

RADBOUD UNIVERSITY NIJMEGEN

Template Matching for Artifact Detection and Removal

by

R.Barth

supervised by

prof. dr. ir. P.Desain and drs. R. Vlek

A thesis submitted in partial fulfillment for the
degree of Bachelor of Science in Artificial Intelligence

in the

Faculty of Social Sciences

Department of Cognitive Artificial Intelligence

July 2009

RADBOUD UNIVERSITY NIJMEGEN

Abstract

Faculty of Social Sciences
Department of Cognitive Artificial Intelligence

by R.Barth
supervised by
prof. dr. ir. P.Desain and drs. R. Vlek

In this thesis a method for artifact detection and removal in EEG is presented and tested. This method is based on a template matching technique using cross-correlations in the time domain. A template is created by averaging hand-picked examples of the artifact. Removal of the artifacts is done with use of three different template subtraction methods. The quality of the removal is assessed by an averaging paradigm, as well as a frequency analysis. Briefly the algorithm's generalizability is tested with use of a secondary data set containing different artifacts. At last a hypothesis on the source of the artifacts is presented. Results show that template generation, artifact detection as well as removal is successful. Generalizability to other artifacts is good, but performs slightly worse.

“Working with Artifacts you need strong fingers.”

- Urza, Lord High Artificer

Acknowledgements

I would like to thank drs. Rutger Vlek and prof. dr. ir. Peter Desain as primary advisors for their help and ideas during the whole period of my internship. It was great working with you together. Also I would like to thank dr. Jason Farquhar for his initial ideas and implementation during the beginning of the project. The whole research group deserves my compliments as well, for they were always there for help, even in busy periods. To my fellow interns, thanks for the wonderful cooperation. I will certainly miss working with you together in the same lab. The ambient of the the total group was always positive in every sense, I couldn't have found a better place to work.

Contents

Abstract	i
Acknowledgements	iii
1 Introduction	1
1.1 Problem Definition and Hypotheses	1
1.2 Previous Work on Artifact Detection and Removal	3
2 Specification Data sets	5
2.1 Primary and Secondary Data sets	5
2.2 Visualization of Artifact Polluted Channel	6
3 Artifact Detection: a Template Matching Method	7
3.1 Introduction	7
3.2 Establishing Ground Truth	8
3.3 Template generation	8
3.3.1 Generating Template by Averaging	9
3.3.2 Generating Template by Genetic Algorithms	10
3.3.2.1 General Evolutionary Computing Theory	10
3.3.2.2 Representation of Individuals	11
3.3.2.3 Environment and Fitness Function	12
3.3.2.4 Parental Selection	13
3.3.2.5 Recombination Techniques	13
3.3.2.6 Results	13
3.3.3 Template Comparison	14
3.3.4 Variability in Artifact Occurrences	15
3.4 Template matching using cross correlation techniques	17
3.4.1 Matching Performance	21
3.4.1.1 Cross Validation	21
3.4.1.2 Influence of sample size	23
4 Artifact Removal	24
4.1 Whole template subtraction	24
4.2 Decorrelation and Amplitude adjusted template subtraction	25
4.3 Raised Cosinus Filtering Prior to Subtraction	25
4.4 Removal Results	25

4.5	Multi channel templates	30
5	Discussion	32
5.1	Generalizability	32
5.1.1	Matching results	33
5.1.2	Remarkabilities	33
5.2	Artifact Source	34
6	Conclusion	37
A	Matlab Template Matching Code	41

Chapter 1

Introduction

1.1 Problem Definition and Hypotheses

In the field of brain activity research, scientists are interested in recording proper cerebral activity. The most common techniques of today for retrieving these brain activity signals include functional magnetic resonance imaging (fMRI) and electroencephalography (EEG). Unfortunately, both of these techniques suffer from artifact intrusions which mask the cerebral signal. In this thesis, we shall focus on artifacts in electroencephalography. These artifacts are defined as unwanted recorded electrical activities arising from sources other than the cerebral matter. They can be divided into two categories: physiologic and extra physiologic artifacts. Physiologic artifacts are generated by the subject, but they arise from sources other than the brain (e.g. body muscle-, glossokinetic-, respiration- and skin artifacts). Extra physiologic artifacts arise from outside the body (e.g. equipment 50hz noise, movement in the environment or high-frequency radiation interference from electronic devices).

Artifacts are not necessarily a bad thing. Some of these artifacts can be correlated with certain brain activity. For example, the physiologic eye movement artifacts are useful for predicting sleep stages [6]. Although these implications of artifacts might be useful in this case, artifacts are a problem for neuroscientists only interested in signals originating from the brain, like researchers of Brain Computer Interfaces. Artifacts make it harder to retain focus on true brain signals, artifacts mask the brain signal and thereby making further analysis more complicated. For Brain Computer Interfaces, the masking effect of artifacts may cause performance drops of the system.

Ideally we would like not to record artifacts in the first place. However, methods to achieve this are intrusive to our subjects and still only reduce certain types of artifacts,

leaving others to distort. Intracranial electroencephalography (I-EEG) for example, sometimes called sub-dural electroencephalography (SD-EEG), is a method where the scalp of the subject is removed, and electrodes directly on the brains are placed. This greatly reduces physiologic artifacts but also presents a risk of infection to the subject. Aside from the extra physiologic artifacts which still remain, this method is obviously not favorable yet. Other methods use paralyzing drugs in order to relax the muscles to prevent unnecessary movement, reducing thereby physiologic artifacts. This method also has clear downsides for it paralysis the subjects, disabling them more than necessary.

Because we cannot properly evade recording of a subset of artifacts, there are two ways we could handle the retrieved electroencephalography data. The first considers discarding the data wherever artifacts are found. This method is not favorable due to the fact that a lot of data will be lost after selection. The second method is the one we will address in this thesis. It is the post processing of the data. This method tries to localize a certain type of artifact whereupon we try to eliminate the artifact from the signal, restoring the signal a step closer to its pure cerebral form.

However, multiple problems have to be dealt with using this approach. The first is dealing with the recognition of possible artifacts, which entails the question when a piece of signal distorted by what kind of artifact. We have to identify the occurrence of an artifact in time, as well as the waveform of electrical activity of the signal. However it is not known how this precise electrical signal of the artifact is composed from its sources. We can only record the resulting EEG, which is a mix of cerebral activity and artifacts. In order to solve this problem, clever techniques have to be used in order to specify the general form of the artifact. A way to overcome this problem is suggested and tested in this thesis: using templates and template matching techniques.

The second problem is the artifact removal. Once we know when an artifact occurs and what it looks like, we want to separate the artifact from the rest of the signal. In this thesis methods of removal are proposed and tested, trying not to introduce new artifacts instead.

Concluding, 5 main research questions shall be addressed in this paper. First of all can we find one or more templates to quantify the artifact electrical source? Second, can we use it to find the artifacts in time? Third, can we use the template for proper removal? Fourth, is the method suitable for other artifacts? Fifth, what is the source of the artifacts in our primary dataset?

1.2 Previous Work on Artifact Detection and Removal

In order to place the template matching method discussed in this thesis into a broader perspective, it is important to explore the current available techniques in the ongoing research field for artifact removal. Most research done on this topic today focusses on ocular artifacts. This type of artifacts are well visible in the electro oculargram (EOG) and propagates signal to other channels. Removal techniques of ocular artifacts can be generalized to other types of artifacts found in data used in this thesis. We shall briefly present an overview of the current removal techniques around [5].

The first attempts of artifact removal focused on eye blinks using simple regression techniques [11] [10]. These methods rely on using the electro-oculogram to locate where artifacts occur. After this detection a part or a certain factor of the EOG is subtracted from the EEG. Since the EOG also contains signal from the EEG, this method also undesirably removes partially the signals of interest. However, subtracting EOG from the EEG is still common use.

Multivariate statistical analysis techniques, such as independent component analysis (ICA) [9] [13], are a newer generation of methods based on source separation. ICA assumes EEG observations are generated by the linear mixing of a number of source signals which are statistical independent. It tries to separate the different sources, of which one specific source can be discarded before the signal is put back together. This way an artifact signal can be detected [1] and/or removed [14] [15] [2]. These techniques are currently of most common use to remove artifacts. There is a wide variety between ICA techniques being practiced today [17].

Most methods still rely on visual inspection and are not yet automated. Automation is not only is favorable to reduce scientist's workload, but also important for online brain computer interfaces. Those systems rely on the real time classification of EEG signals, therefore post processing of offline data comes a few steps too late. However, offline processing is also important to retrieve cleaner EEG data to facilitate research. In the Conclusion we will discuss the position of the template matching technique in this spectrum.

The template matching technique discussed in this thesis is relatively unused. In comparison with the other techniques it has the most in common with early regression techniques, where a quantification of the artifact is subtracted from a detected artifact in the data. In contrast to those techniques, the template matching technique will try to reduce the partial removal of EEG signals. The proposed template matching technique will primarily work best low variant types of artifacts. For the detection it can handle more variant types, as it uses a correlation factor for similarity. For proper removal

though, the technique is perfect for non variant artifacts as long as enough samples are available. For variant types it can be adjusted to, for example, handle variance in amplitude. Therefore knowledge about the artifact in question has to be gathered. Techniques such as ICA also prefer non variant artifacts and variance is always a factor of influence. But it can be handled by modeling the artifact. The method in this thesis can specifically do this, in contrast to ICA.

Chapter 2

Specification Data sets

2.1 Primary and Secondary Data sets

Two EEG data sets were used for this paper, each containing differing artifacts. The primary data set was used for intensively testing an artifact matching and removal algorithm. The secondary data set was used for testing generalisation of these algorithms.

Primary data set: This data set was recorded for the use of a Brain Computer Interface using a subjective rhythmization paradigm [18]. The raw EEG signal was originally sampled at 512 Hz and it was downsampled to a frequency of 256 Hz. Spatial downsampling was applied on the original 256 channel cap. The 256 channels were downsampled to a 10-20 cap with 64 virtual electrodes using a local weighed average of the original 256 channels. A time window of -150ms to 350ms was chosen around the presentation of metronome stimuli, resulting in a total time window of 500ms per data segment. These data segments will be called trials. In total 212 trials were recorded of which 8 were discarded due to too invasive muscle artifacts, abnormal amounts of 50 Hz line noise and/or static discharges. Also some preprocessing was done like detrending, low- and highpass filters.

Secondary data set: This data set was also recorded for the use of a Brain Computer Interface, using a imagined music paradigm. The raw EEG signal was originally sampled at 2048 Hz and it was downsampled to a frequency of 256 Hz. Also spatial downsampling was applied on the original 256 channel cap. The 256 channels were downsampled to a 10-10 cap with 64 virtual electrodes using a local weighed average of the original 256 channels. Also some preprocessing was done like detrending.

2.2 Visualization of Artifact Polluted Channel

The primary and secondary data set are contaminated with artifacts. In specific channels they are seen best, these channels are visualized in in figure 2.2 and figure 2.2.

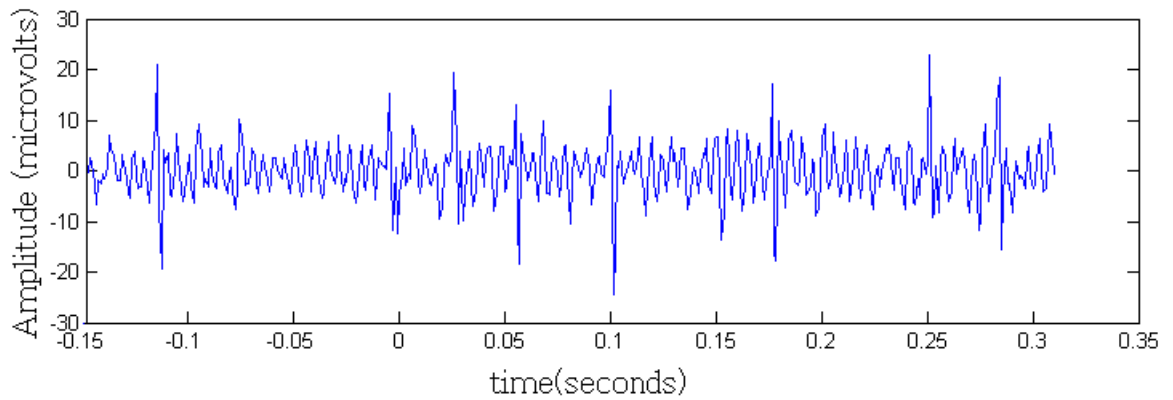


FIGURE 2.1: Artifacts in the primary data set. A single trial of virtual channel 'FPZ' (central-frontal) is visualized.

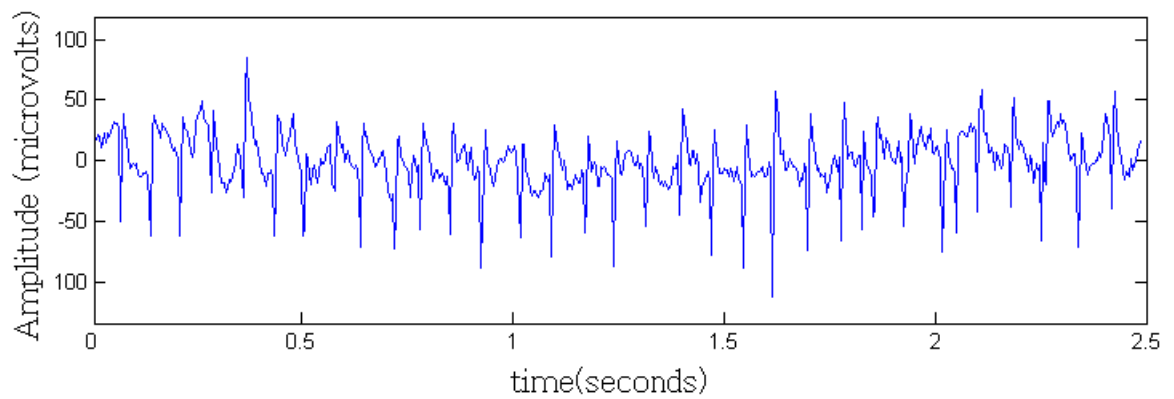


FIGURE 2.2: Artifacts in the secondary data set. A single trial of virtual channel 'FP1' is visualized.

Chapter 3

Artifact Detection: a Template Matching Method

3.1 Introduction

The first step in the process for our eventual goal of artifact removal is the detection of the artifacts in time. However, this is not a trivial task. The recorded signals are a mix of different cerebral sources and artifacts. Therefore we can only observe a weighed sum of all the electrical activity from those sources, where weights are assumed to be distributed differently over each channel following certain linear laws, on each measured channel of the EEG. This makes it hard to manually detect artifacts, because they dissolve between the other signals. This can be circumvented by recording and observing a channel close to the artifact source, which increases the amplitude of the artifact signal relative to the other signals. When plotting the virtual 'FPZ' channel, artifacts can be seen by eye. Note that this is a subjective classification, prone for human errors. The assumption made here is that pieces of signal differing significantly from the rest of the signal, cannot be the result of cerebral activity, but result from other sources.

We want the detection to be automated to act as an artifact filter, trying to reduce human effort and interference to a minimum. The technique chosen here to realize this filter is a template matching approach. Template matching is originally a technique in image processing [4] where a small template image is matched for occurrence in an other image. This strategy can be translated to the field of signal processing where we can use a template piece of signal to match for occurrence in a larger piece of signal. In our case, the template piece of signal should ideally be the isolated electrical signal from the artifact source. The signal to match it with should be the recorded electroencephalography data. The technique to compare the template and the signal shall be based on the

cross-correlation, which shall be further be explained in the following sections. In order to detect the occurrences of artifacts with use of template matching techniques, we first require a template.

3.2 Establishing Ground Truth

In order to generate that template and to check template matching performance later on, we need to establish a ground truth. A ground truth is a user specified subjective classification of the occurrences of artifacts in time. This is top-down information by subjective user input which is necessary in the first steps of our matching. Eventually, when we have specified the template, we do not need this information anymore because it will be implicitly coded in the template. The algorithm is thus not fully automatic because it requires initialisation of the template. In chapter 3, it will be investigated how many samples are needed for a good performance.

The ground truth is established by manually picking time points in the signal where we think the artifacts occur. As noted previously for our primary dataset, this can be done best using the 'FPZ' channel, for there the artifact can be seen best. A small graphical interface helps us defining these points by selecting the artifacts in a plotted signal with the use of a mouse. The selected time points with the mouse are not accurate however. But what defines it to be accurate? Because we cannot say when the artifact starts or ends, we have to assume and define a landmark of the artifact. This landmark is a unique point every instance of the artifact possesses. In this case we define this point as the first high peak in the signal. In order to make our manual estimation more accurate, another algorithm specified in the Appendix, aligns these estimated rough points around the peaks of the signal.

After this process we have attained our ground truth. Note that truth still is a subjective term. In our primary data set we have attained 2872 artifact occurrences manually. In the next section we discuss why the ground truth is relevant to generate our template.

3.3 Template generation

After manually selecting the subjective occurrences of the artifacts in time on a single channel, we can use this information to generate a template. We define a template as the invariant electrical signal which a source, in this case the artifact, contributes to the recorded EEG. Invariant because we assume that this signal does not change majorly over time or over trials. Support for this assumption is given in subsection [3.3.4](#).

3.3.1 Generating Template by Averaging

The first method used to obtain a template is by averaging slices of signals around artifact occurrences [12]. A certain time window is chosen which defines the region of interest around the artifact. Slices are chosen in such a fashion that the landmark, in this case the peaks, are centered in the slice. This uses the previously obtained ground truth since it defines when the artifacts should occur. This averaging method is based on the assumption that signals other than the artifact signal mixed in the slices are all differing between slices. Since the artifact signal occurs in each slice and presumed not differs between slices, the other signals should cancel each other out to an near flat line. That way, we are left with the average of the artifact signal and we can assume that the average signal is highly identical with the true average artifact signal. The more slices we use for averaging, the more other other signals cancel each other out, the more refined our template will be. In this case we used all 2872 artifacts to create slices for the averaging process. The result can be seen in figure 3.1.

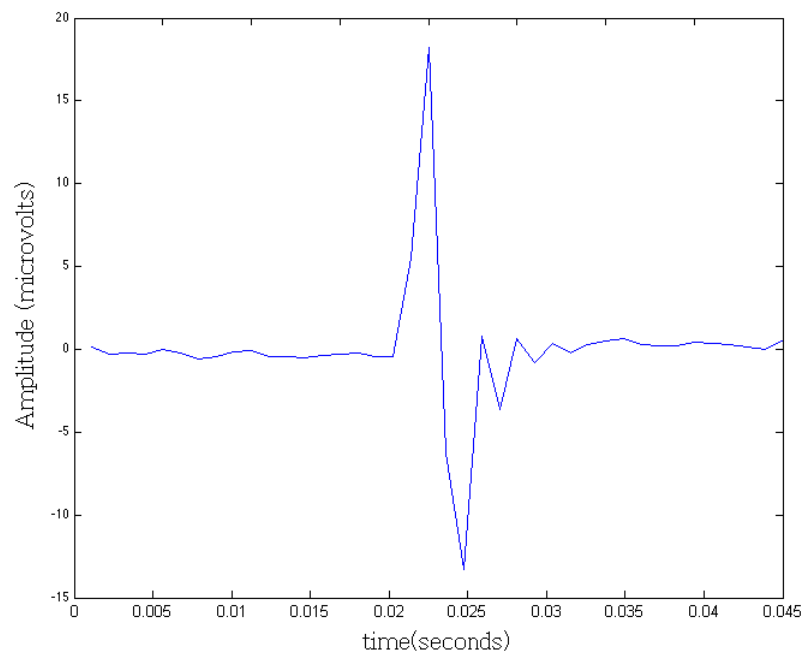


FIGURE 3.1: Artifact signal template by averaging.

The acquired template matches our expectation of its landmarking peaks and of the near zero line before and after its domain. The duration of the artifact approximately 4.5 milliseconds. Also note the decaying oscillation after its initial rapid decline. This could be the effect of a temporal filter in the preprocessing of the data.

3.3.2 Generating Template by Genetic Algorithms

An other approach to generate us a template is by formulating the template problem differently and let it be solved by using evolutionary computing. In this case, we formulate the template problem as the search for the most correlating signal with all the slices of mixed signals where artifacts occur. The assumption here is that if a signal highly correlates with all slices of data which contain one artifact each, than that signal should also highly correlate with the ideal isolated electrical signal from the artifact source. There are several techniques to maximize correlations, however my personal interest in evolutionary computing thrived me into a genetic algorithmic solution.

3.3.2.1 General Evolutionary Computing Theory

Generally speaking, evolutionary computing is based on Darwin's theory of fittest surviving individuals. In theory, this principal leads to well adapted individuals to their environment. There is strong evidence supporting this theory, not only in our physical world, but also in our simulated computer models [8].

In these computer models an environment is specified and filled with individuals that strive for survival and reproduction. An individual has a genetic coding which represents a probable solution to the problem. Each individual has an amount of fitness, determined by its environment, which directly relates to its chance of survival and reproduction. Parents are selected by this chance to produce children by recombining the genetic material of the parents. The new children form a new population, they are the new generation. Then the process loops until a certain desired fitness level is reached. The pseudo code for this process can be cut down to the following global steps:

- 0) Generate random genetic material for all individuals.
- 1) Calculate the fitness for all individuals.
- 2) Select parents in a fitness proportionate manner and recombine their genetic material into new differing children. Repeat this step until population maximum is reached. Delete the old population.
- 3) Mutate the children slightly to introduce new genetic material.
- 4) repeat from 1

This is a rather global overview of the process, in detail there are many parameters that can be set. Genetic algorithms sure can operate autonomous, but also in this field of

research there is no such thing as a free lunch. Just as in nature, finding the optimal parameters to let a population flourish is a harsh job to accomplish. The code for implementing this process specific to our problem can be found in the Appendix. The following sections will discuss this implementation and its parameters in further detail.

3.3.2.2 Representation of Individuals

First of all we need a representation of individuals which each represent a solution to our problem. In our case, we are looking for a slice of signal to act as a template for our matching process. In the previous sections this piece of signal was composed of an array of doubles. Fortunately genetic algorithms can handle these kinds of representations very well, so it is trivial to choose the representation as such. As an initial population of individuals, we choose a random set of doubles for each individual. It is common to use random initialisation, for it results in a widely varied genome pool. This wide variance is positive in the sense it does not exclude certain possible solutions we did not have foreseen. Also from an aesthetical point of view, it is beautiful and astounding how one can create a solution for a problem just from random noise. To enhance understanding of our chosen representation, a random genome with 20 time steps as a size is given below. Also, this specific genome from the first generation is visualized in figure [3.2](#).

Individual representation (random genome):

```
[ 12.2145790895318 -13.6412273083556 13.5658570654620 2.09574847210159
11.6448552386956 -9.05046048883771 4.65961418449152 -8.45057813334994
-11.8664558480435 0.00201029373814288 -4.40456644748248 -3.24861021578565
-7.72177901301240 -7.35456550152097 8.57154891253136 -9.88905510185456
0.875746972647026 -8.53477134180811 5.21397861670785 9.17075396173173 ]
```

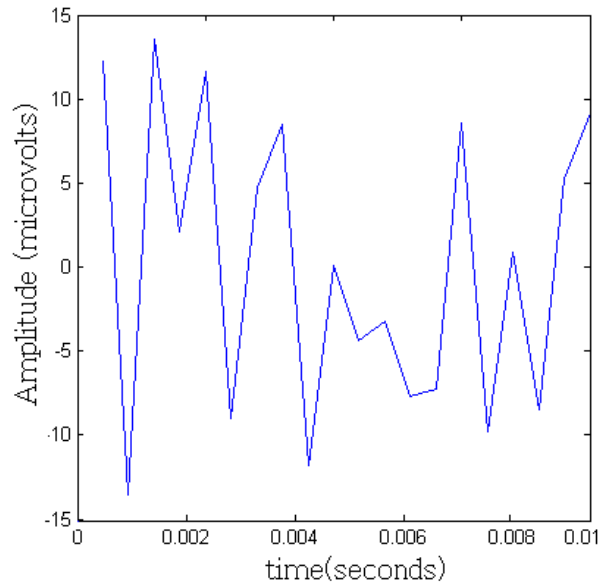


FIGURE 3.2: plotted random genome

3.3.2.3 Environment and Fitness Function

Now we have specified the representation of the individuals, we can determine the environment. The environment has a direct influence on the population of individuals because it determines the fitness of each individual. Furthermore, the environment directly forces the population to evolve since only the fittest get the highest chance to reproduce. We shall not define the environment itself, which is pretty abstract in our case. What we will define is the influence of the environment on the fitness of individuals. In our case, the environment consist of a set of EEG signals. These are pretty static, so we will define the influence of the signals on the individuals with a function of fitness.

We say an individual is highly fit when it highly correlates with all the slices of signals where the artifacts occur. Therefore we use the ground truth to generate these slices, and sum up the correlations with the individual's genome and a slice. This total value

is the fitness the individual receives. The function is based on the assumption that the desired artifact signal should be the one signal which correlates the most with all the artifact occurrences in our data. Therefore, given enough samples, the template found with this approach should correspond highly with the signal we are looking for.

3.3.2.4 Parental Selection

Selecting parents to be used for reproduction is another key process in our simulated evolution. Parental selection can be implemented in a variety of ways. The following methods were implemented for the selection procedures:

- * **Linear Fitness Proportional Selection (LFPS):** The chance of an individual being selected as a parent increases linearly with its fitness. (variants of this method can be based on non-linear functions as well)
- * **Rank Selection (RS):** Only a top percentage of fit individuals is selected as parent.
- * **Tournament Selection (TS):** At random, two or more individuals are selected at random of which only the fittest becomes a parent.

3.3.2.5 Recombination Techniques

After parents are selected, they have to produce offspring to populate the new generation. Also this process can be handled in different ways. The following two techniques were implemented and used:

- * **Single point crossover (SiPC):** The two parents genomes are cut in two at a random crossover point. Two new individuals are now created as children by switching the tail pieces of each parent with one another.
- * **Uniform crossover (UniPC):** Two new children are generated from genes at random of one or the other parent's genome.

3.3.2.6 Results

Before running the simulation, there are many parameters to be set. Finding the optimal set of parameters by hand is unrealistic since there is a combinatorial explosion of available settings. Therefore we restrict ourselves to a subset of settings listed below.

Since we had to save computation time, we restricted our search to a template of only a size of 10 time steps.

```

Population Size : 1000/25000
# Generations  : 10/100
Mutation Rate  : 15% chance per gene
Genome Length  : 10 doubles
Selection      : LFPS/RS/TS
Recombination  : SiPC/UniPC

```

Different combinations were tested and solutions were checked. The simulations did all find the same solution and only differed in computation time. The beautiful process of increasing fitness over generations of a single simulation can be seen in figure 3.3. At the final generation all individuals have the same high fitness value. This found solution can be seen in figure 3.4.

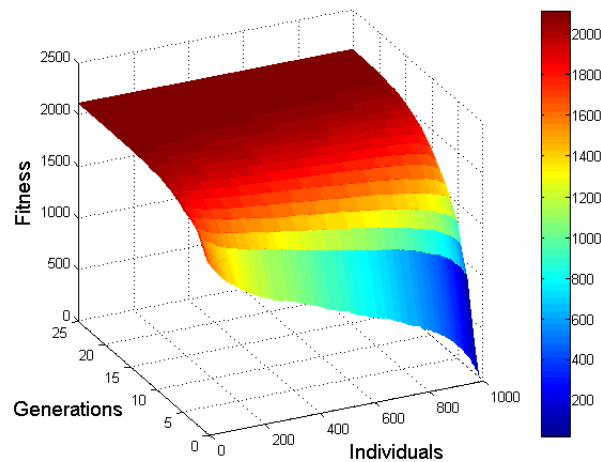


FIGURE 3.3: Fitness increase of whole population over the generations.

3.3.3 Template Comparison

A visual comparison of the retrieved templates by the different methods can be seen in figure 3.4. They are almost identical by eye, this is confirmed by the mutual correlation of 0.998. The equalness of the found solutions support each other in the hypothesis that the template is a proper description of the produced signal of the artifact in the EEG. However, it can be argued that the two methods calculate the same. Since the highest correlating signal can only be the signal most common in all the samples, and that is the average.

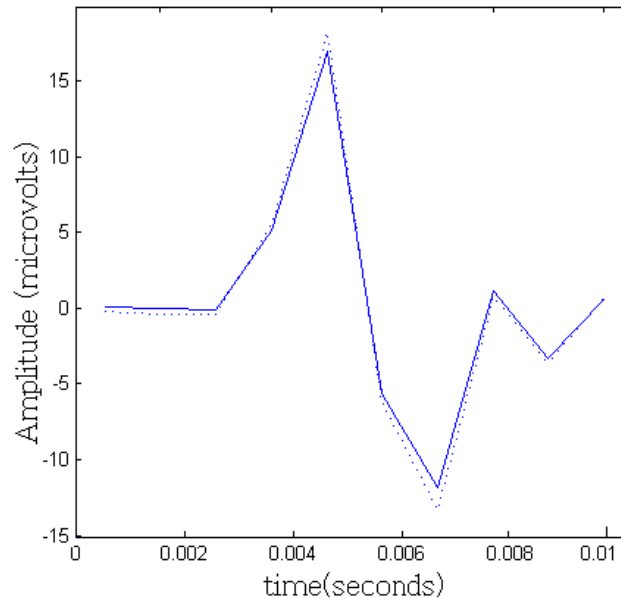


FIGURE 3.4: Templates calculated by different methods. The ‘.’ line is the visualization of the template by averaging. The solid line is the result of the genetic algorithm.

3.3.4 Variability in Artifact Occurrences

A key assumption for using and generating a template is the invariability of the signal of the artifact. In other words, we assume that the artifact does not change over time or between trials for a single subject. This assumption is necessary because otherwise we would use a kind of average of multiple variant artifacts for the template matching, not highly correlating with any of the individual artifacts at all. This in return would result in bad matching performances and inappropriate template removal. It is inappropriate because we would use a template signal to remove an artifact signal which is very different from the template. Therefore we must be sure the variability between the signals of the artifacts is generally low.

We start of with the general form of the artifact signal. It is hard to determine if there are differences between occurrences of artifacts, because we see a mix of artifacts and other signals. We can not really say the general form does not vary between artifacts, but we can assume. This assumption is confirmed when we visualize the data and see that every artifact has the same shape in common: a maximum followed by a minimum.

Next we investigated what we *can* measure and calculate the mean and standard deviation of the amplitudes all artifact occurrences This resulted in MEAN: 20.0709, STD: 4.6736 . The standard deviation seems higher than wished for. However in figure 3.5, where we plot two artifacts having this variability, we can intuitively see that differences

are only slight relative to their mean. So the amplitude does seem to vary, however almost certainly not enough to discard the assumption.

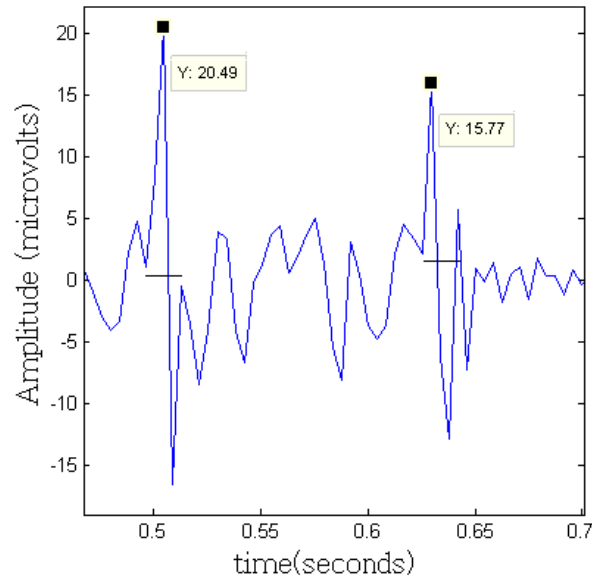


FIGURE 3.5: Two successive artifacts occurring, differing approximately 1 STD in amplitude. The horizontal lines point out the duration.

We should also check that the amplitudes over time do not change. This type of variance information could be useful in order to generate more accurate templates for certain time slots. The distribution of the ground truth can be seen in figure 3.6. As you can see, there is almost a horizontal distribution. The correlation between the two factors is -0.0153 , which implies there is very little change of the amplitudes over time. Note that at the end of a few signals, there are some outliers in amplitude. When we check these artifacts in our data these are erroneous drifts, probably due to movement of our subjects at the end of the trial when they lose focus.

What also could change over time is the general signal form of the artifact. This topic is further addressed in 3.4.1.1. There we also check if this possible variation in time has influence on the performance of our matching algorithm.

Another property of the artifacts that could be variant, is the duration. In figure 3.5 the durations are indicated with the horizontal lines, measured from the beginning of the first maximum, until the end of the first minimum. In our data, they are all approximately 55

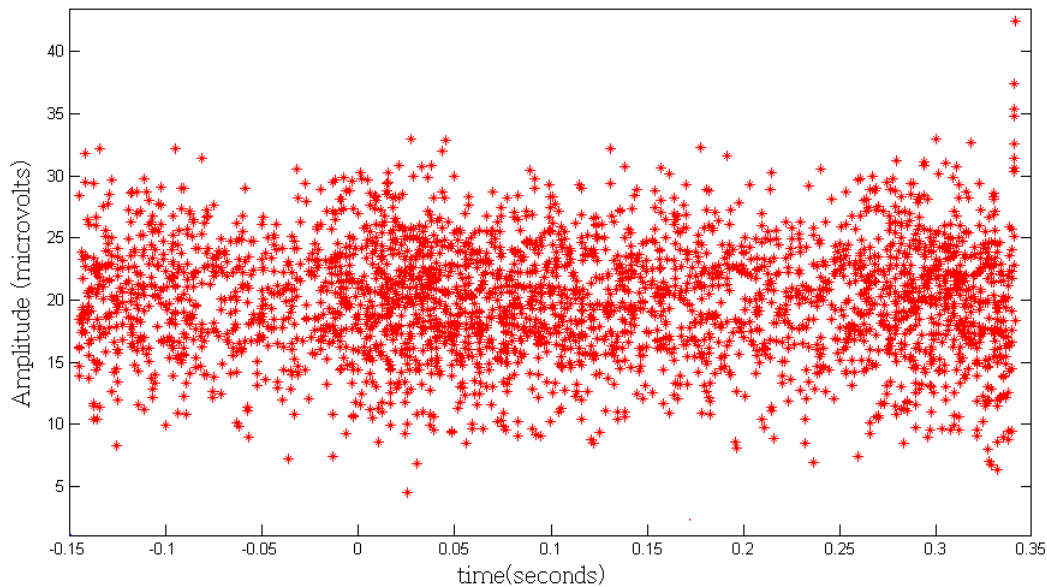


FIGURE 3.6: Distribution of artifacts over time and their corresponding amplitudes.

milliseconds in duration. This is irrespective of the amplitude. The implications of the combination of varying amplitudes and constant durations is positive for our matching later on. It means that the correlation between two artifacts of different amplitudes is not different than when the amplitudes were equal. If we would say that correlation between two signals measures the similarity between those signals, in this case, the artifacts are mutually almost the same. However the low sampling rate could introduce round-off errors. Therefore it cannot be fully certain the artifacts have a precise equal duration in reality.

Concluding, we can assume the general shape, the amplitude and duration in a single subject is invariant enough between artifacts and over time. The artifact could be variant between subjects. But this is not interesting yet, since it does not have implications for results of the data of a single subject. We further address this possibility in section 5.1.

3.4 Template matching using cross correlation techniques

Now we have generated a plausible general form of the artifact's signal, we can use it as a template for a template matching algorithm. This template matching is based on cross-correlation. Cross-correlation can be seen as a measure of similarity of two waveforms. It is also known as a sliding dot product or inner-product. It is commonly used to search a signal of longer duration for a shorter, known feature. In this thesis, the artifact based template. For discrete functions as in our case, the cross-correlation is defined as:

$$(f \star g)[n] \stackrel{\text{def}}{=} \sum_{m=-\infty}^{\infty} f^*[m] g[n + m].$$

The function has its maximum value when the two signals match, in other words when they're aligned so that they are shaped as similarly as possible. In figures 3.7, 3.8 and 3.9 the original signal, the cross-correlation with the template and the cross-correlation to the power of 4 can be seen respectively. Note that the second figure looks very similar to the first, as if we didn't make any progression. But this is wrong, the information represented in both figures is very different. The first contains information about signals from the brain, the second of the correlation of those signals with the template. It is peculiar though that they look alike, but it is important to make this step. With other data it might be the case that the original signal isn't similar to the cross-correlation sequence, and it is the cross-correlation information we're interested in.

As an important side note, we calculated the crosscorrelation between the derivative of the signal en the derivative of the template instead of using the original signals. We do this in order to use the structure of our data optimally. Because the artifacts are shaped as a faster increase and decrease of signal relative to the other data, the derivative of

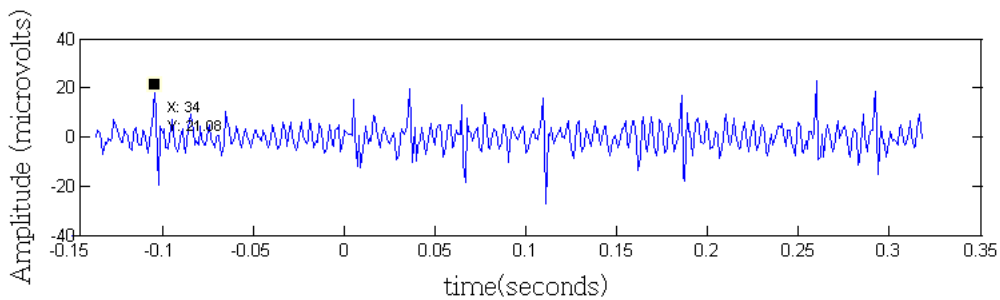


FIGURE 3.7: Original EEG signal with artifacts, on of which is labeled at time point 34.

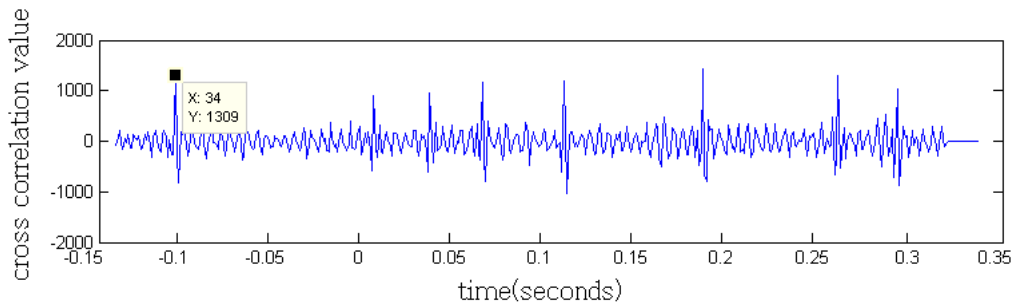


FIGURE 3.8: The cross-correlation between the template and the original signal. A peak of high correlation is labeled at time point 34.

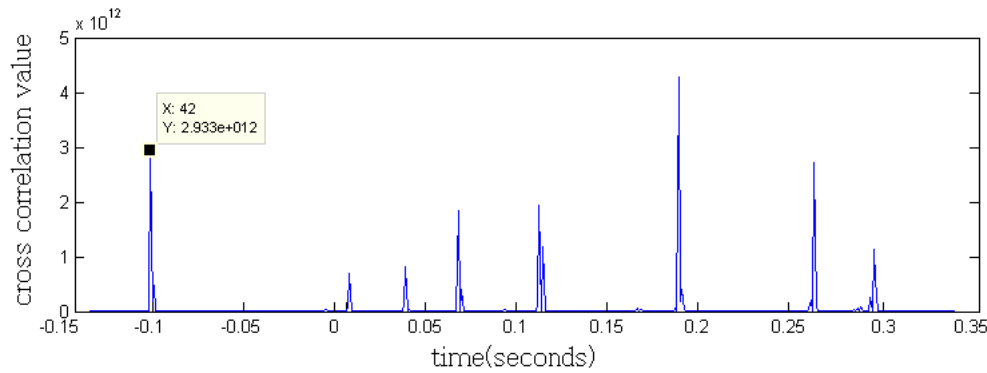


FIGURE 3.9: The cross-correlation between the template and the orinal data, to the power of 4. A peak is labeled at time point 42.

the signals hold more distinctive properties between artifacts and original signal. And because the more distinctive these two signals are, the better our matching algorithm can perform. You could argue against this step of processing because it might not apply on other data with other artifacts. However, our template method is based on visually detectable artifacts, which automatically implies greater amplitudes and thus better distinguishable derivatives. Therefore we didn't omitted this step.

Figure 3.9 shows us the crosscorrelation of the derivative to the power of 4. Other power settings can be used in order to obtain different distributions of peak heights, but this setting proved to be effective. Again, in other data this parameter could not be optimal. Choosing this value could be automated however by searching for values that result in lowest variance between peaks. We can see that at time point 42 there is the first high peak. That does not correspond to time point 34 of the artifact occurrence in our data. This is partially the result of the shifting property of the crosscorrelation function, and partially because the derivative of the signal has peaks not where the amplitude is highest, but where the increase is highest. This is however not a problem, because we can calculate backwards to obtain the position where the peak originally should be. Furthermore, the mutual distances between peaks is equal to those of in the original signal.

We now have obtained peaks of crosscorrelations. They point out where the artifacts occur, thus our next step is to select the time points of the peaks with a peak detection algorithm. However this is more complicated than it may sound because some peaks are the result of original data which are not artifacts, but do correlate high with our template. Thus simply selecting every peak as an artifact results in bad performance. Therefore we must differentiate between peaks: only a subset of the peaks which are higher than a certain threshold are to be accepted as an artifact match. The value of this threshold is calculated by taking the highest peaks, preferably of all 204 trial signals,

and calculating the average and standard deviation of those peaks. We then use this average and standard deviation to select peaks which differ a certain standard deviation factor of the average. Only these peaks are considered to be artifacts. Other peaks differ too much from the average highest crosscorrelations, and thus cannot be considered as artifacts. The influence on the performance of the threshold factor can be seen in figure 3.10. A value of 2 results in best performance, as we will discuss further in the next section.

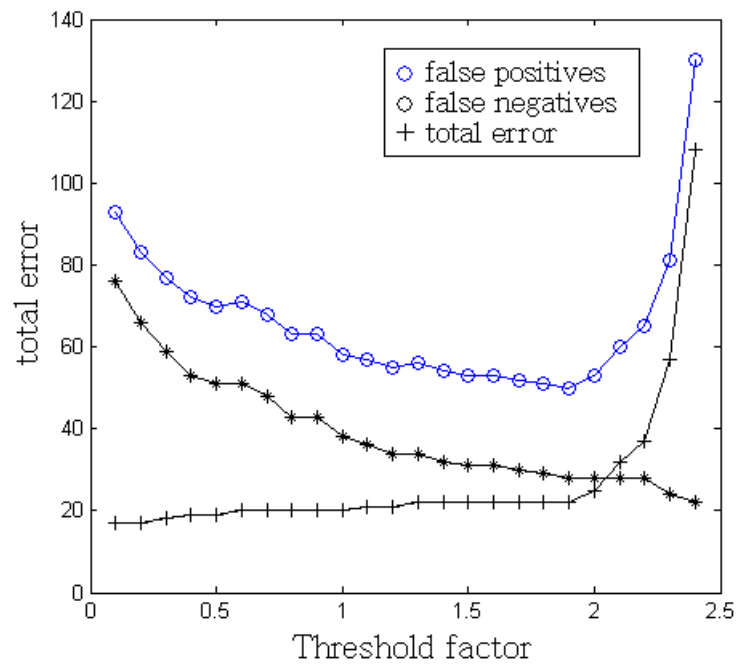


FIGURE 3.10: Influence on error by the threshold factor

3.4.1 Matching Performance

In the previous section we discussed the template matching algorithm using crosscorrelations. Applying this algorithm on our data gives us a list of matched positions in time where artifacts are detected by the algorithm. In the next section we will use these detected occurrences to remove the artifacts. But it is important to check the performance of our matching algorithm. This can be done by comparing the guessed positions with our previously attained ground truth. Since the ground truth exactly by human capabilities tells us when the artifacts occur, we can deduct when a matched artifact is a valid match, or is matched wrong. We can differentiate between two sorts of mismatches: false positives and false negatives. A false positive like oversensitivity, there is a mismatch that is erroneously positive when a situation is normal. A false negative is a match result that fails to reveal a situation.

When using our previously attained template to match on our data, we can reveal 99.19 percent of the 2856 occurring artifacts. Thus this gives us 23 false negatives. The number of false positives is equal to 26.

Visually a perfect result can be seen in figure 3.11. The ground truth is marked with an 'o' at the specific coordinates. A matched position is marked with a '*'. A lesser result can be seen in figure 3.12. As you can see there is a false positive marked only with an '*', and two false negatives marked with a 'o'.

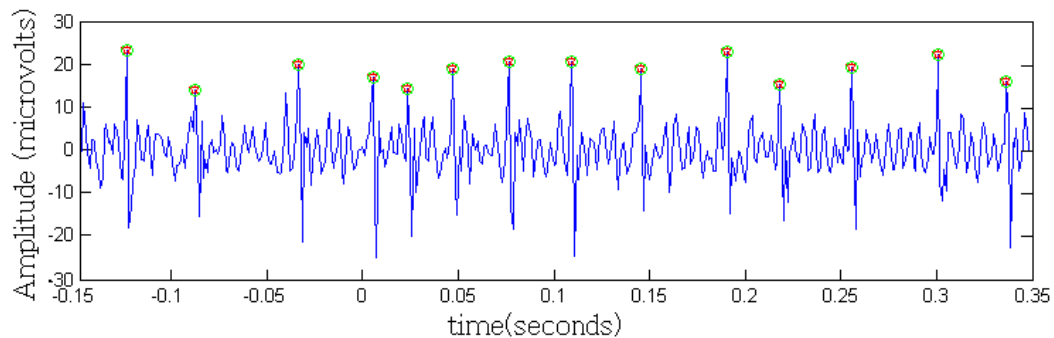


FIGURE 3.11: A trial where all artifacts are correctly recognized

3.4.1.1 Cross Validation

Because the previous performance was based solely on the template calculated from all 2856 artifact occurrences in the ground truth, it is proper to test whether or not the artifacts differ over time.

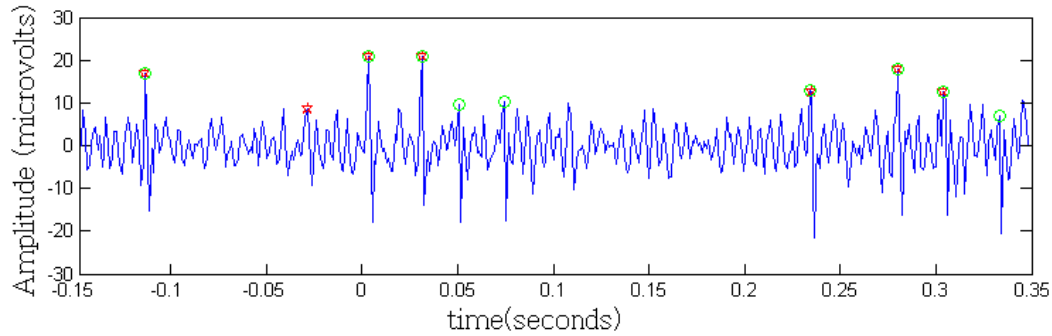


FIGURE 3.12: A trial which has one false positive, and two false negatives.

To test if the artifact changes over time and how that influences the matching performance, we can use a cross validation method. Cross-validation is a technique for assessing how an algorithm will generalize to an independent, or new and unseen, data set. It is mainly used in settings where the goal is prediction, and one wants to estimate how accurately a predictive model will perform in practice on new unseen data. The first step in cross-validation involves partitioning the data into complementary subsets. Next, one subset is used to train the model, and the other subsets are used to test the model on. Multiple rounds of cross-validation can be performed using different partitions.

In our case, we partitioned all signal trials into ten time bins. Each time bin now has a certain time span of data of all trails. Next we pick one time bin and use only this time bin to generate a template using the averaging technique. We then use the time based template to match artifacts in all the remaining time bins using our template matching method. This results in a set of matches of which we too can derive performance into false positives and false negatives. After this we pick an other time bin, and redo the previous steps. When we have done this for all time bins, we retrieve performances in the table beneath. From the table can be derived that performance does not fluctuate when we use different time bins for template generation. Also, the correlation between the different templates is very high: 0.998. Therefore we can safely conclude that there is no artifact variation over time and it has no influence on the overall performance.

Time bin:	1	2	3	4	5	6	7	8	9	10
false -	22	22	22	24	22	24	22	22	22	22
false +	29	28	29	29	30	29	28	29	29	29
total error -	51	50	51	53	52	53	50	51	51	51

3.4.1.2 Influence of sample size

What we implicitly already examined using the cross-validation above, is the how the number of samples we used for the generation of the template affects the performance. We used only ten percent of the samples with the cross-validation instead of all 2856. It is interesting to further investigate of how the number of samples affects the performance of the matching. It would be positive if we'd only needed a small data set to generate a template in order to match everything in a bigger data set. In figure 3.13 the number of random samples used to generate the template and the total error is plotted.

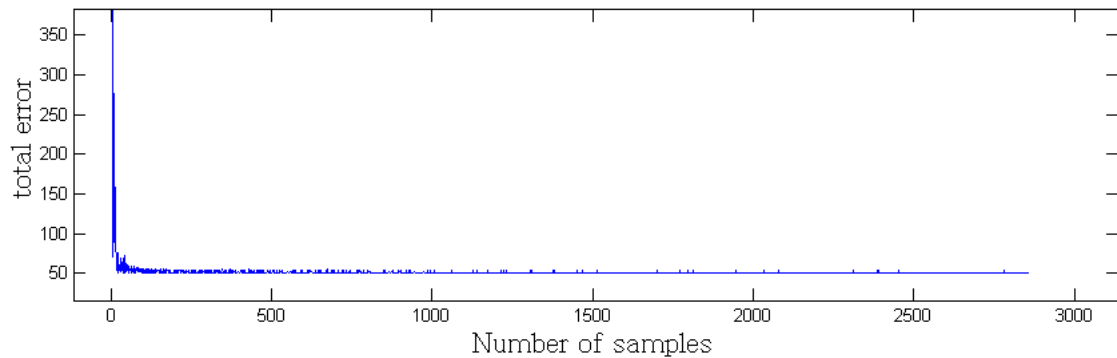


FIGURE 3.13: The number of samples used to generate the template plotted against total error.

The figure shows that around 70 samples are needed to achieve reasonable matching performance. The more samples, the lower the chance of producing error. Unfortunately the error does not drop to zero, but it does tend to stabilize. There is one downside, we need the template to be as accurate as possible for later removal methods. Therefore relying on a 70 sample template is not acceptable. However a solution to this would be using 70 random samples of artifacts to match on our data, and use the matched list to generate an accurate template. This way we have both advantages of picking only a few samples by hand, and retrieving an accurate average templates of the artifact.

Chapter 4

Artifact Removal

After successfully detecting the artifacts, our next wish is to remove them in a proper way. As discussed in the introduction, there are various approaches to remove artifacts. However, our methods differ from these in the sense they use more top down information. Because we have already obtained information about the occurrence of the artifacts in time, the general signal form and variance of the artifacts, it would be wise to use this information further for the removal. Assuming this information is correct, our main approach will therefore be focused on intelligent subtraction of the template of our artifact at the right point in time our matching algorithm found the artifacts. First the artifacts will be removed in a single channel. In a later section we will use more top-down information in order to also remove artifacts in all the channels. In the next sections we will discuss three approaches for the removal of artifacts.

4.1 Whole template subtraction

One way of artifact removal is the subtraction of the template as a whole from the slice. We know the artifacts are invariant in a lot of ways: they do not differ in length of time and all possess more or less the same shape. Therefore, the template represents the signal of the artifact rather well and we can use this to subtract it from slices of signal where the artifacts occur. There is one downside however, the amplitude does vary as we have seen in figure 3.6. Therefore we may have to come up with some more intelligent manners of subtraction.

4.2 Decorrelation and Amplitude adjusted template subtraction

An other way to remove a artifact is by subtracting a normalized version of the template multiplied by a certain factor. The normalisation makes sure the values of the template has a maximum of one. The multiplication factor indicates the strenght of the template in the signal. We can compute this strenght factor in two ways.

The first factor is based on a correlation. We calculate the correlation between the template and the artifact in our signal. The second is by retrieving the amplitude of the artifact in our signal.

This factor is then multiplied with the normalised template and is subtracted from the signal.

4.3 Raised Cosinus Filtering Prior to Subtraction

Before we actually subtract a signal retrieved by the methods above, we apply a raised cosinus filter over it. Raised cosinus filters are electronic filters frequently used for pulse-shaping in digital modulation due to its ability to minimise inter symbol interference. An example of the filter we used can be seen in figure 4.1. Basically the reason why we use it here is because we do not want to subtract the template directly from the signal. This could introduce new artifacts because there is no gentle transition between untouched data and the newly calculated slice. A raised cosinus filter smooths this transition which is favorable. Furthermore, the filter maintains spectral properties of what it filters, which again is favorable because we do not want the filtered signal to be transformed too much in both time and frequency content. The filter is applied by multiplying it with the template prior to subtraction.

4.4 Removal Results

It is hard to determine whether or not the removal was properly done because we never can know how the artifact free signal should look like. One way is to determine this is by simply looking at the cleared signal and intuitively see how the artifacts can no longer be seen by eye. In figure 4.2 a cleared signal can be seen atop of the original artifact

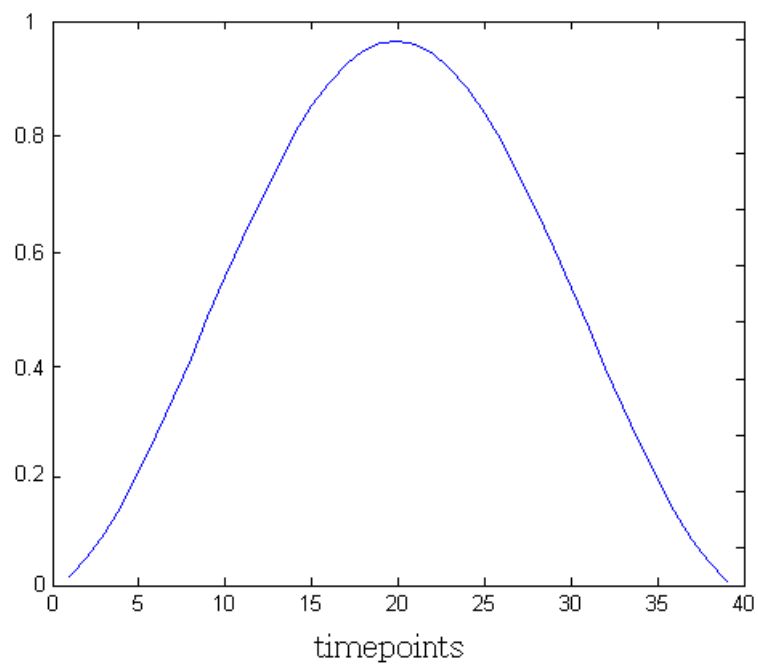


FIGURE 4.1: Raised cosine filter used to smooth transition between original and artifact removed signal.

polluted signal. The peaks are clearly gone. However, the human eye is not capable of analyzing the removal more specific than that. That's why we need some other smarter methods.

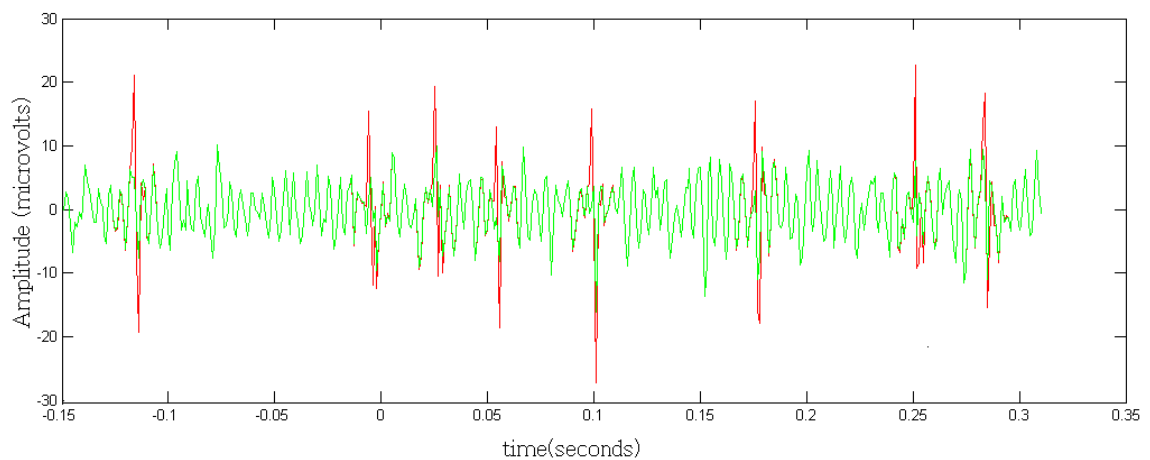


FIGURE 4.2: Artifact removed signal (green) plotted on top of original signal(red).

One of these methods is to use the ground truth to recalculate the average signal after the removal has taken place, around the places where the artifacts occurred. Before

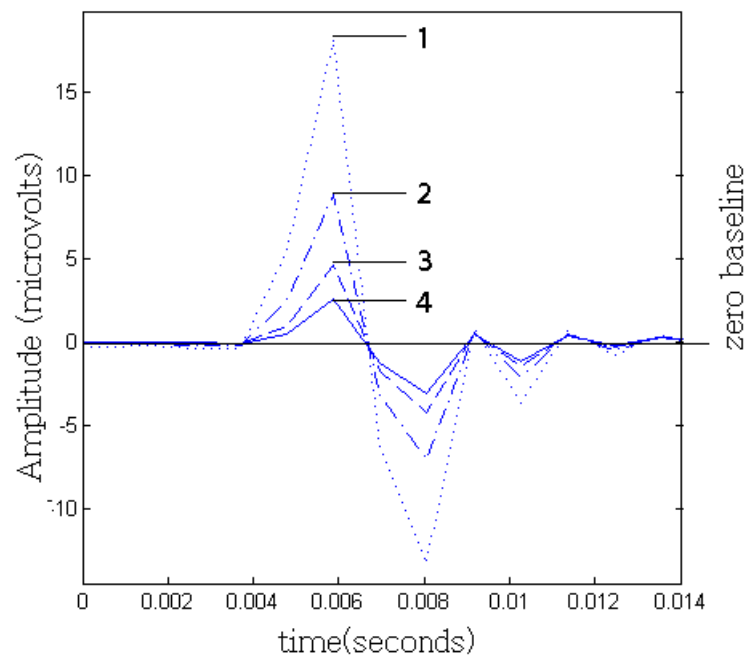


FIGURE 4.3: Average signals where previous artifacts occurred. (1)Before removal (2)Removal using correlation approach (3)Removal using whole template subtraction (4)Removal using amplitude adjusted template

removal this resulted in our template. And in theory, after removal, there should be a flat line resulting instead. Because all that should remain are random EEG and no artifacts. Therefore averaging should cancel the EEG out to a near flat line. This way we can compare the three ways of removal for effectiveness, these results are shown in figure 4.3. In that figure, signal(1) is our original average before any artifact removal has occurred. Signal(2) is the average result when we use the decorrelation approach. Signal(3) is the average result of when we subtract the whole template and signal(4) is the result when we apply amplitude adjusted subtraction.

From these results we can conclude that amplitude adjusted template removal results in the least average signal after artifact removal. This suggests that this approach for this type of artifact is the most effective. The remaining average signal after removal still is not a flat line, but this is not unexplainable. Our matching algorithm still produces a few errors in which some artifacts are missed. The average of those missed artifacts can be seen as a left over in the figure above. This also means the removal is more successful than the figure implies.

For this type of artifact, the amplitude adjusted removal thus produces the best result. For other types of artifacts, which for example are also variant in duration, this method

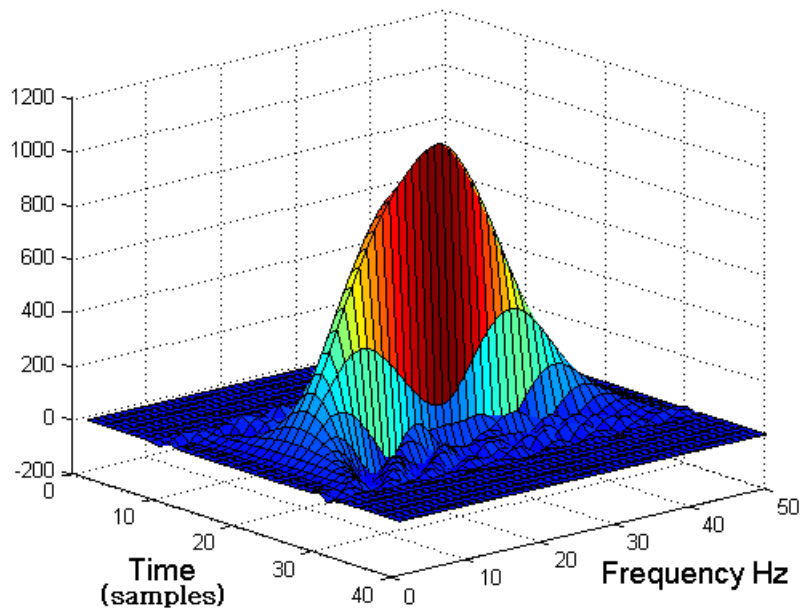


FIGURE 4.4:

could perform less. Those artifacts scale up in both directions, in contrast to ours, which alters only in the amplitude direction. In such cases, it might be more effective to use another kind removal based on linear scaling. This should not only take amplitude but also duration into account. This type of removal for other types of artifacts is however further research.

A second method to determine the properness of the removal is by a frequency analysis. In [18] the same data was analyzed. There it was suggested that the artifact is composed of high frequency signals. In figure 4.4 a frequency analysis is visualized in the form of a spectrogram. The template consists indeed of high frequencies, up to 45 Hz.

These high frequencies should be visible in a spectrogram frequency analysis of the original data as well. But this also implies they should be disappeared in the artifact removed data. In figure 4.5 a spectrogram is presented of trial 1 of the original data, before any artifact removal. High frequencies can be seen around the time points of artifact occurrences (time points: 34, 144, 204, 249, 326, 401, 434). In figure 4.6 a spectrogram of the same trial after removal of the artifacts can be seen. The high frequencies are now less powerful. This supports the hypothesis that the removal of artifacts was properly done.

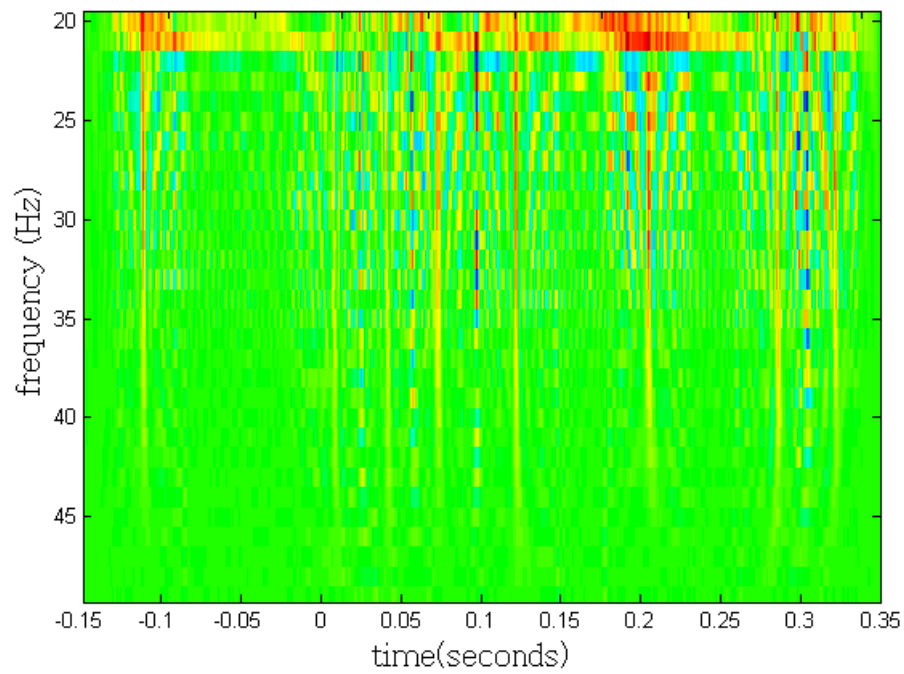


FIGURE 4.5: Spectrogram of trial 1 of the original data, before artifacts are removed. High frequencies are present around time points of artifacts occurrences (34, 144, 204, 249, 326, 401,434).

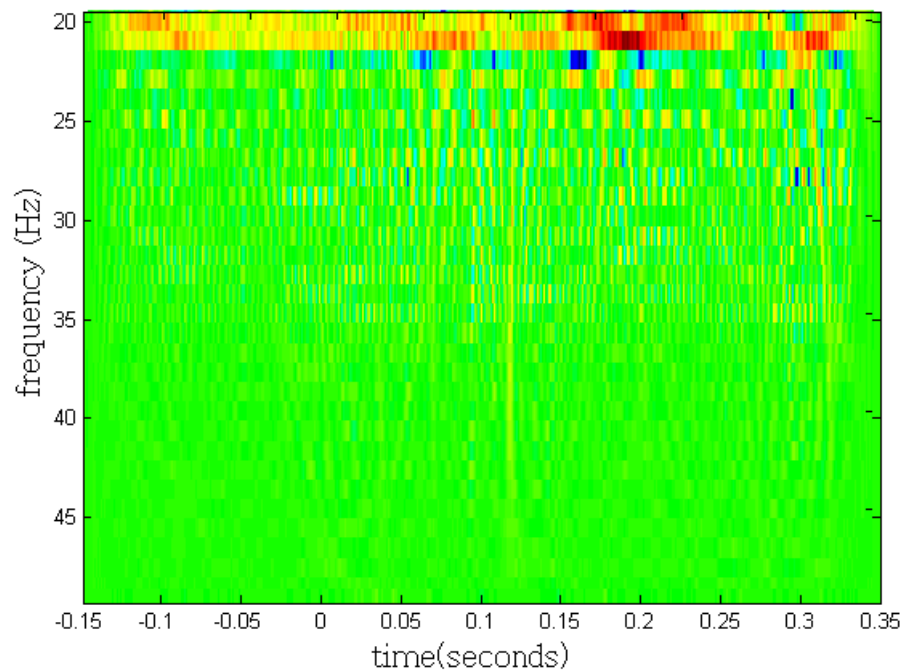


FIGURE 4.6: Spectrogram of trial 1 of the original data, after artifacts are removed. High frequencies are less powerfull.

4.5 Multi channel templates

Until now we have focused only on the detection and removal on a single electroencephalography channel. However, usually artifacts do not occur on a single channel alone, the signals propagate to all channels. In order to clean all the channels contaminated with this type of artifact, we could use the the same steps we used for the 'FPZ' channel. Running the algorithm on other channels doesn't produce any good results. The latter can be explained of the fact the artifact's signal decays rapidly. Because the further away we examine a channel from our presumed artifact source, the less detectable our artifact becomes. The channels don't seem to be contaminated as bad as we expected. Because the signal is weak in other channels, our template does not cross-correlate highly with the signals. There is relatively too much other signal that correlates with it, thus obscuring our matches.

Some other method has to be found since we cannot use the regular steps. A solution to this problem is the use the already found match results as top-down information for the other channels. We already obtained information of the occurrence of artifacts in time. What we do need are new templates for each individual channel. Therefore we can use the same averaging paradigm and generate an average of points in time around each artifact occurrence. Since electrical signals propagate immediate, not time lag in artifact occurrence in other channels is expected. When done, we obtain the set of templates for each channel visualized in figure 4.7.

It immediately becomes visible that the source of the artifacts does not propagate strongly to other channels. Propagation does occur when we zoom into channel like 'AF4', 'AFZ' and 'FZ'. However, the amplitude is relatively low and only occurring in these channels. It might as well be the effect of the preprocessing steps like down sampling because those methods take some average into account into other channels.

We could take every average as a template, and subtract these from their respective channel's signals in the same way for a single channel. Unfortunately the artifacts are not detectable in any other channel, which makes the removal not difficult to proceed with. These results also implies some new insights for the source of the artifacts which we will discuss in section 5.2.

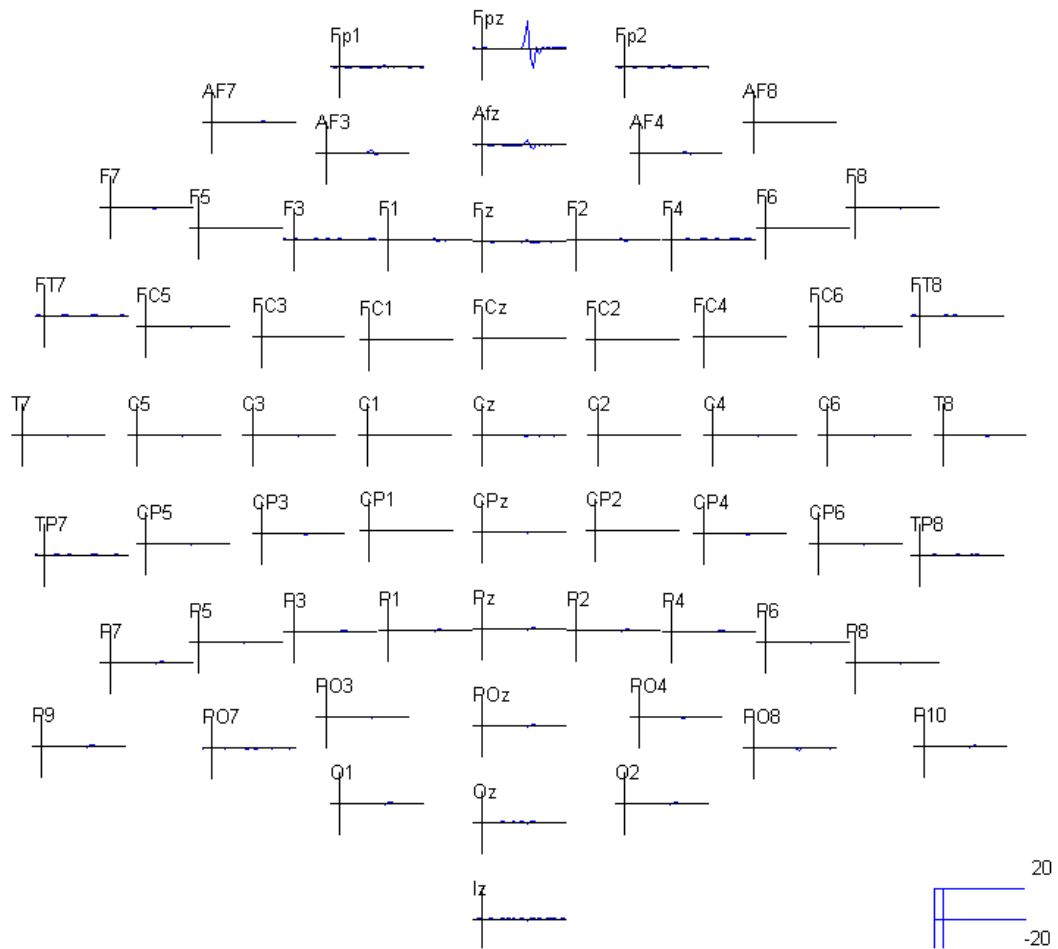


FIGURE 4.7:

Chapter 5

Discussion

5.1 Generalizability

In order to obtain more positive evidence for the usefulness and robustness of the method of template matching and removal, it is wise to look if it is applicable to other data with other artifacts. For that purpose we have used the secondary data set containing 747 trials with (down sampled to) 64 channels of 256Hz data. To restrict our efforts we only focus on a single virtual channel 'FP1' where a different type of artifact is most visible for the human eye to classify. Note this is a different channel then before, in channel 'FPZ' the artifact is not visible. In figure 5.1 a visualization of a single trial in 'FP1' can be seen.

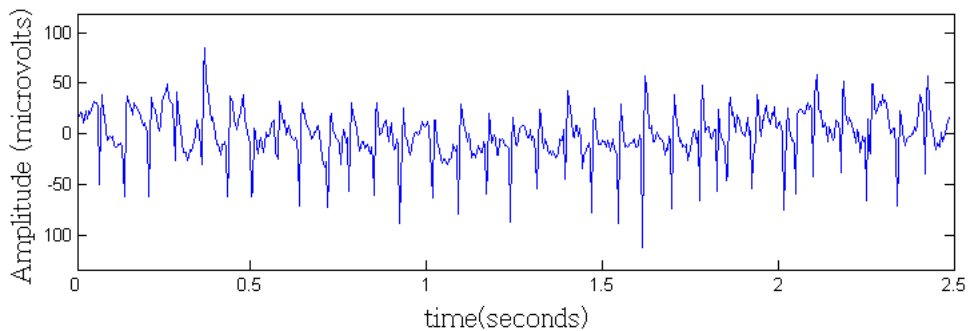


FIGURE 5.1: Trial of channel 'FP1' in the secondary dataset containing artifacts

At first sight the general form of the artifact seems different then in our previous data. To verify that properly we calculate a template of this type of artifact by manually selecting 77 time points in the data where the artifacts occur. For selecting them we

use a different kind of landmark for the artifact: we select the peak minimum of the artifact. These minima are representative properties of this type of artifact. Note we only select 77 artifacts by hand, since from figure 3.13 we concluded this is a value which should yield reasonable results for matching. For removal however, we should recreate the template from all matching results since that gives us more robust signal. In figure 5.2 the template of the artifact can be seen.

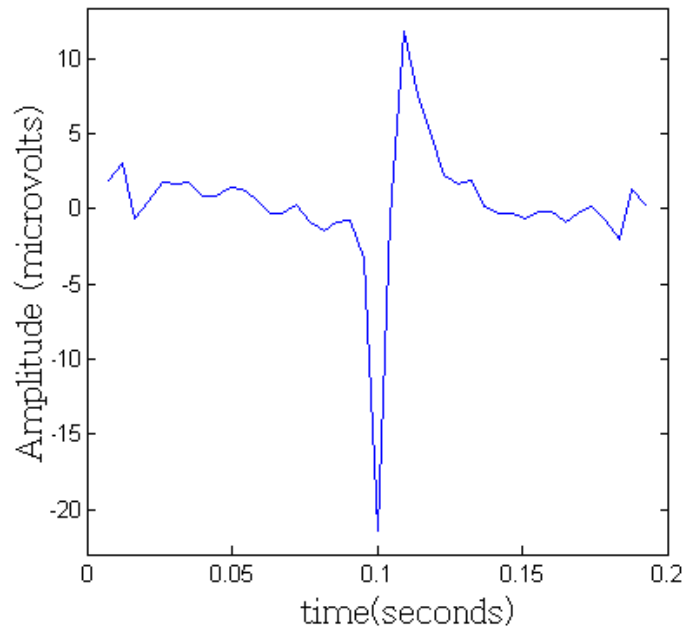


FIGURE 5.2: Template of secondary dataset, by averaging 77 samples of artifacts

5.1.1 Matching results

Using this template on our secondary data using our matching algorithm and equal parameters, this results in matching 74 of the 77 artifacts correctly and produces 3 false positives. The total error thus 6. This is relatively higher than the error on our primary data set, though still a reasonable performance for the first run. The performance could be increased by further fine tuning the threshold value or using more samples to generate the template since it might be possible that figure 3.13 is not representable for this type of artifact.

5.1.2 Remarkabilities

When we look at the general form, this second type of artifact seems to be the reverse of our template of our primary data set. It is possible that the different artifacts have the

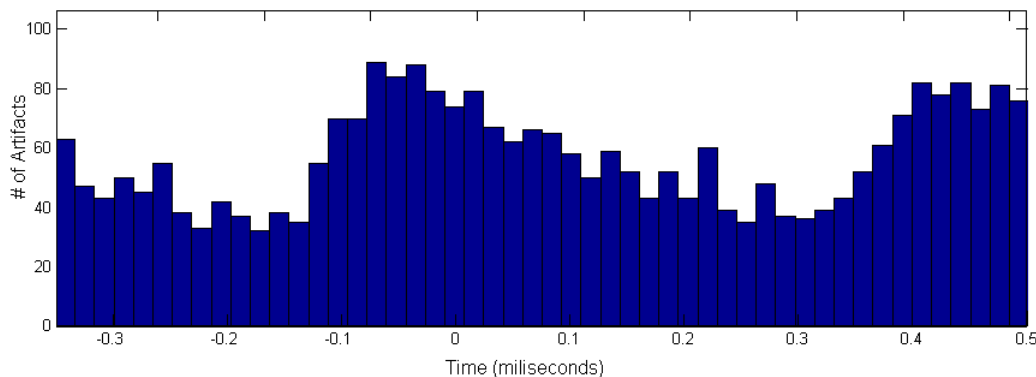


FIGURE 5.3: Distribution of artifacts over time and their corresponding amplitudes.

same source, but switched polarity due to a differing electrodes setup. The duration of both templates is equal of 55 ms, which feeds this thought even more. Also the channels where both sorts of artifacts occur are neighboring, as can be seen in figure 4.7. This makes it all the more likely that they have the same source. However the timing and frequency of occurrence is very different.

5.2 Artifact Source

The source of the artifacts in our primary data set is not known, but we do have evidence where they might be from. In [18] it is suggested that the source of these artifacts is of micro saccadic eye movement. Saccades are defined as small involuntary eye movements. They are considered micro-saccades when the movement of the eye is less than 0.5° . There are many types of saccades, differing in amplitude, duration and waveform [3]. The main evidence for this suggestion is that the distribution of the artifacts over time correlates with assumed attention of the subject. In [16] experiments were done where saccade rates for all subjects dropped around 100-150ms following stimulus onset. The frequency rebounded to a peak between 200-300ms after stimulus onset. They concluded that saccadic inhibition occurs shortly after a new stimulus, after which it then increases. Also it was suggested this effect may generalize to other sorts of stimuli.

In figure 5.3 the distribution in the ground truth of the number of artifacts over time over all trials can be seen. At time 0 and 0.5 the subject heard a metronome tick of approximately 88dB(A) and was instructed to imagine an accented beat previously heard before. This task was repeated every 500 milliseconds, thus in our figure are two stimuli presented and two tasks performed during the total time span. According to

[16], if the artifacts are indeed saccades, an artifact drop can be expected starting at 0 seconds and an increase peaking after 200-300 milliseconds after stimulus onset. This is indeed the case, supporting the hypothesis the artifacts are of saccadic origin.

There is however evidence against this hypothesis. One that could falsify the saccadic origin directly. In channel 'CP5' of our secondary data set, also artifacts occur. Different than the artifact in channel 'FP1' of our secondary data set, but alike the artifact in our primary data set. A single trial is plotted in figure 5.4. A closer look of a occurring artifact can be seen in figure 5.5.

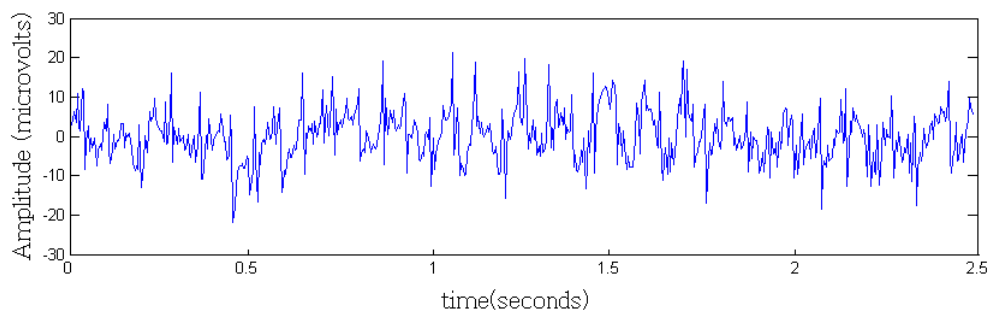


FIGURE 5.4: A single trial of channel 'CP5' of our secondary data set where artifacts seem to occur.

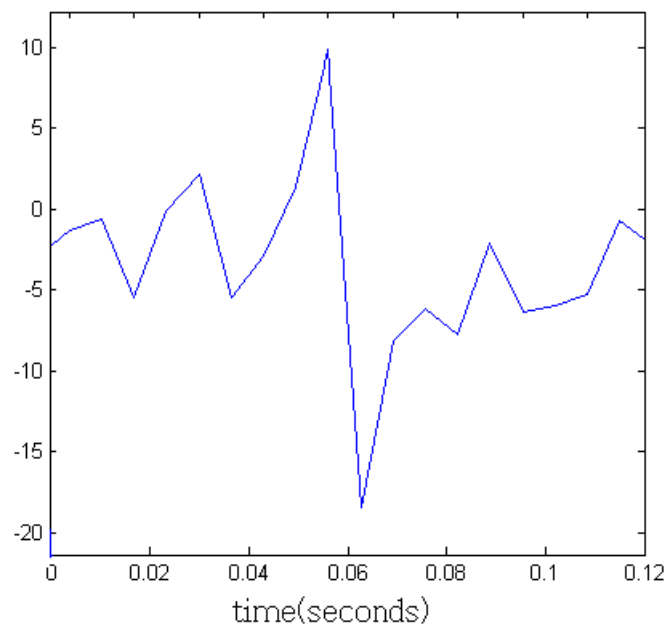


FIGURE 5.5: An artifact in channel 'CP5' in our secondary data set.

This artifact shows a slight resemblance to our presumed saccadic artifact. However the place of occurrence on the scalp, as can be seen in figure ??, is localized far away of 'FPZ' and 'FP1'. Since artifact in those channels only slightly seemed to propagate their signal to neighboring channels, it is unlikely that in channel 'CP5' this artifact has

a saccadic source. Due to the resemblance of these artifacts but differing locations, this also implies the artifact in channel 'FPZ' of our primary data set is more unlikely to be saccadic origin. But although the signal form shows resemblance, the duration of this artifact is approximately 20 milliseconds. That is a factor 5 in respect to the artifact in our primary dataset.

There is a feasible explanation however. In [7] there is a case study presented where spike like artifacts were seen in the EEG when only one of the two experimenters was working with the subject. In different trials, one of the two experimenters clapped hands in order to investigate the responsiveness to acoustic stimuli. The spike like artifact in the EEG seemed to vary in configuration and channels depending on where one of the experimenter was standing. After eliminating all possibilities, they finally pinpointed the source of the artifact. One experimenter wear a cotton t-shirt underneath the synthetic lab coat. Due to the clapping, the fabrics moved, producing electrostatic discharges which were picked up by the EEG. In the experiments for collecting our data, sound-speakers were used to introduce a metronome tick for our subjects. It is not unlikely that the movement of the inner cones of those speakers, produce electrostatic discharges as well since they partially made of cellulose and synthetic material. This could also explain the distribution of artifacts over time of figure 5.3. When a speaker produces a metronome tick, the inner cone increasingly vibrates, resulting in possibly more electrostatic discharges. However, further evidence for this phenomem should be gathered in order to conclude the source.

Chapter 6

Conclusion

In conclusion we answer the six main questions this thesis started with.

Quantifying the artifact's signal from the EEG proved successful. Two different ways were implemented which produced equal results, supporting each others findings. In order to generate a successful template for matching purposes, around 70 samples suffices. After that, matching results can be used to generate an even more accurate template. A downside is that for the template these initial samples have to be handpicked. With a simple graphical user interface however, efforts of this task can be reduced to mere minutes. The quantification can be considered as proper, since there was only variance of the artifacts in amplitude. Duration and general form over time do not vary. Also cross-validation showed that the artifact over time stayed equal since high correlations were found and no great influence on performance was detected.

Automatically finding artifacts in time proved successful. 99.19 percent of the 2856 artifacts in the primary data set were found, thus missed 23 artifacts. An almost equal amount of 26 time points were marked as an artifact when they were not. Also with cross-validation techniques it was proved that the algorithm can generalize to a new unseen data set of the same kind.

Removal of the artifacts was investigated with three different methods. Amplitude adjusted template removal resulted in best removal performance, tested by averaging point in time where artifacts occurred before removal. Also the frequency analysis showed that the high frequencies of the artifacts disappeared, pointing out proper removal.

The template matching method can be considered useful for other types of artifacts. An differing artifact type with the same invariant and variant properties in a secondary data set was used to generate a template as well. The matching algorithm performed relatively worse than in our primary data set, but still within reasonable bounds. Further research

has to be done in order to find if the performance can be maximized for this type of artifact. For other types of artifacts, which could vary in duration as well, the algorithm should be adjusted. A scaled template removal approach therefore is suggested.

The source of the artifacts was presumably at first of a saccadic source. Evidence of the distribution over time supported this thought. However, the similar artifacts were found in other channels. However the duration of these artifacts are far greater. Although the general form looks the same, they might have a different source. It is suggested that the spikes are the result of electrical discharges near the EEG setup, however not much evidence supports this thought yet. The source remains unknown.

The method in contrast to other known techniques is also discussed. The method can easily be automated for a single subject for a single experiment when in a train fase a few minutes of human effort is performed. However, it is not yet known how this method generalizes between sessions. The template may alter significantly when the EEG cap is removed and placed back. Also this method is not yet fully automated, in contrast to some other techniques. After further research, it may be concluded that certain types of artifacts are invariant between subjects and between sessions. In that case the template could be hard coded and act as an automated artifact filter. The technique is especially robust for non variant artifacts. For the detection variance is allowed. However, for proper removal, variance factors should be taken into account. These factors can be handled by modeling the artifact after knowledge about variance in amplitude, duration and general form is known.

Bibliography

- [1] T. Sejnowski A. Delorme and S. Makeig. Enhanced detection of artifacts in eeg data using higher-order statistics and independent component analysis. *NeuroImage*, 34:1443–1449, 2007.
- [2] J.Pripfl G.Dorffner A. Flexer, H.Bauer. Using ica for removal of ocular artifacts in eeg recorded from blind subjects. *Unpublished*, May 2005.
- [3] R.V. Abadi and E.Gowen. Characteristics of saccadic intrusions. *Vision Research*, 44:2675–2690, 2004.
- [4] R. Brunelli. *Template Matching Techniques in Computer Vision: Theory and Practice*. 2009.
- [5] R.J. Croft and R.J. Barry. Removal of ocular artifacts from the eeg: a review. *Clinical Neuropsychology*, 30, 2000.
- [6] W. Dement and N. Kleitman. Cyclic variations in eeg during sleep and their relation to eye movements, body motility and dreaming. *Electroencephalography and Clinical Neurophysiology*, 9:673–690, 1957.
- [7] J. Dressnandt and H. Brunner. Spike like eeg artefacts due to electrostatic discharge. *Clinical Neurophysiology*, 120(155), 2009.
- [8] A.E. Eiben and J.E. Smith. *Introduction to Evolutionary Computing*. 2007.
- [9] A. Hyvrinen and E. Oja. Independent component analysis: Algorithms and applications. *Neural Networks*, 13:411–430, 2000.
- [10] M.N. Verbaten J.C. Woestenburg and J.L. Slangen. The removal of the eye-movement artifact from the eeg by regression analysis in the frequency domain. *Biological Physiology*, 16:127–147, 1983.
- [11] F. Lue J.L. Witton and H. Moldofky. A spectral method for removing eye-movement artifacts from the eeg. *Electroencephalography and Clinical Neurophysiology*, 44:735–741, 1978.

-
- [12] R.M. Frank C.Davey J.Dien A.D. Malony K.A. Glass, G.A. Frishkoff and D.M. Tucker. A framework for evaluating ica methods of artifact removal from multi-channel eeg. *Unpublished*.
- [13] R. Strungaru M. Ungureanu, C. Bigan and V. Lazarescu. Independent component analysis applied in biomedical signal processing. *Measurement science review*, 4:411–430, 2004.
- [14] E. Urrestarazu P. LeVan and J. Gotman. A system for automatic artifact removal in ictal scalp eeg based on independent component analysis and bayesian classification. *Clinical Neurophysiology*, 117(4):912–927, 2006.
- [15] S. Clos S. Gimenez M.J. Barbanoj S. Romero, M.A. Mananas. Reduction of eeg artifacts by ica in different sleep stages. *Engineering in Medicine and Biology Society*, (3):2675–2678, December 2003.
- [16] A.S. Keren I. Nelken S. Yuval-Greenberg, O. Tomer and L.Y.Deouell. Transient induced gamma-band response in eeg as a manifestation of miniature saccades. *Neuron*, 58:429–441, May 2008.
- [17] P. M. Manoj Kumar K. Shivakumar V. Krishnaveni, S. Jayaraman and K. Ramadoss. Comparison of independent component analysis algorithms for removal of ocular artifacts from electroencephalogram. *Measurement Science Review*, 5, 2005.
- [18] R.J. Vlek, R.S. Schaefer, C.C.A.M Gielen, J.D.R. Farquar, and P Desain. Subjective rhythmization as a paradigm for bci. *Unpublished*, October 2008.

```

%-----
% Script: bootData
%-----
% This specific script loads data from one subject, one condition
% and molds it into a fieldtrip ready form.
% It also preprocesses the data.
%-----
% Output: preproc_channels{i}, where i equals the number of channels.
% Every channel contains a matrix of trials x timepoints of data.
%-----
% Subsection1 : Loading Data
%-----
% Settings
datachanged = 0;
number_of_channels = 64;

% Set paths for toolboxes, raw data and output dir
addpath(genpath('~\source\BCI code\toolboxes\numerical_tools\'));
addpath(genpath('~\source\BCI code\toolboxes\classification\'));
addpath(genpath('~\source\BCI code\toolboxes\utilities\'));
addpath(genpath('~\source\BCI code\toolboxes\plotting\'));
addpath(genpath('~\source\BCI code\Utilities\ElectrodePos\caps\'));
addpath(genpath('~\source\BCI code\external_toolboxes\eeglab\'));
addpath(genpath('~\source\BCI code\external_toolboxes\fieldtrip\'));
name so don't switch order!
addpath(genpath('~\source\BCI code\toolboxes\signal_processing\'));
dirin = '\Network\Servers\mmmserver.nici.ru.nl\Volumes\Xserver_RAID\
Users\ruudbarth\Data\Raw\';
dirout = '\Network\Servers\mmmserver.nici.ru.nl\Volumes\Xserver_RAID\
Users\ruudbarth\Data\Results\';
addpath(genpath(strrep(which('setpipepaths.m'),'setpipepaths.m','')));

%Set subject-value to 1, represents first datafile in raw Data folder
PPi = 1;
CND = conditionlist;
PP = subjectlist;
cA = 1;
task = 1;

cfg = [];
cfg.dirin = dirin;
cfg.dirout = dirout;
cfg.datachanged = datachanged;
cfg.subject = strrep(PP{PPi}.datadir, '/', '');
cfg.inputfile = [cfg.dirin cfg.subject '/' cfg.subject];

cfg.condA = CND.list{cA};
cfg.condA.task = {CND.tasknames{task}};
cfg.condA.abbr = CND.abbrs{task};
cfg.condA.str = [cfg.condA.abbr 'p'
num2str(cfg.condA.period) 'b'
num2str(cfg.condA.beat)];
cfg.condA.sessions = 1:PP{PPi}.nsessions;

```



```
cfg.condA.reject_artifacts = 1;
cfg.condA.reject_falseanswer = 1;
cfg.condA.reject_firstperiod = 1;
cfg.condA.reject_badtrial = 1;
dataA = [];

dataA = getData(cfg, cfg.condA);

%-----
% Subsection1 : Preprocessing Data
%-----
% Band pass filtering (4-48 Hz)
hpcfg = [];
%hpcfg.lnfilter = 'yes';
hpcfg.hpfilter = 'yes'; % highpass filter
hpcfg.hpfiltord = 6; % highpass filter order
hpcfg.hpfiltype = 'but'; % digital filter type, 'but' or 'fir'
hpcfg.hpfreq = 10; % highpass frequency in Hz
hpcfg.detrend = 'yes';
hpcfg.reref = 'yes';
hpcfg.refchannel = {'all'};
[dataA] = preprocessing(hpcfg, dataA);

% Put it in format: channel x time x trials
dat = cat(3, dataA.trial{:});

for(i=1:number_of_channels),
preproc_channels{i} = shiftdim(dat(i,:,:));
end
```

```

function [ground_truth] = generate_ground_truth(channel, raw_artifact_positions)
%-----
% Function: generate_ground_truth
%-----
% This script determines the exact time position of saccade occurrence by
% refining the rough estimated artifact positions, given
% - original signal data
% - rough estimated positions of saccade peaks, retrieved with the
%   manual_artifact_gui.m
%
% In order to retrieve a reasonable form of Ground Truth, we must determine
% the occurrence of saccades in time. Because the estimated positions are
% only rough estimates, it is useful to determine a more exact form of
% saccade occurrence. This script does this by finding a local maximum
% around the estimated coordinates since this is our landmark of our
% specific artifact.
%
%-----
% Requires: - raw_artifact_positions guessed imprecise x-y-channel positions
%           of artifact occurrence.
%           - channel, signals of a single channel
% Output:   - ground_truth
%-----

% defines how much of the signal around the estimated saccade position
% should be considered to find a maximum in.
search_window = 5;

% For every found rough saccade, determine local maximum.
for i=1:length(raw_artifact_positions);

    % Retrieve raw signal where saccades were estimated.
    % Retrieve rough time position.
    % Calculate range where maximum should be.
    % Create a searchbin which contains piece of signal.
    % Retrieve maximum
    signal = channel(:,raw_artifact_positions(3,i));
    xrough = raw_artifact_positions(1,i);
    xrange = [floor(xrough-search_window), floor(xrough+search_window)];
    xmax = 0;
    ymax = 0;

    if xrange(1) <= 0,
        searchbin = signal(1:xrange(2));
        for k=1:length(searchbin),
            if(searchbin(k)>ymax)
                ymax = searchbin(k);
                xmax = k+(search_window)-length(searchbin)+(floor(xrough));
            end
        end
    else
        if xrange(2) > length(channel),
            searchbin = signal(xrange(1):length(channel));
            for k=1:length(searchbin),

```

```
        if(searchbin(k)>ymax)
            ymax = searchbin(k);
            xmax = k+floor(xrough)-(search_window+1);
        end
    end
else
    searchbin = signal(xrange(1):xrange(2));
    for k=1:length(searchbin),
        if(searchbin(k)>ymax)
            ymax = searchbin(k);
            xmax = k+floor(xrough)-search_window-1;
        end
    end
end
ground_truth(i,1) = xmax;
ground_truth(i,2) = ymax;
ground_truth(i,3) = j;
end
```

```

function [template_avg] = generate_template_avg_single(channel, truth)
%-----
% Function: generate_template_avg
%-----
% Template generation using ground truth positions and averaging.
%-----
% The ground truth positions of the saccades are aligned around their
% peaks. On every side of the peak, a certain span of signal is extracted.
% All these extracted signals are summed. After summation the summed
% signals are averaged by the number of saccades used in the ground truth.
%
% Function parameters:
% - channel: original signals of one channel
% - ground_truth: true positions of saccade peak occurrence
%
% Function result:
% - a template
%-----

% The desired width of the template is defined here.
timewindow = 20;
template_avg = zeros(timewindow*2,1);

for i=1:length(truth),

    signal = channel(:,truth(i,3));

    xpeak = truth(i,1);
    xrange = [(xpeak-timewindow),(xpeak+timewindow)];

    for j=1:(2*timewindow),

        if( (xrange(2)<=length(signal)) && (xrange(1)>0)),
            template_avg(j) = template_avg(j) + signal(xpeak-timewindow+j);
        end
    end
end

template_avg = template_avg/length(truth);

```

```

function [multi_template] = generate_template_avg_all_channels
(channels, ground_truth, number_of_channels)
%-----
% Script: generate_template_avg_all_channels
%-----
% Template generation for other channels using template matched retrieved
% positions.
%-----
% Function uses:
% - original signals: channels{k}, where k is the number of channels.
% - ground_truth      : true positions of saccade peak occurrence. Note that a
%                       list of template matched positions can also be used,
%                       though less accurate.
%
% Function creates:
% - multi_template{k}, template for each k channels
%-----

% The desired width of the template is defined here.
timewindow =20;

for k=1:number_of_channels,

    multi_template{k} = zeros(timewindow*2,1);

    for i=1:length(ground_truth),

        signal = channels{k}(:,ground_truth(i,3));

        xpeak   = ground_truth(i,1);
        xrange  = [(xpeak-timewindow),(xpeak+timewindow)];

        for j=1:(2*timewindow),

            if( (xrange(2)<=length(signal)) && (xrange(1)>0)),
                multi_template{k}(j) = multi_template{k}(j)
                    + signal(xpeak-timewindow+j);
            end
        end
    end

    multi_template{k} = multi_template{k}/length(ground_truth);

end

```

```

for j=1:length(CC),

    if CC(j)>(M-thresholdfac*S),

        [Y X] = max(CC(j-8:j+8));

        matches_(counter,1) = X + j - 16;
        matches_(counter,2) = signal_original(X + j - 16);
        matches_(counter,3) = i;
        counter              = counter+1;

    end

end

matches_ = unique(matches_, 'rows');

% Here it finds a local maximum to align the estimate of the saccade.
% Note that this is artifact specific. To generalize to other
% artifacts, you may need to change this function to find a landmark
% of that specific artifact. In this case the landmark is a high
% indistiguisable peak.
signal_shifted = [0;0;0;0;0;signal_original;0;0;0;0;0];
for k=1:length(matches_),

    slice = signal_shifted((matches_(k)-5)+5:(matches_(k)+5)+5);
    [Y X] = max(slice);

    if(floor(matches_(k)+X-6)>460),
        matches_(k,1) = 460;
        matches_(k,2) = signal_original(matches_(k,1));
        matches_(k,3) = i;

    else
        matches_(k,1) = floor(matches_(k)+X-6);
        matches_(k,2) = signal_original(matches_(k,1));
        matches_(k,3) = i;
    end

end

matches = [matches ; matches_];
matches_ = [];
counter = 1;

end

matches = unique(matches, 'rows');
if(length(matches) > 0),
    matches = sortrows(matches,3);
end

```

```
function performancesCC = crossvalidation(channel, ground_truth, NofSets)

performancesCC = [];

setLength = length(channel)/NofSets;

for i=0:NofSets,

    test_truth = [];

    bound_left = (i*setLength)+1
    bound_right = bound_left+setLength

    for j=1:length(ground_truth),

        if(ground_truth(j,1)>bound_left && ground_truth(j,1)<bound_right)

            test_truth = [ test_truth ; ground_truth(j,:)];

        end

    end

    test_truth

    template_subset = generate_template_avg_single(channel, test_truth);
    figure;
    plot(template_subset);

    matches = match_template_CC(template_subset, channel, 1.8);

    [performance, false_positives, false_negatives] = calculate_performance
        (ground_truth, matches);

    performancesCC{i+1} = [ false_positives
        false_negatives false_positives+false_negatives];

end
```

```

function [channel_clean c1 c2] = decorrelate_single(matches, template, channel)
%-----
% Function: decorrelate_single
%-----
% This function decorrelates a single channel with the occurring artifacts.
% For this decorrelation a template and template-matched positions are used.
%-----
% Function uses: - matches, template matched positions of artifact occurrences.
%               - template.
%               - channel, the channel with signals to decorrelate
%
% Function output: - channel_clean, clean decorrelated channel.
%-----

template = template-mean(template); %zero-mean the template
counter = 1;
channel_clean = channel;

corrs = 0;
corrs_after = 0;

for i=1:length(matches),

    side_left = false;
    side_right = false;

    signal = channel(:,matches(counter,3));

    left = matches(counter,1)-(length(template)/2);
    right = matches(counter,1)+(length(template)/2);

    if(left<0),
        template_ = template(abs(left)+1:end);
        slice = signal(1:length(template_));
        side_left = true;
    else
        if(right>length(signal)),
            template_ = template(1: length(template)-(right-length(signal)));
            slice = signal((length(signal)-length(template_))+1:end);
            side_right = true;
        else
            template_ = template;
            slice = signal( matches(counter,1)-length(template_)/2+1 :
                            matches(counter,1)+length(template_)/2);
        end
    end

    sliceor = slice;
    correlation = corr(template_, slice);
    corrs = corrs + correlation;
    transformed_slice = slice - max(slice) * (template_/max(template_));
    correlation = corr(template_, transformed_slice);
    corrs_after = corrs_after + correlation;

```

```
        if(side_left == true),
channel_clean(1:length(transformed_slice),matches(counter,3)) = transformed_slice;
        else
            if(side_right == true),
channel_clean(length(signal)-length(transformed_slice)+1:end,matches(counter,3))
= transformed_slice;
            else
channel_clean(matches(counter,1)-(length(transformed_slice)/2)+1:matches(counter,1)
+(length(transformed_slice)/2),matches(counter,3)) = transformed_slice;
            end
        end
        counter = counter+1;
end

disp('Corr before:');
c1 = corrs/i
disp('Corr after:');
c2 = corrs_after/i
disp('=====');
```

```
function [clean_channels] = decorrelate_all(matches, multi_template, channels)

clean_channels = [];

for i=1:length(channels),
    clean_channel = decorrelate_single(matches, multi_template{i}, channels{i});
    clean_channels{i} = clean_channel;
end
```
