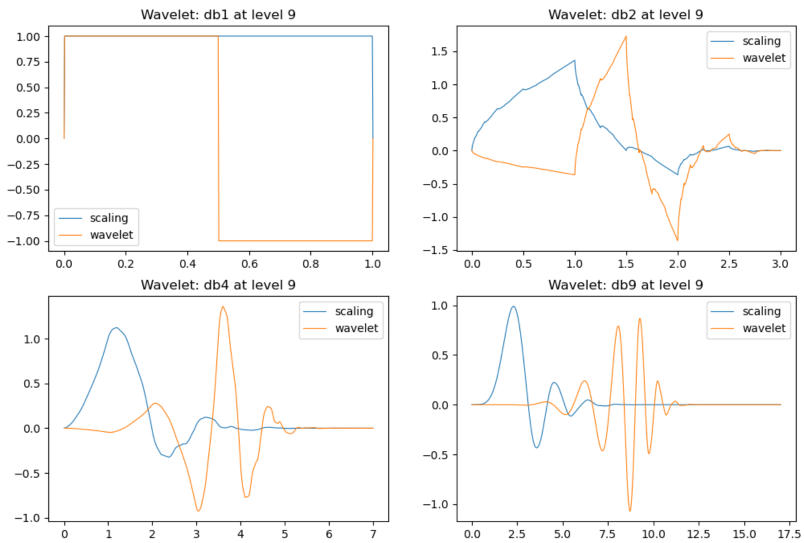

Figure 2.4: Daubechies wavelet plots for db1, db2, db4 and db9

2.2.3 Saccade Detection

We attempted to reduced the aggressive trend removal of the initial baseline drift removal algorithm by accounting for saccades. In a similar fashion to [5, Section 4.3.1], we firstly analysed the EEG stream in batches and used a basic thresholding method to detect possible saccades. The maximum and minimum values of the EEG signal were calculated for each batch, and if the difference between the maximum and minimum values was greater than a threshold, then a saccade was detected and the baseline drift was not performed for that batch.

Chapter 3

Experiments

For the experiments, data from EEGEyeNet [3] was used. The dataset "dots" consists of a 128-channel EEG signal time series as well as EOG eye movement data. A total of 8 channels were used, based on the Top3 electrodes from [4, Section 4.3]: 128, 32, and 38 for the left electrodes, 125, 1 and 121 for the right, and 17 for the front.

Code for all experiments can be found at
https://github.com/way-ze/eegeyetracking

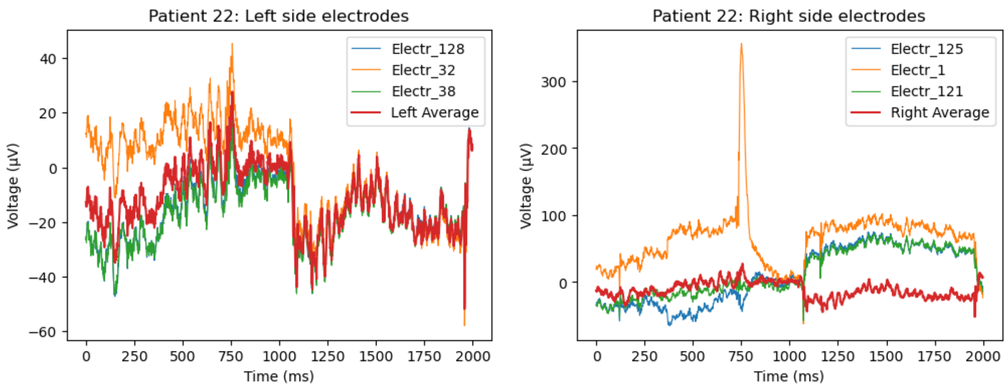

Figure 3.1: Left and right EEG channels averaged

3.1 Preprocessing

We use the median filter to smoothen out the signal. From now all, all data analysis will be perfomred on the filtered signal components.

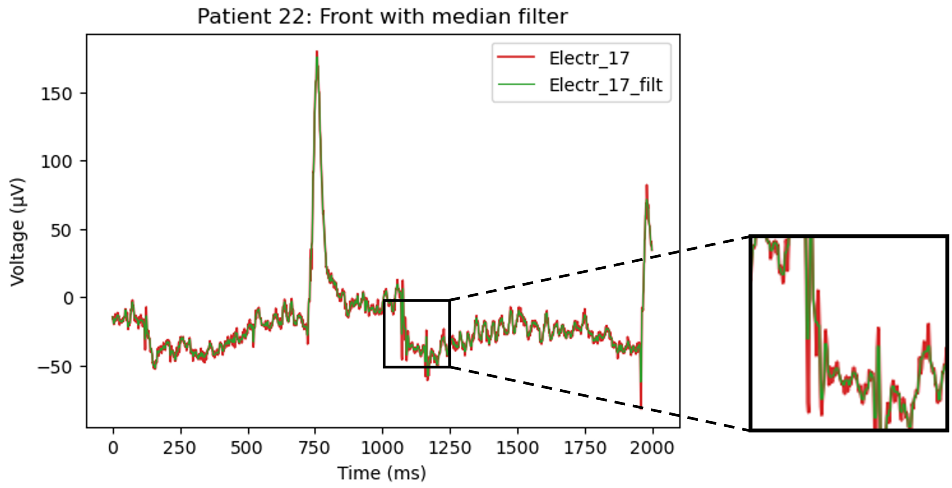

Figure 3.2: Front EEG channel with median filtering

3.2 Wavelet Transform

The package `pywavelets` [12] was used to perform the multilevel 1-D wavelet transform at level 9. The Daubechies-4 wavelet was used.

What was observed was that the coarse coefficients (denoted by `pywavelets` as `cA`) made the biggest difference in terms of baseline drift removal (see figure 3.3). Setting all the `cA` coefficients to zero exhibited in setting the mean of the signal to zero. This is because removing the coarse coefficient essentially removes *trend* elements. Removing the detail coefficients exhibited in a smoothing effect (see figure 3.4).

3.3 Linear Regression

The python package `sklearn` was used to perform linear regression on the baseline transformed data with coefficients `cA, cD7` and `cD8` removed. The linear regression model was trained on the `EP22_DOTS1_EEG` consisting of 156908 samples, and then tested on another data series from the same patient `EP22_DOTS2_EEG` with another 157658 samples. The MSE was calculated on the test set.

3.3.1 Saccade Detection

Following Section 2.2.3, we used a basic thresholding method to detect possible saccades. The thresholds were set to 60 for the left and right channels, and 100 for the front channel and were tuned by manual inspection. Batch size was set to 400 samples.

3.4 Results and Plots

3.4.1 Results

Method	MSE x	MSE y
Linear Regression, no BLR	**16568**	31144
Basic BLR, cA, cD7, cD8 removed	42622	26624
BLR, cA, cD7, cD8 removed with Saccade	33806	**25540**

3.4.2 Plots

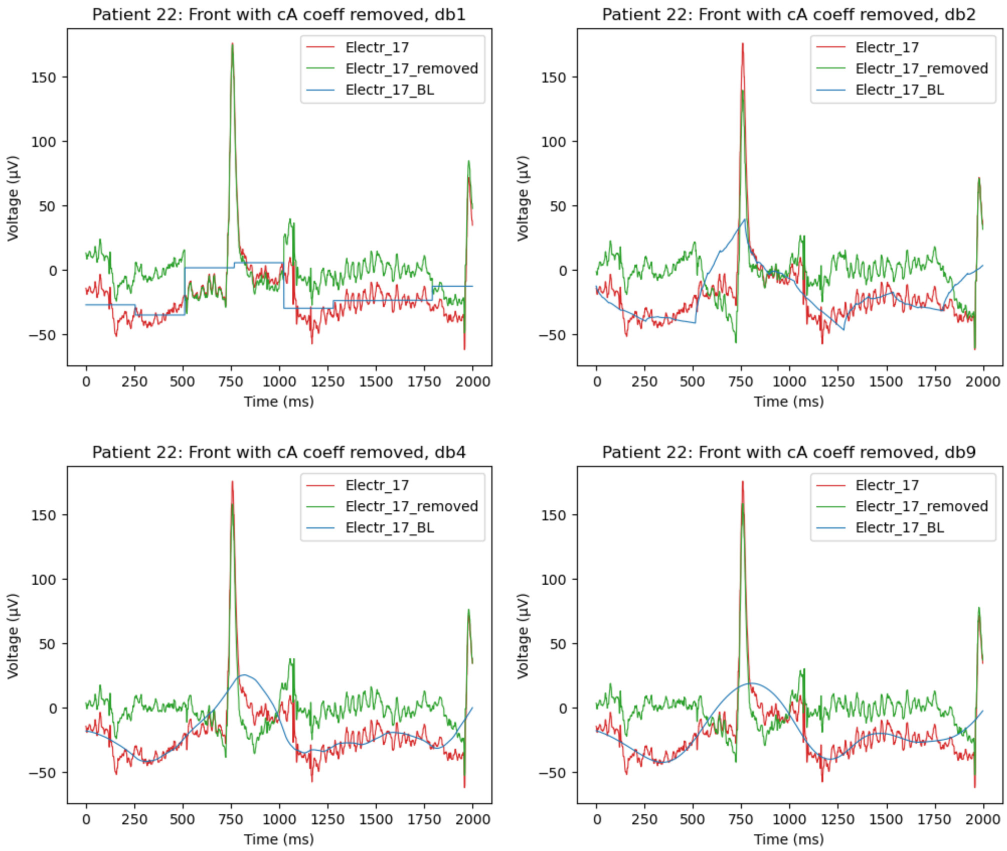

Figure 3.3: Wavelet transform of EEG signal with and without coarse coefficients. Note the similarities between the baseline (blue) and the scaling functions 2.4

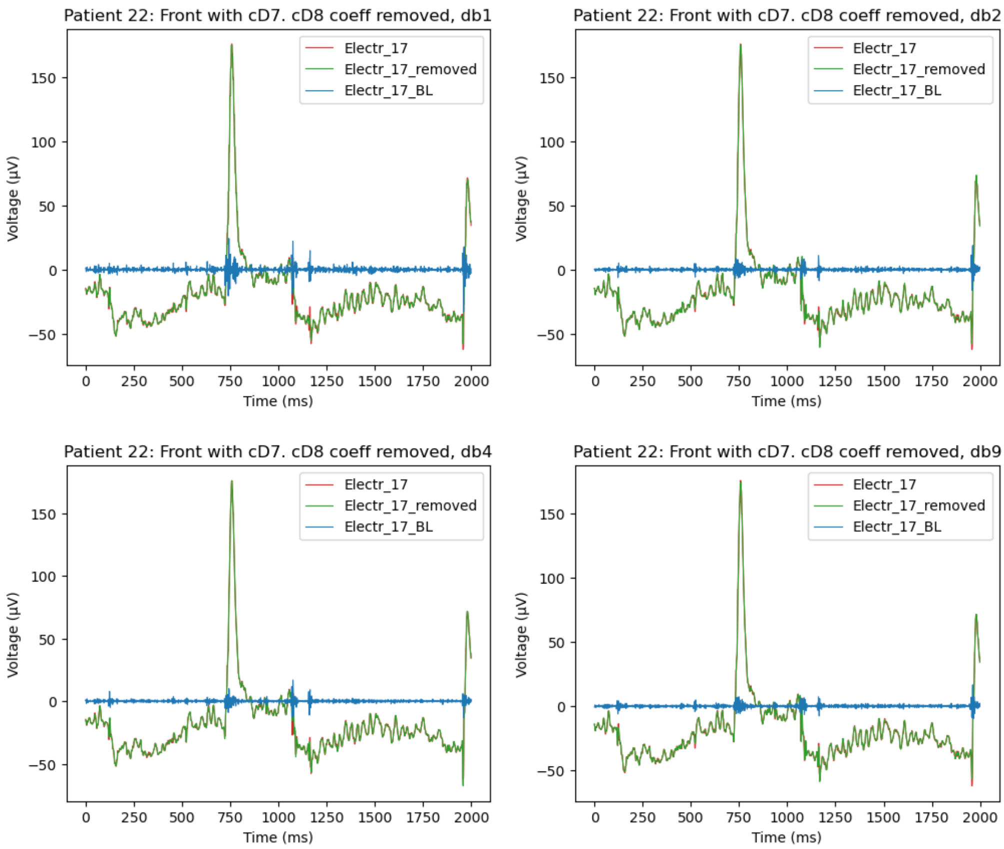

Figure 3.4: Wavelet transform of EEG signal with and without detail coefficients 7 and 8, resulting in a smoothening effect 2.4

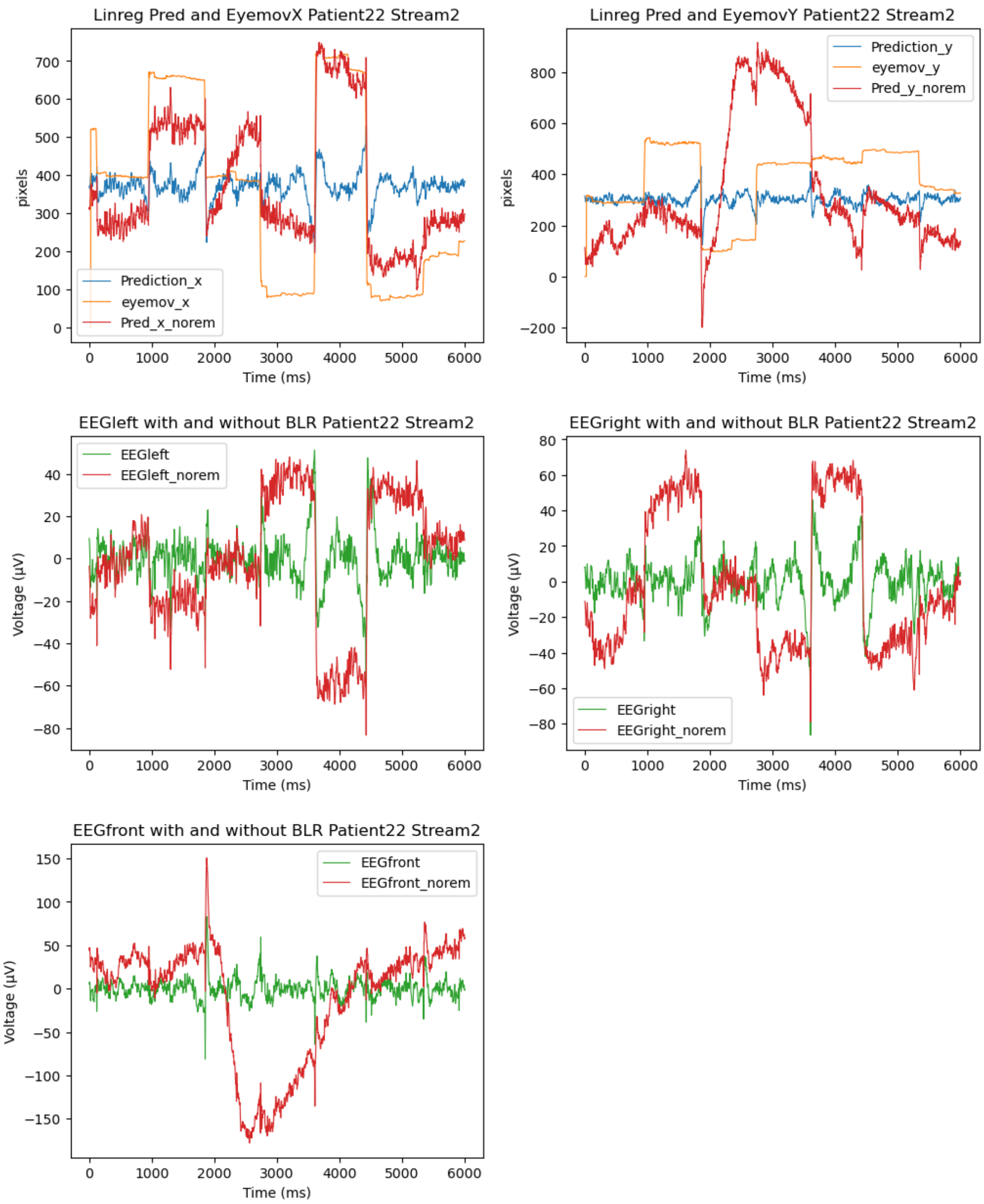

Figure 3.5: Linear regression performed on baseline drift removed EEG signals

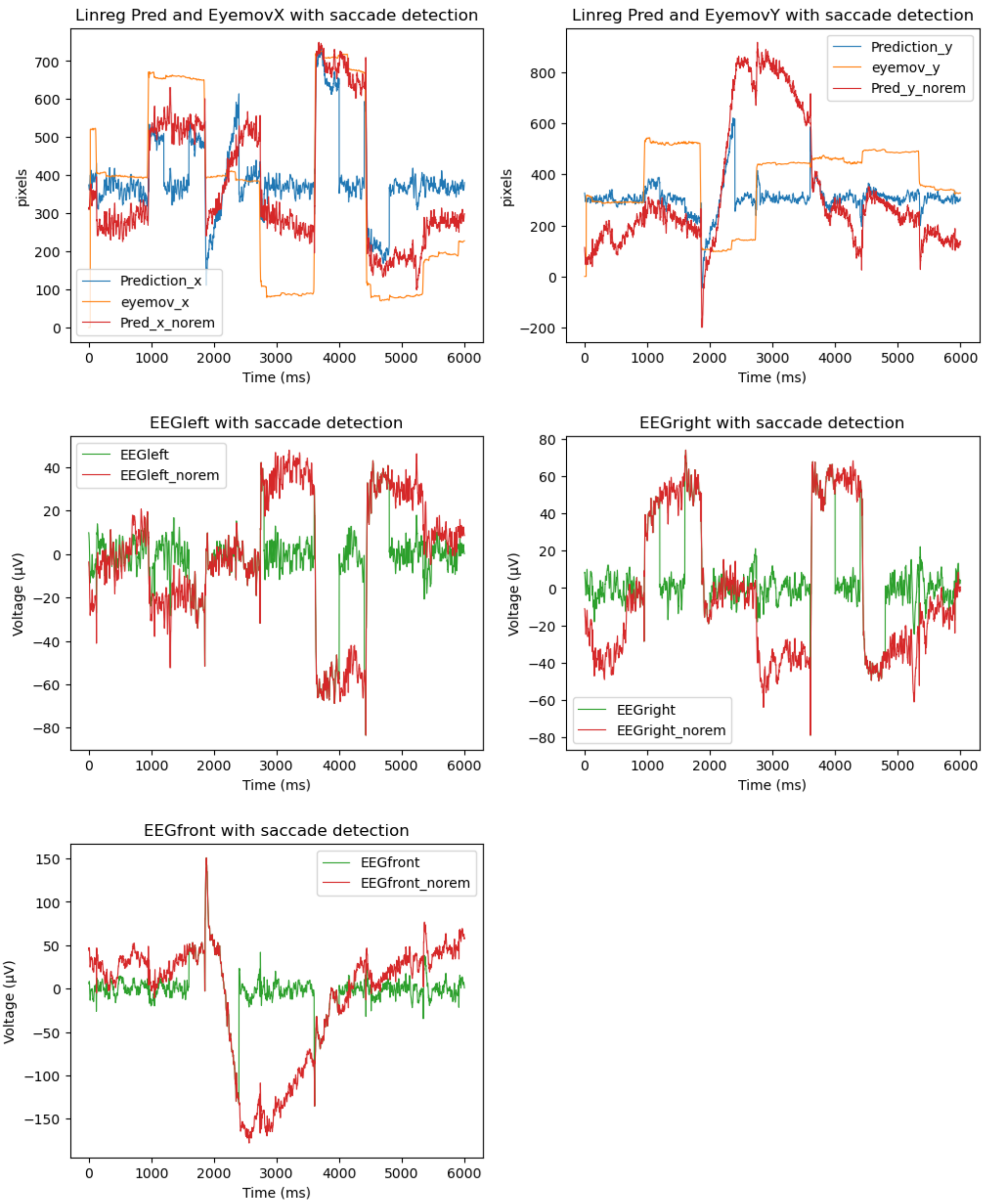

Figure 3.6: Linear regression performed on saccade thresholding baseline drift removed EEG signals

3.5 Discussion and Outlook

The plots show that the baseline drift removal is not functional. The baseline drift removal algorithm effect can be seen in Figure 3.5. We notice a similar effect to Figure 3.3 where the entire trend was removed, and the predicted eye movement was centered around the mean of eye movement $x = 400$ and $y = 300$ respectively. This phenomenon can be further observed in the three other plots with the left, right and front EEG signals at the same time points, where the trend is completely removed.

With the naive saccade detection thresholding method, we were able to reduce the MSE from the basic baseline removal algorithm, however, in the eye tracker x-component, the MSE was still higher than basic linear regression without baseline drift removal.

Performance of the baseline drift removal algorithm can be further improved by using a more sophisticated saccade detection algorithm. For example, when saccades are not detected, the baseline drift removal algorithm means the EEG signal to 0 for that window instead of where the saccade was left off, resulting in an artificial jump in the EEG signal.

Another possible improvement would to use a more sophisticated thresholding method incorporating a continuous wavelet transform as described in [5, Section 4.3.1].

Lastly, instead of using linear regression, we could use other regression models such as LASSO or Ridge regression to further improve the predictions.

Bibliography

[1] [Online]. Available: https://en.wikipedia.org/wiki/Electroencephalography

[2] [Online]. Available: https://www.mayoclinic.org/tests-procedures/eeg/about/pac-20393875

[3] A. Kastrati, M. B. Płomecka, D. Pascual, L. Wolf, V. Gillioz, R. Wattenhofer, and N. Langer, "Eegeyenet: a simultaneous electroencephalography and eye-tracking dataset and benchmark for eye movement prediction," 2021. [Online]. Available: https://arxiv.org/abs/2111.05100

[4] A. Kastrati, M. B. Plomecka, J. Küchler, N. Langer, and R. Wattenhofer, "Electrode clustering and bandpass analysis of EEG data for gaze estimation," in *NeuRIPS 2022 Workshop on Gaze Meets ML*, 2022. [Online]. Available: https://openreview.net/forum?id=We5h1Vcezei

[5] A. Bulling, J. A. Ward, H. Gellersen, and G. Troester, "Eye movement analysis for activity recognition using electrooculography," 2011.

[6] M. A. Tinati and B. Mozaffary, "A wavelet packets approach to electrocardiograph baseline drift cancellation," 2006.

[7] A. Go Lee, A. Kini, and B. Al Othman, "Saccade," Sep 2022. [Online]. Available: https://eyewiki.aao.org/Saccade

[8] B. Bernhardsson, "Lecture notes: Modeling and learning from data - bayesian estimation and sensor fusion." [Online]. Available: https://canvas.education.lu.se/courses/6993

[9] S. Markou, "The kalman filter and smoother." [Online]. Available: https://random-walks.org/content/misc/kalman/kalman.html

[10] K. Wirsing, "Time frequency analysis of wavelet and fourier transform," in *Wavelet Theory*, S. Mohammady, Ed. Rijeka: IntechOpen, 2020, ch. 1. [Online]. Available: https://doi.org/10.5772/intechopen.94521

[11] I. Daubechies, *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, 1992. [Online]. Available: https://epubs.siam.org/doi/abs/10.1137/1.9781611970104

[12] G. R. Lee, R. Gommers, F. Waselewski, K. Wohlfahrt, and A. O'Leary, "Pywavelets: A python package for wavelet analysis," *Journal of Open Source Software*, vol. 4, no. 36, p. 1237, 2019. [Online]. Available: https://doi.org/10.21105/joss.01237