Wavelet-based Preprocessing of Electroencephalographic and Electromyographic Signals for Speech Recognition

Michael Wand

June 19, 2007

Studienarbeit

Interactive Systems Laboratories
Carnegie Mellon University, Pittsburgh, PA, USA
Institut für Theoretische Informatik
Universität Karlsruhe (TH), Karlsruhe, Germany
Advisors: Dr. T. Schultz, Prof. Dr. rer. nat. A. Waibel

Abstract

Electroencephalography (EEG)-based communication for situations in which normal speech may not be uttered has been investigated several times. Recently, experiments showed that besides giving simple commands i. e. to a computer, the recognition of actual unspoken words may also be feasible.

Wavelet-based signal processing has been employed increasingly often for all kinds of signals. The development of the Fast Wavelet Transform (FWT) as a counterpart to the Fast Fourier Transform has made this preprocessing usable with all kinds of environments, computing hardware and time constraints.

This research is dedicated to investigate the potential of the Wavelet Transform for EEG signal preprocessing for the recognition of words. I could show that while the FWT does not seem suitable for this task, the Double-Tree Complex Wavelet Transform, a relatively easy variation of the FWT, improves word recognition accuracy considerably.

The second part of this work deals with the question whether Wavelet methods are suitable for speech recognition via electromyographic signals generated by the articulatory muscles. In these experiments, a phoneme-based speech recognition was attempted. Again, I could show that the Wavelet Transform may be successfully applied in this field and outperforms pure spectral features.

Acknowledgements

I would like to thank my advisor Dr. Tanja Schultz for supporting my work with her advice and for giving me the opportunity to work on this intriguing field of research both in Karlsruhe and in Pittsburgh.

My work on EEG speech recognition would not have been possible without the foundations laid by Marek Wester and Jan Calliess, who not only investigated this topic before me, but also created the signal recordings I used. Similarly, my EMG work is based on the previous work by Szu-Chen Jou.

Jan Calliess and Szu-Chen Jou initiated me into the secrets of the JANUS Speech Recognition Toolkit and often helped me out when problems occured. Special thanks to them!

This research was partly funded by the Baden-Württemberg-Stipendium (InterACT) granted by the Landesstiftung Baden-Württemberg.

German Summary

Diese Studienarbeit befasst sich mit der Anwendung verschiedener Varianten der Wavelet-Transformation (WT) auf elektroenzephalographische (EEG) und elektromyographische (EMG) Signale zum Zwecke der Spracherkennung. In der EEG-Erkennung habe ich Aufnahmen sowohl von gesprochener als auch von nur "gedachter" Sprache zugrundegelegt. Im Fall der EMG-Erkennung beruhte meine Arbeit auf einem aus 500 Äußerungen bestehenden Datensatz verbal ausgesprochener Sprache.

Der Hauptteil dieser Arbeit betrifft die Erkennung von ganzen Worten anhand von EEG-Aufnahmen. In den Sektionen 1 und 3 gebe ich eine Einführung in die biologischen Grundlagen der Elektroenzephalographie und die Methodik der Datengewinnung. In Kapitel 2 gehe ich detailliert auf die Wavelet-Transformation ein. Es werden sowohl die mathematischen Grundlagen entwickelt als auch die konkrete Anwendung in der Signalverarbeitung im allgemeinen und in meinen Experimenten im speziellen aufgezeigt.

Das Resultat der Experimente findet sich in Kapitel 4. Es erweist sich, dass die *Double-Tree Complex Wavelet Transform*, eine spezielle redundante Variante der diskretisierten Wavelet-Transformation, die Erkennungsraten für alle Aufnahmen deutlich verbessert.

Der zugrundeliegende Datensatz umfasst Aufnahmen verschiedener Typen: Dies betrifft zunächst die Aussprachemodalität (gesprochene oder nur gedachte Worte), aber insbesondere auch die Anordnung der Worte bei der Aufnahme. Dabei gab es die folgenden Varianten (siehe Sektion 3.1):

- Im *blockweisen* Modus wurden alle Samples eines Wortes am Stück aufgenommen, ehe zum nächsten Wort übergegangen wurde.
- Im *sequentiellen* Modus hingegen wurden der Reihe nach alle Worte je einmal aufgenommen, und diese Sequenz wurde mehrfach wiederholt.

Eine zentrale Erkenntnis der Arbeit [3] ist es, dass diese Modalitäten und Aussprachetypen zu höchst unterschiedlichen Ergebnissen führen. In meiner Arbeit zeige ich auf, wie meine Experimente diese These unterstützen und was sich aus der Zerlegung des Eingangssignals in Bestandteile verschiedener Größenordnungen ("Skalen"), die die Wavelet- Transformation bietet, schließen lässt.

In Kapitel 5 wird die Anwendung Wavelet-basierter Vorverarbeitungsmethoden auf EMG-Aufnahmen beschrieben. Das Hauptresultat dieses Abschnittes ist es, dass die *Redundante Diskrete Wavelet-Transformation* unter den hier getesteten Varianten der WT die besten Ergebnisse liefert und – korrekt angewendet – weitaus bessere Erkennungsergebnisse liefert als die gewöhnliche Fourier-Transformation.

Mit komplizierteren Wavelet-basierten Vorverarbeitungsmethoden erreiche ich eine maximale Erkennungsrate von 66,50%, was weniger als zwei Prozentpunkte von der optimalen Erkennungsrate von 68% in [8] abweicht. Wenngleich ich jene Ergebnisse nicht übertreffen konnte, zeigt meine Arbeit aber trotzdem

auf, welche Aspekte der Vorverarbeitung die Erkennungsrate beeinflussen. Eine besondere Rolle kommt dabei dem Kontext eines Feature-Vektors zu.

Die Anhänge umfassen eine Anleitung zur Benutzung der EEG-Klassifikationsumgebung, wie sie an der CMU verwendet werden kann und wie sie von mir und meinen Vorgängern erweitert und angepasst wurde (Sektion A), sowie eine tabellarische Auflistung aller Ergebnisse meiner Experimente.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

Speech recognition provides a natural form for the communication with computers. However, there are a lot of situations in which recognition of audible speech is unfeasible or impossible.

Such situations include quiet places or environments like meetings (during which the need for outside communication may easily arise), but also noisy environments (where normal speech recognition may be all but impossible) and places where uttering speech is physically impossible. Another focus are physically disabled persons who can not utter ordinary speech. For such persons, achievements have been made by the use of electroencephalographic (*EEG*) data to give basic commands to a computer [2], but recognition of actual speech has not yet reached a productive level.

Another motivation for research on EEG-based speech recognition is obviously the desire to enhance the understanding of the human brain. The functionality of the human brain is one of the most intriguing aspects of biological, medical and also philosophical research. As of now, there is no model which describes a "thought" in a brain. EEG speech recognition may provide a way to advance this knowledge.

Of course, the possibility to "read" brainwaves presents various ways for mischief. Thoughts have always been considered the most private part of a person's life, and gaining access to them can be considered as crossing a dangerous line of technique.

For the foreseeable future, however, recognizing random thoughts does not seem probable. First of all, the recording of EEG is bound to a direct contact of the recording device to the subject's head. Also, even if this is given, it seems very unlikely that the multitude of thoughts of a person may be classified. All current efforts are directed at recognizing a limited "vocabulary" of brain activity. In our case, these are spoken and unspoken words, in other cases, mental disposition (e. g. [5]), or illnesses affecting the brain in medical research. Finally, as long as there is no mathematical model which describes the formation of speech in the human brain *for all persons in general*, our recognition task can not be performed without the collaboration of the subject.

Nonetheless, I explicitly declare that I do not intend my research to serve as a lever for invading a person's brain against its will. It should be used solely to gain insight into the work of the human brain to promote knowledge and to increase the quality of life for mankind.

1.2 Biomedical Background of EEG Measuring

The grounding our research is based on is that speech production in the brain is reflected by physiological processes in the brain which by a suitable measuring device can be recorded and converted to a "signal" in terms of signal processing.

There are various ways of detecting brain activity; our choice was mainly limited by two constraints:

- The recording method must have a high temporal solution. This is based on the assumption that the swift variance of human speech finds its reflection in a swift variance of the state of the speech-producing brain regions. Of course, an optimal spatial resolution of the recording method is also desirable.
- In order to facilitate experiments, the method should be easily applicable and not demand excessive preparations and effort to work.

As described in [3], electroencephalography as a recording method optimally satisfies both requirements above. It is also the oldest and most widely used method of brain imaging, meaning that a certain level of experience in EEG processing is available.

The basis of EEG is the recording of *electric activity* in the human brain (hence the name). It is widely known that this activity exhibits properties of waves, especially that in certain conditions, different frequency patterns can be measured, and that these frequency patterns contain information about the state of the brain as a whole and certain brain regions in particular. The assumption of our work is that different unspoken words the subject imagines also incur different properties of the brain waves, so that on this basis a distinction of words is possible.

In section 3.3, I give details about the electrode setup used for these measurements and about the brain regions covered.

1.3 Related Work

This article is intended to continue the thesis of Marek Wester [19], who first investigated EEG recognition at InterACT[1]. EEG recognition, however, has been dealt with earlier: For isolated words, Suppes et al. were successful in [18].

Research directed to recognize EEG data which does not represent words, but rather more general thoughts, has also been done. An example is the *Thought Translation Device* [2] from the year 2000.

Beyond the realm of speech recognition, EEG analysis for clinical tasks has been done for several decades now. Since the first EEG recordings in the 1920s, it has been used successfully as a diagnostic tool for various illnesses affecting the brain. The data analysis methods which have been used also include wavelet transforms, as in [12].

1.4 Goals of my Research

My goal was to extend the previous thesis of Marek Wester [19] by investigating wavelet-based preprocessing methods for EEG data. I implemented the

[1] InterACT is the *International Center for Advanced Communication Technologies*, a joint center between the University of Karlsruhe, Germany, and the Carnegie Mellon University, Pittsburgh, PA, USA. Information can be found at http://interact.ira.uka.de.

Fast Wavelet Transform, the Redundant Discrete Wavelet Transform and the Double-Tree Complex Wavelet Transform as described in the following section and applied them to existing data. This document is intended to describe the results of these experiments. All of these transforms use several parameters that need to be tuned. Therefore, portions of my experiments refer to their optimization.

In the course of my research, it turned out that it would we worthwhile applying wavelet preprocessing methods to electromyographic data as well.[2] The results of this work are described in their own section 5.

1.5 Structure of This Paper

This thesis is laid out the following way:

- After this introduction, section 2 gives a general introduction to the Wavelet Transform, its discrete variants and their implementation.
- Thereafter, section 3 is dedicated to the setup of the EEG experiments. I describe the method of data acquisition, the hardware and software which was used and the way the recognition task was set up.
- Section 4 gives all results of my EEG tasks and exhibits first conclusions.
- Section 5 deals with the application of wavelet preprocessing methods to electromyography.
- Section 6 draws a general conclusion from the experiments reported in this paper.
- The appendices contain a documentation of the software I used, implemented and extended for my work. It is intended as a guide for those who follow after my at CMU. The second part of the appendices gives a tabular overview of all experimental results in EEG and EMG recognition.

2 The Wavelet Transform

In recent years, the various variants of the Wavelet Transform have emerged as an indispensable tool for the analysis of a multitude of different kinds of signals. They have been used successfully in both EEG analysis [12] and EMG analysis [7]. I will give here a short overview of the basics of the Wavelet Transform and the specific implementations I use, based on the textbook [10].

[2] This is data gained by recording electric muscle activity—see section 5 for a thorough description.

2.1 The Continuous Wavelet Transform

First of all, we have to define what a *wavelet* is. We limit ourselves to wavelets in $L^2(\mathbb{R})$, the space of square-integrable functions [15].

Definition 1. For a function $\psi \in L^2(\mathbb{R})$, let

$$\hat{\psi}(\omega) = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} \psi(x) e^{-i\omega x} \mathrm{d}x \tag{1}$$

be its Fourier Transform. A function $\psi \in L^2(\mathbb{R})$ which satisfies the *admissibility condition*

$$0 < c_\psi := 2\pi \int_{\mathbb{R}} \frac{|\hat{\psi}(\omega)|^2}{|\omega|} \mathrm{d}\omega < \infty, \tag{2}$$

is called a *wavelet*.

This condition is relatively easy to satisfy. It should be noted, however, that it implies one central property of any wavelet: Since $\hat{\psi}$ is continuous, ψ must satisfy

$$\int_{\mathbb{R}} \psi(t) \mathrm{d}t = \sqrt{2\pi}\hat{\psi}(0) = 0, \tag{3}$$

i.e. the average of ψ is zero.

Let now ψ be a wavelet. The *Continuous Wavelet Transform (CWT)* maps a one-dimensional function $f \in L^2(\mathbb{R})$ to a function $(\mathcal{W}_\psi f)(a, b)$ on a two-dimensional definition set $(a, b) \in \mathbb{R}^* \times \mathbb{R}$, where $\mathbb{R}^* = \mathbb{R} \setminus \{0\}$. The operator $\mathcal{W}_\psi$ is defined thus:

Definition 2. For $f \in L^2(\mathbb{R})$, $b \in \mathbb{R}$, $a \in \mathbb{R}^*$, let

$$\begin{aligned} (\mathcal{W}_\psi f)(a, b) &= \frac{1}{\sqrt{c_\psi}} \int_{\mathbb{R}} (T^b D^a \psi)(t) f(t) dx, \qquad \text{where} \\ (T^b D^a \psi)(t) &= |a|^{-1/2}\, \psi\left(\frac{t-b}{a}\right). \end{aligned} \tag{4}$$

D^a is the *Dilation Operator*, T^b is the *Translation Operator*.

Equation 4 is nothing but a standard scalar product in $L^2(\mathbb{R})$ (times a constant), i. e.

$$(\mathcal{W}_\psi f)(a, b) = \frac{1}{\sqrt{c_\psi}} \left\langle (T^b D^a \psi)(t) \,\middle|\, f(t) \right\rangle. \tag{5}$$

The interpretation of $\mathcal{W}_\psi f$ is that $|(\mathcal{W}_\psi f)(a, b)|$ is *large* if f has got a “detail” of “size” a at $t = b$. Thus, the CWT has the following central properties:

- It is *localized* both in time and in frequency: $(\mathcal{W}_\psi f)(\cdot, b)$ describes f around $t = b$, and $(\mathcal{W}_\psi f)(a, \cdot)$ represents the frequencies of f which correspond to the scaling parameter a.

- It extracts details on different scales, much like the Fourier Transform, but as I will explain below, the Wavelet Transform can be "fine-tuned" much better than the Fourier Transform.
- Given the right choice of filters, it may simulate a decomposition of f into frequency bands.

The CWT can be implemented by discretization, but it remains a highly redundant representation of f. Furthermore, in applications f is always a discrete signal. Therefore several discrete variants of the Wavelet Transform which may be computed efficiently have been developed.

2.2 Multi-Resolution Analysis

The Continuous Wavelet Transform defined above offers a decomposition of a signal into stretched and shifted versions of a wavelet ψ. The first step towards an efficient discretization of this tranformation is to ask how to reduce the number of coefficients (i. e. of values of $\mathcal{W}_\psi f$) which are needed to convey the full information about f, such that f could be reconstructed from these values.

The way to achieve a decomposition of a function $f \in L^2(\mathbb{R})$ into countably many coefficients is the use of a *Multi-Resolution Analysis (MRA)*([10], Chapter 2.2). All wavelet methods I have used in this work are derived from the MRA.

Before giving the definition for the MRA, I first define a new notation which will prove useful in dealing with the discretization of the Wavelet Transform:

Definition 3.

$$\text{Let} \qquad f_{a,b} := T^{2^a b} D^{2^a} f = 2^{-a/2} f(2^{-a}x - b). \tag{6}$$

Now we can define the MRA:

Definition 4. An MRA is a sequence $(V_j)_{j\in\mathbb{Z}}$ of closed subspaces of $L^2(\mathbb{R})$ with the following properties:

1. For $j \in \mathbb{Z}$ holds $V_{j+1} \subseteq V_j$.
2. $\overline{\bigcup_{j\in\mathbb{Z}} V_j} = L^2(\mathbb{R})$.
3. $\bigcap_{j\in\mathbb{Z}} V_j = \{0\}$.
4. $f \in V_0 \Longleftrightarrow 2^{-j/2} f(2^{-j}\cdot) \in V_j$.
5. There is a *Scaling Function* $\varphi \in V_0$ such that $\{\varphi(\cdot - b) | b \in \mathbb{Z}\}$ is an orthonormal base (ONB) in V_0.

This implies that for each j, the set $\{\varphi_{j,k} | k \in \mathbb{Z}\}$ forms an orthonormal base of V_j.

If such an MRA has been found, the scaling function has the representation

$$\varphi = \sum_{k\in\mathbb{Z}} h_k \varphi_{-1,k} \tag{7}$$

with $\sum_{k\in\mathbb{Z}} |h_k|^2 = 1$. This is a direct consequence of the relation $V_0 \subseteq V_{-1}$ ([10], Lemma 2.2.3 and Satz 2.2.9).

We can now construct a decomposition of $L^2(\mathbb{R})$ into mutually orthogonal subspaces. Let W_j be the orthogonal complement of V_j in V_{j-1}, i.e. for $j \in \mathbb{Z}$, define W_j such that

$$V_{j-1} = V_j \oplus W_j \qquad \text{and} \qquad V_j \perp W_j. \tag{8}$$

Then the W_j have got the following properties:

Theorem 5. 1. For $j \neq m$, $W_j \perp W_m$.

2. The spaces W_j inherit the scaling property 4.4 from the spaces V_j.

3. Since for each $m > j$, we can write $V_j = \bigoplus_{\mu=j+1}^{m} W_\mu \oplus V_m$, we see by letting $j \longrightarrow -\infty$ and $m \longrightarrow \infty$ that the W_j exhaust the space $L^2(\mathbb{R})$.

These three properties imply that if we find a $\psi \in L^2(\mathbb{R})$ such that $\{\psi_{j,k} \,|\, k \in \mathbb{Z}\}$ is an ONB of W_j for each $j \in \mathbb{Z}$, the set $\{\psi_{j,k} \,|\, j,k \in \mathbb{Z}\}$ will be an ONB of $L^2(\mathbb{R})$. How can we construct such a ψ? The theory of the MRA gives an answer to this.

Theorem 6. If $(V_j)_{j\in\mathbb{Z}}$ is an MRA with scaling function φ, and if ψ is defined by

$$\psi(x) = \sum_{l\in\mathbb{Z}} g_l \varphi_{-1,l} = \sqrt{2} \sum_{l\in\mathbb{Z}} g_l \varphi(2x - l) \qquad \text{with} \qquad g_l = (-1)^l h_{1-l}, \tag{9}$$

where h_l is given by the formula from equation 7, then:

1. $\{\psi_{j,k} \,|\, k \in \mathbb{Z}\}$ is an ONB of W_j for each $j \in \mathbb{Z}$
2. $\{\psi_{j,k} \,|\, j,k \in \mathbb{Z}\}$ is an ONB of $L^2(\mathbb{R})$
3. ψ is a wavelet with $c_\psi = \sqrt{2 \ln 2}$.

Thus, for $f \in L^2(\mathbb{R})$, we have

$$f = \sum_{j,k\in\mathbb{Z}} \langle f \,|\, \psi_{j,k} \rangle \, \psi_{j,k} \tag{10}$$

and we have an expansion of f into a weighted sum of countably many stretched and shifted copies of a wavelet ψ (provided that the sum converges).

2.3 The Fast Wavelet Transform According to Mallat

The decomposition of a function $f \in L^2(\mathbb{R})$ into countably many stretched and shifted copies of a wavelet ψ has given rise to several algorithms to calculate the coefficients of this composition. The most basic algorithm is the *Fast Wavelet Transform (FWT)* which was first introduced in 1989 by S. G. Mallat in his paper [13].

Let $(V_j)_{j\in\mathbb{Z}}$ be an MRA with scaling function φ and corresponding wavelet ψ. Assume that $f \in V_0$. For $j \in \mathbb{N}_0$ and $k \in \mathbb{Z}$, we define

$$\begin{aligned} c_k^j &= \langle f \,|\, \varphi_{j,k}\rangle \qquad \text{and} \qquad c^j = (c_k^j)_{k\in\mathbb{Z}} \\ d_k^j &= \langle f \,|\, \psi_{j,k}\rangle \qquad \text{and} \qquad d^j = (d_k^j)_{k\in\mathbb{Z}}. \end{aligned} \tag{11}$$

Then since the $\varphi_{0,k}$ form an ONB of V_0, we have the representation

$$f = \sum_{k\in\mathbb{Z}} c_k \varphi_{0,k}. \tag{12}$$

The central idea of the FWT is to perform all calculations on these coefficients without ever resorting to the original wavelets. The definitions of c^j and d^j and the scaling properties of φ and ψ give rise to the following equation:

$$\begin{aligned} c_k^j = \langle f \,|\, \varphi_{j,k}\rangle &= \sum_{l\in\mathbb{Z}} h_l \, \langle f \,|\, (\varphi_{-1,l})_{j,k}\rangle = \\ = \sum_{l\in\mathbb{Z}} h_l \, \langle f \,|\, \varphi_{j-1,2k+l}\rangle &= \sum_{l\in\mathbb{Z}} h_l c_{2k+l}^{j-1} = \sum_{m\in\mathbb{Z}} h_{m-2k} c_m^{j-1}. \end{aligned} \tag{13}$$

Similarly, we have

$$d_k^j = \langle f \,|\, \psi_{j,k}\rangle = \sum_{l\in\mathbb{Z}} g_l \, \langle f \,|\, (\varphi_{-1,l})_{j,k}\rangle = \sum_{m\in\mathbb{Z}} g_{m-2k} c_m^{j-1}. \tag{14}$$

h_l and g_l are given by equations 7 and 9. Let

$$\ell_2 = \left\{ (x_k)_{k\in\mathbb{N}} \,\middle|\, x_k \in \mathbb{R}, \sum_{k\in\mathbb{N}} |x_k|^2 < \infty \right\} \tag{15}$$

be the space of real square-summable sequences. On this space, we define

$$\begin{aligned} H : \ell_2 \longrightarrow \ell_2, (c_k) \mapsto \left(\sum_{m\in\mathbb{Z}} h_{m-2k} c_m \right) \qquad \text{and} \\ G : \ell_2 \longrightarrow \ell_2, (c_k) \mapsto \left(\sum_{m\in\mathbb{Z}} g_{m-2k} c_m \right). \end{aligned} \tag{16}$$

Then for each $j > 0$,

$$c^j = Hc^{j-1} \qquad \text{and} \qquad d^j = Gc^{j-1}. \tag{17}$$

These operations can be easily implemented in any programming language. In the section "Experimental Setup", I have described the details of the implementation I used.

It turns out that if the initial coefficient sequence c^0 is finite, and w. l. o. g. its length is divisible by 2^L, where L is the maximal decomposition level we use, each c^j and d^j is *precisely half as long* as the data c^{j-1} from which it stems. The naïve interpretation of this fact is that c^j and d^j contain all the information of the original sequence, and since the operations used to create c^j and d^j were complimentary, we do not have any redundancy—the amount of information remains the same, but its representation is significantly different.

Knowing all this, we can regard the convolutions in 13 and 14 as a filtering with an LTI (linear time-invariant) filter and successive downsampling by the factor two. Figure 1 gives a schematic representation of this structure, however, it should be clear that the FWT algorithm interweaves these two steps in the most efficient manner possible.

In order to use this algorithm, two additional problems have to be addressed. The first one is how to deal with data of arbitrary length, i.e. the situation that a filter extends beyond the range of time for which data is available. Several options exist, e.g. zero-padding, smooth padding, extrapolation methods etc. Since my experiments dealt with non-periodic data, I have always used zero-padding.

The second problem is how to obtain the initial sequence c^0. See [10], chapter 3.1.1 for a detailed explanation.

Our signal f is discretely sampled, i. e. $f_k = f(k), k \in \mathbb{Z}$. (We assume w. l. o. g. that f has got sampling rate 1, i. e. there is one sample for each time unit.) The coefficients f_k ideally should be the coefficients of an expansion of f according to the scaling function φ, i.e. the function $\bar{f}$ defined by

$$\bar{f}(t) = \sum_{k \in \mathbb{Z}} f_k \varphi(t-k) \tag{18}$$

should equal f.

This is generally not true for orthogonal scaling functions. However, the error when using the signal f_k as sequence c_k^0 remains bounded and can be estimated if f satisfies certain regularity properties [10].

In my experiments, I chose the "direct" way of letting $f_k = c_k^0$. As described above, this incurs a certain margin of error, but on the other hand greatly simplifies the underlying mathematics. It may be a question for future work whether this error affects the results of the classification experiments at all—a heuristic argument against this is that the error occurs uniformly in *all* calculations and does not reduce the amount of information in the resulting feature vectors. Thus, the influence of this error on the classification, which essentially measures the similarity and distinctness of feature vectors, should be relatively limited.

2.4 The Double-Tree Complex Wavelet Transform

The FWT described above is computationally very efficient. However, it turns out that it has got several flaws [16]. For us, particularly the lack of shift invariance may cause problems in the discrimination of different patterns.

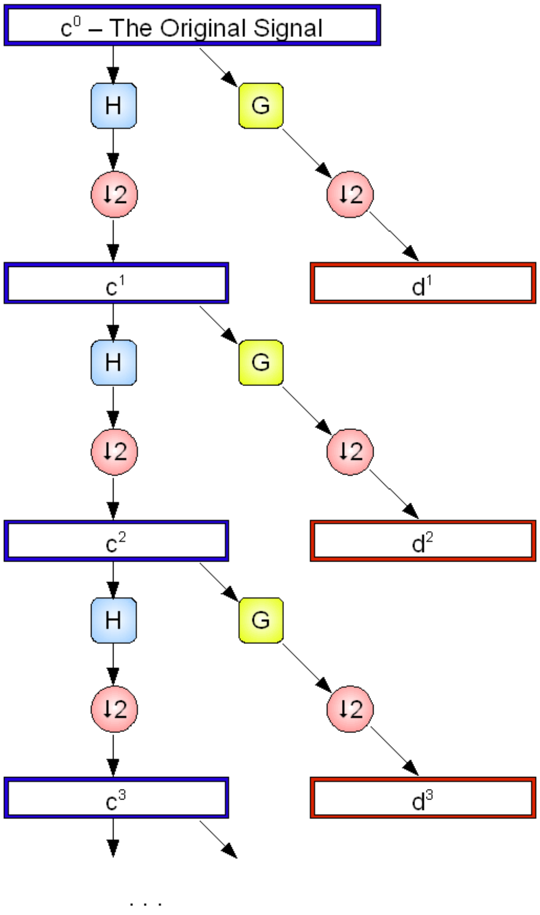

Figure 1: The Fast Wavelet Transform as a Partial Binary Tree

One solution to this problem, proposed by Nick Kingsbury and developed in detail in [16], is the *Dual-Tree Complex Wavelet Transform (DTCWT)*. The essential ideas of this algorithm are as follows:

- We let *two* trees grow instead of one. Thus, we achieve a redundancy of factor two in the decomposition. (This explains why the transform is called “Double-Tree” transform.)

- We choose the filters carefully such that the *redundancy* of the representation becomes an actual *oversampling* of the original signal by the factor two. This property is responsible for the improved shift-invariance of the DTCWT.

- The two trees can be interpreted as the real and the imaginary part of a discrete Wavelet Transform with a *complex* wavelet. Hence the name "complex" wavelet tranform.

- The resulting feature vector is calculated by interpreting two corresponding coefficients in the two trees as real and imaginary part of a complex coefficient and then calculating the complex absolute value of this complex coefficient. Thus, the resulting feature vector has got the same dimensionality as the feature vector when the FWT is used.

- The two trees are calculated separately; no complex arithmetics are necessary.

Figure 2 gives a graphical image of this algorithm.

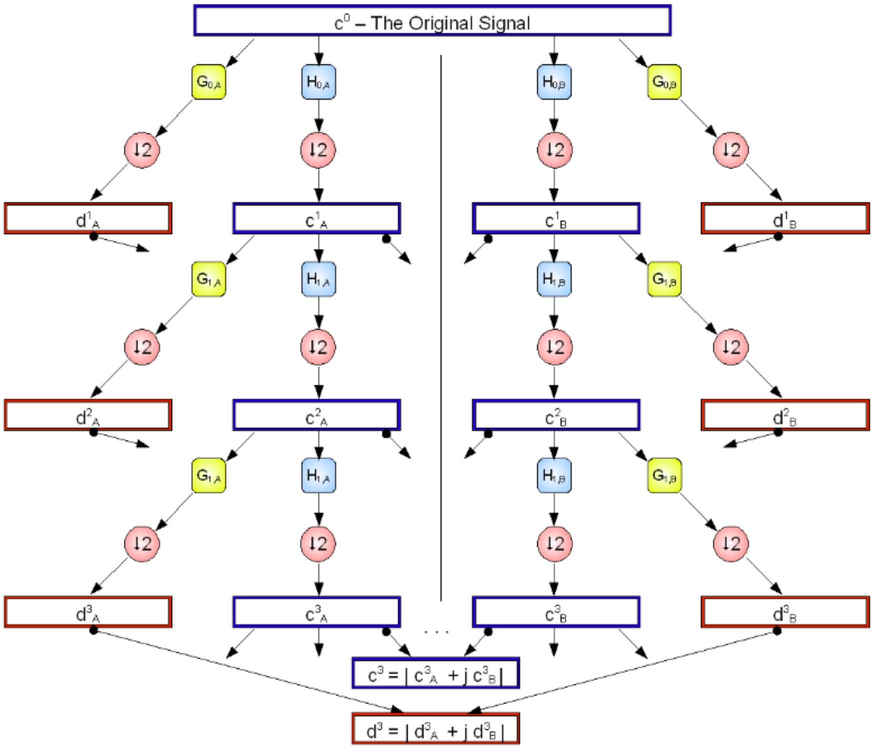

Figure 2: The Double-Tree Complex Wavelet Transform as a Partial Binary Tree

The choice of filters for this alogrithm turns out to be significantly more difficult than for the FWT. Altogether, we need *four* low-pass filters: In each tree, there is one high-pass filter for the first step of the transformation, and one high-pass filter for all subsequent stages. In the figure, these filters are denoted $H_{0,\Delta}$ for the first stage and $H_{1,\Delta}$ for the subsequent stages, where Δ takes the values A or B for tree A or tree B.

The high-pass filters, however, are always calculated according to the corresponding low-pass filter as described in section 2.3; the high-pass filter corresponding to $H_{i,\Delta}$ is called $G_{i,\Delta}$, where i is 0 or 1.

Thus in order to utilize this algorithm, we need to provide the four low-pass filters in advance.

In summary, the calculations are as follows:

1. Calculate the coefficients c^i_Δ according to section 2.3. Replace the filters H and G appropiately, i. e. use filters $H_{0,\Delta}$ and $G_{0,\Delta}$ for the first step and $H_{1,\Delta}$ and $G_{1,\Delta}$ for the subsequent steps.

2. Determine the final coefficients $c^i = (c^i)_k$ and $d^i = (d^i)_k$ via:

$$\begin{aligned}(c^i)_k &= |(c^i_A)_k + j \cdot (c^i_B)_k| = \sqrt{(c^i_A)^2_k + (c^i_B)^2_k} \\ (d^i)_k &= |(d^i_A)_k + j \cdot (d^i_B)_k| = \sqrt{(d^i_A)^2_k + (d^i_B)^2_k}\end{aligned} \tag{19}$$

   for each k, where $j = \sqrt{-1}$.

There are several possible improvements to this algorithm (mainly based on using a more complex filter setup [16]), which I did not use for my research.

2.5 The Redundant Discrete Wavelet Transform

The *Redundant Discrete Wavelet Transform (RDWT)* is another attempt to obtain shift-invariance in the Wavelet Transform. It resembles the FWT because it uses only one decomposition tree and only one high-pass filter, so the results of the FWT and the RDWT are directly comparable. A good explanation of the RDWT can be found in [17].

The RDWT achieves a multi-scale representation of the original signal not by downsampling the transformed signal after each step (see figure 1), but it rather upsamples the two filters (the high-pass filter and the low-pass filter) by the factor two in each step.

Thus, the main difference between the FWT and the RDWT is not the actual way the coefficients are calculated, but rather the time grid. Figure 3 shows this difference graphically:

In summary, the calculation works as follows:

1. Initialize the filters (h_k) and (g_k) as described in the FWT section. The initial sequence c^0 is given, as well as the maximum decomposition level L. Let $i := 1$.

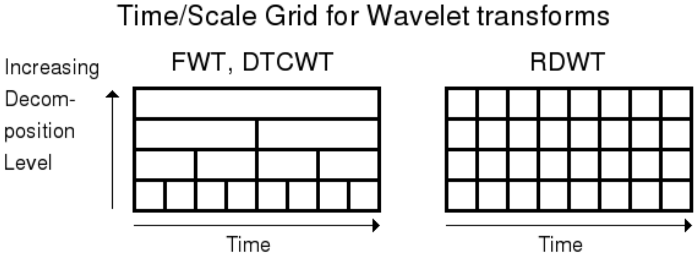
Time/Scale Grid for Wavelet transforms
Increasing Decom-position Level
FWT, DTCWT
RDWT
Time
Time

Figure 3: Comparison between the time grids for the FWT and the RDWT

2. Calculate c^i and d^i, $i \leq L$, by

$$\begin{aligned} c_l^i &= \sum_{k \in \mathbb{Z}} c_k^{i-1} h_{k-l} \\ d_l^i &= \sum_{k \in \mathbb{Z}} c_k^{i-1} g_{k-l} \end{aligned} \tag{20}$$

3. Upsample the filters (h_k) and (g_k), i. e.

$$\begin{aligned} h_k^{NEW} &= \begin{cases} h_{k/2}^{OLD} & \text{if } k \text{ is even} \\ 0 & \text{if } k \text{ is odd} \end{cases} \\ g_k^{NEW} &= \begin{cases} g_{k/2}^{OLD} & \text{if } k \text{ is even} \\ 0 & \text{if } k \text{ is odd} \end{cases} \end{aligned} \tag{21}$$

4. Jump to step 2 if $i < L$ and increment i by 1.

3 Experimental Setup

This section describes how the data collection for EEG experiments was run, and it explains the software system which I used and extended for the purpose of EEG recognition.

All my experiments were done with data collected by Marek Wester for his Master's Thesis [19] and by Jan Calliess for his thesis [3].

3.1 Corpora and Modalities

All EEG recognition was done *word-based*. This means that the recognition task was to distinguish between a set of *whole words*, called the *corpus*. In order to concentrate on the optimization of signal preprocessing, I limited the different kinds of recording setups in the following way compared to [19]:

From the corpora (vocabularies) of [19], section 5.1, I used only two sets, namely the sets *digit5* and *alpha*:

Name	Vocabulary
Digit5	{ One, Two, Three, Four, Five }
Alpha	{ Alpha, Bravo, Charlie, Delta, Echo }

Each corpus consists of five *words*. In addition to these words, in each recording five to seven samples of *silence* were recorded, i. e. the subject was told not to think of anything (as much as possible). These samples were never used for evaluation, but always entered the training set.

The rationale for the choice of corpora is that the different words should *not* convey a hidden meaning to the user. Otherwise, the recognition results might be biased by the influence of these hidden connotations, which would obviously leave a certain "trace" in the respective brain activity recording.

I experimented with only two speech *modalities* as opposed to five modalities in [19], section 5.2. These were *Unspoken Speech* and *Spoken Speech*. The latter consists of ordinarily pronounced words, whereas for Unspoken Speech, the subject (the person whose thoughts were recorded) was asked to think of pronouncing the respective word *without actually moving any muscle*.

In each session, every word of the vocabulary was recorded between 25 and 30 times. (The varying numbers occurred because in some cases, single samples had to be discarded later on due to too much noise in the data.) One further property of the recording setup turned out to be very important. There were *blockwise* and *sequential* recordings. In sessions with blockwise recordings, the subject recorded *all* samples of one vocabulary item as a block before proceeding to the next vocabulary item. In sequential sessions, a sequence of one sample of each word (i. e. five samples altogether) was recorded, and this was repeated the above-mentioned 25 to 30 times. The table below gives an overview of this process (here r means the number of repetitions):

Name	Ordering
Blockwise	{ Alpha $\overset{r}{\ldots}$ Alpha, Bravo $\overset{r}{\ldots}$ Bravo, ... }
Sequential	{ Alpha, Bravo, Charlie, Delta, Echo, Alpha, Bravo, ... }

3.2 The Recording Process

Creating useful EEG recordings is particularly difficult because the data is especially susceptible to any kind of noise and artifacts:

- Any kind of muscular activity, no matter whether it is voluntary or not, causes significant artifacts. This can, however, also be used to control the recording device, as I explain below.

- The brain state of the subject will greatly influence the recording. This brain state is influenced by physical feelings such as pain or tiredness, but it also involves mental issues like concentration, boredom or stress. Indeed, M. Honal developed a system to determine a user's mental state by EEG recordings [5].

- Nothing is harder to control than thoughts—the subject must be relaxed, and its mind should be free of other matters.

- The placement of electrodes on the scalp means that the brain signal is disturbed by the subject's skull, skin and hair.[3] In these recordings, a standard conductive gel is used to improve connectivity.

- One electrode measures the neuronal activity not only in a localized brain region, but receives signals from the whole brain.[4] The situation is further complicated by the fact that with our method, we map the three-dimensional brain onto the two-dimensional scalp surface of the subject. This means that even by inceasing the electrode density on the subject's scalp, the resolution of the recording cannot exceed a certain intrinsic limit.

To limit the number of artifacts, we used a specific recording environment and an adapted recording software. The recordings were taken in quiet office rooms, with the subject sitting in front of a desk. A screen showing instructions to the subject stood on the desk.

Opposite the subject, a supervisor controlled the recordings on a second screen. The recording was started and stopped for every single utterance, and if the supervisor or the subject itself noticed a flaw in the recordings (particularly artifacts from eye movement), the utterance was deleted and re-recorded.

Before the recording session started, the subject was explained this process and the aims of the experiment. The subject was told that he or she could quit the process at will, and also that it could ask for breaks at any time. During breaks, light snacks and drinks were provided, so that the subject could recover concentration.

Since we experimented with word-based recognition, a major problem during the actual recording was to determine the signal bounds. These are not at all obvious from the signal. Our solution was to let the subject do an eyeblink before and after the actual process of thinking/uttering a word. These eyeblinks cause very recognizable artifacts in the signal which can be detected automatically. However, an utterance during which the subject inadvertently moved or blinked the eyes had to be retaken since this would have rendered the signal useless. Marek Wester developed the original eyeblink detector, which was refined by Jan Calliess.

Altogether, the steps for the recording of each utterance were (adapted from [19], Section 4.1.2):

1. The subject sat quietly and without any movement in front of a white screen.

2. The supervisor started the recording process by pressing a button.

[3] This can be improved by surgically inserting electrodes into a person's brain, but this is not desired in our kind of research due to ethical and also practical reasons.

[4] [3] contains some estimates on how much a signal generated in one area of the brain influences the potential measured by an electrode at a different position.

3. The screen showed the words which should be uttered in black letters. In brackets it showed the modality of the utterance.

4. After 1 second the screen showed the words: “inhale and exhale”.

5. After another second the screen turned black.

6. After another 2 seconds the screen turned white. (This was done to remove the visual stimulus from the subject’s mind.)

7. The subject was instructed to wait for about 1 second.

8. The subject blinked, uttered the word which had been shown on the screen in step 3 and blinked again.

9. The supervisor stopped the recording with the pressing of a button after the second eyeblink. (The supervisor could visually detect the eyeblink from his control monitor.)

3.3 EEG Recording Hardware

The recordings were done with “ElectroCap” EEG recording devices. They have the form of a flexible cap which is attached to the subject’s head. Electrodes are sewn into the cap at certain positions.

After attaching the cap to the subject’s head, it was fixed by a strap around the body of the subject. Then, a standard connectivity gel was inserted into the electrodes in order to get a good connection to the subject’s scalp.

We experimented with various caps having different electrode layouts. The cap used for the first experiments, dubbed *low-density cap*, had an electrode layout conforming to the international 10-20 standard [6]. In the course of the experiments in [3], a new *high-density cap* with 128 electrodes was obtained. For each layout, we actually used two caps of different sizes according to the subject’s head circumference. The two layouts are shown below.

During each recording session, 16 electrodes were actually used. In addition, two clips were fixed to the ears of the subject in order to serve as a common reference for the measurements. In the left picture, these are marked A1 and A2, they are not shown in the right picture.

In both figures, the set of used electrodes is drawn in a darker shade than the remaining ones. The choice of electrodes in the original low-density setup reflects our knowledge about the production of speech: electrodes O1 and O2 primarily reflect optical/visual information, and F8 was left out since speech production mainly occurs on the left hemisphere of the brain ([19], Section 4.1.4). In the high-density setup, we tried to increase resolution in the area identified as important by the results from the low-density experiments, the orofacial motor cortex ([3], Section 3.2.1). In addition, electrode 1 optimally detects the eyeblinks used for signal segmentation.

In pilot experiments we investigated whether the results based on the high-density cap differ from the results based on the low-density cap when the same

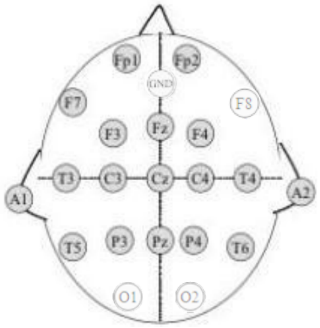

Figure 4: Electrode Setup on Low-Density ElectroCap

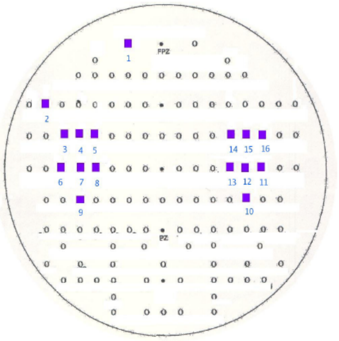

Figure 5: Electrode Setup on High-Density ElectroCap

number (i. e. 16) and position of electrodes were used. The differences were small with respect to the variance between sessions, a common trend could not be detected. For this reason, I do not distinguish between the different caps further on in this document. As a final remark, note that this result is consistent with [19], Section 6.8, where it is shown that leaving out electrodes in "unimportant" regions of the brain hardly affects classification results. But it also supports the unproved hypothesis in section 3.2 that increasing the electrode density cannot improve the recognition data beyond a certain threshold, since the additional data will be fully redundant.

The ElectroCap signals were amplified and digitalized with a VarioPort$^{\text{TM}}$ amplifier. All recordings were done with a sampling rate of 300 Hz. The recording software was specifically created for this task, it is described in the next section.

3.4 Processing and Classification Software

The recordings were done with the UKA EMG/EEG Studio v2.0 software [14]. This program combines routines for recording the amplified and digitalized data, giving instructions to the subject and displaying a control interface to the supervisor.

For the classification of the recorded data, I used the state-of-the-art speech recognizer *JANUS* developed at the CMU and the University of Karlsruhe. My recording setup was based on a speech recognizing setup adapted by Marek Wester to cover the needs of EEG recognition (appendix A of [19]). My own modifications are described in appendix A.

In JANUS, every word was modeled with a five-state left-to-right HMM. The emission density of each state was modeled as a multivariate Gaussian mixture density with 25 Gaussians. All 16 channels were used for the recognition task. LDA was applied to the feature vectors in all experiments, the optimal dimension of the vectors after the LDA was subject to my experiments and is described in detail in the respective result sections.

I only did offline evaluation in order to obtain the best possible accuracy in evaluating different preprocessing methods and to be able to compare the results of different methods. In every experiment described here, we used a "leave-one-out" method, i. e. we ran the recognizing task as many times as there were samples of each vocabulary word, and in each round, one sample of each word was chosen for evaluation, while the remaining samples were used as a training set.

4 Results of EEG Speech Recognition

In this section, I will show which results my experiments yielded. I have split up the entire data into several tables and diagrams, each of which answers a separate question about the "tuning" of the EEG recognizer.

To increase readability, I give a list of all sessions and speakers featuring in this document in table 1. Note that if a pair of sessions is marked as "Double Session", it means that these recordings were actually done as one session, i. e. without removing the ElectroCap between the recordings. This allows us to directly compare different modalities of the recordings.

Speaker	Sex	Session	Modality	Seq. Mode	Remarks
1	Male	1	unspoken	blockwise	
2	Male	1	unspoken	blockwise	Double Session
		2	spoken	blockwise	
		3	unspoken	sequential	Double Session
		4	spoken	sequential	
		5	unspoken	sequential	
3	Female	1	unspoken	blockwise	
4	Male	1	unspoken	blockwise	

Table 1: Session List

4.1 Preprocessing—Are Wavelets Really Better?

The first question I adressed is whether there is any variant of the Wavelet Transform (the specific transformations I used are defined in section 2. To compare the qualities of the Wavelet Transforms and traditional methods, I applied a windowed Fourier transformation as used by [19] with a window size of 26.6 ms and a window shift of 4 ms which decomposed the input signal into 12 subbands. Thus, each input channel yielded an output matrix with 12 rows as a result. Since we used 16 channels (as described in section 3.3), this gave us a total of $12 \cdot 16 = 192$ coefficients per frame.

For all wavelet-based algorithms I used decompositions of the signal up to the levels $L = 4$ and $L = 8$. In either case, both the detail coefficients c^j *and* the approximation coefficients d^j ($j = 1 \ldots 8$) were used in the feature vectors. Experiments using only the detail coefficients showed a large drop in the performance of the recognizer.

For the FWT algorithm, I used a Daubechies-4 filter (section 2.4.3 of [10]).

For the DTCWT algorithm, I used a Daubechies-4 filter pair for the first stage of the decomposition. For the second stage, I compared:

- a 6-tap q-shift filter *(DTCWT1)*
- a 14-tap q-shift filter *(DTCWT2)*

according to [9].

In all cases, a Linear Discriminant Analysis (LDA) was applied to the feature vectors. The dimension of the feature vectors was reduced to 35, which is a first estimate for an optimal LDA dimension. The results of the LDA tuning are described below.

The results can be seen in table 2. They are represented graphically in figure 6. This chart is structured so that most different combinations of speakers and modalities are represented and may be easily compared.

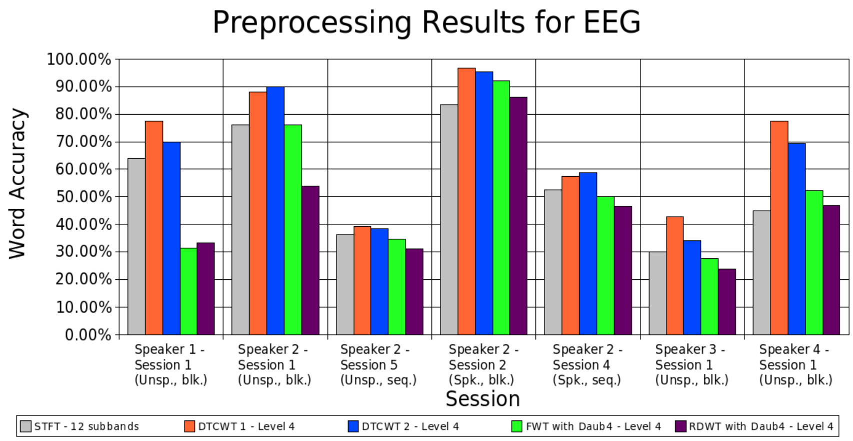

Figure 6: Comparison of Preprocessing Methods for EEG

We see that in all cases, the DTCWT (with two different filter setups) performs best. Therefore, I did all my further experiments only with the DTCWT. Second in line comes the Fourier Transform, which gives quite suitable results.

The two variants of the real discrete wavelet transform—the FWT and the RDWT—yield very bad results. The weakness of the FWT is relatively unsurprising and consistent with [7], however, the fact that the RDWT is mostly even worse than the FWT comes unexpected, particularly since it performs quite good in the case of EMG recognition, see section 5.3. It might be instructive to explore *why* the RDWT performs so badly; however, this is beyond the scope of this work.

After having established the superiority of the DTCWT, I continued my experiments with this transformation only. The next question was how to tune its parameters.

A major result of [3] is that the recognition results vary greatly between different *ordering* setups, i. e. between blockwise and sequential recordings. That document proposes that the reason for this is a kind of time-related artifact which increases the recognition rate for blockwise recordings, thereby producing incorrect results. I will describe my findings for these orderings in different sections and discuss my interpretation of the results below.

4.2 Unspoken Words, Blockwise Recordings

In this section, we *only* deal with blockwise recordings of unspoken words, i. e. no muscular movements were involved.

First I investigated the decomposition level I had to use to obtain best results. In all cases, I used the complete set of coefficients we get: both the detail coefficients *and* the lowpass (approximation) coefficients. As I said in section 4.1, experiments with the lowpass coefficients left out have always shown very bad results, which I do not chart here.

Again, in all experiments, an LDA transformatoin was applied to the feature vectors which reduced their dimension to 35. The one exception was the wavelet decomposition up to level 1—here, the dimensionality of the combined feature vectors of all channels before the LDA was $1 \cdot 2 \cdot 16 = 32$ only, so the dimension after the LDA was also set to 32.

The results can be seen in table 3. They are represented graphically in figure 7. As always, I tried two different filter setups called "DTCWT 1" and "DTCWT 2", see 4.1 for their descriptions. Note that the columns "DTCWT x - 2 .. 5", $x = 1$ or 2, refer to a decomposition till level five where the decomposition coefficients of the first step (i. e. those referring to the smallest possible details) were left out.

Unexpectedly, the results were best for very small decomposition levels (1 to 3). I discuss possible conclusions of this in the next section, in relation to the results for sequential recordings.

The next question was how to optimally choose the dimension to which the LDA should reduce the dimensionality of the feature vectors. I again compared feature vectors preprocessed with the DTCWT with a 6-tap and a 14-tap filter, called DTCWT 1 resp. DTCWT 2. Since on average, a decomposition level of 3 yielded the best recognition results, I chose this level for all experiments. Thus the dimension of the feature vectors before the LDA dimensionality reduction was $3 \cdot 2 \cdot 16 = 96$. For the dimension after the LDA, I tried the values 16, 35 and 60. [19] proposes that for data preprocessed via an STFT, the dimension 16 yields optimal results. My experiments, however, showed that a higher LDA dimension may yet improve the result. The list of word accuracy percentages is found in table 4 and are charted in figure 8.

We observe several things:

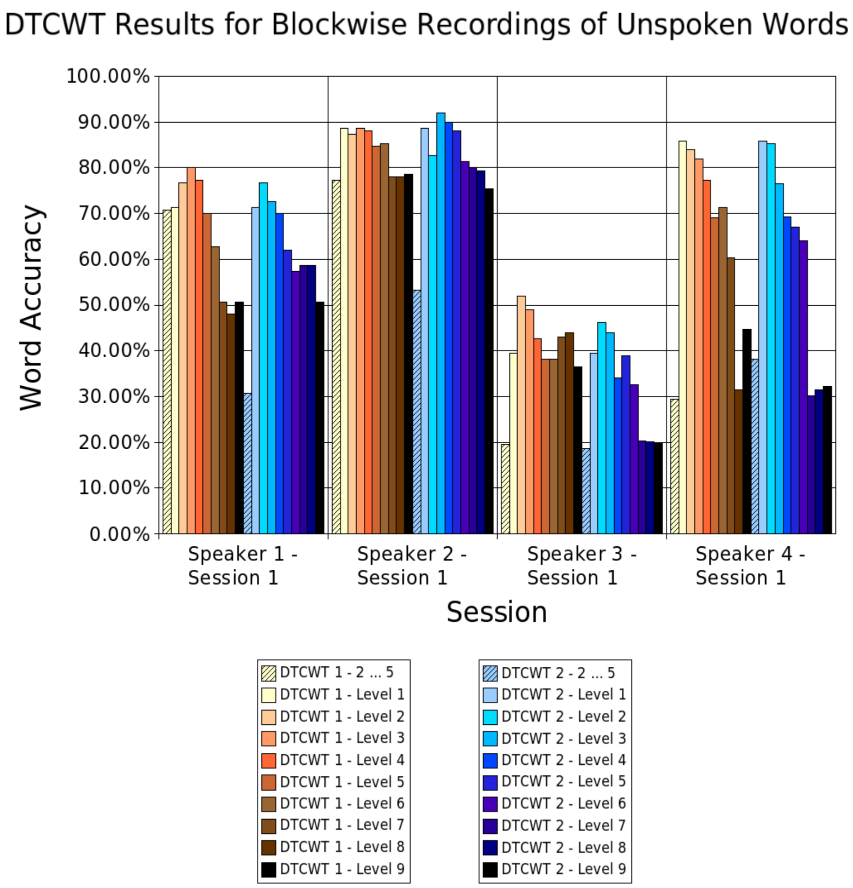

Figure 7: Comparison of DTCWT Decomposition Levels for Blockwise EEG Recordings of Unspoken Words

1. A reduction of the feature vector dimensionality from 35 to 16 dimensions can lead to better or worse recognition results.

2. An increase of the feature vector dimensionality from 35 to 60 dimensions incurs worse recognition results. This is consistent with [19], where an LDA dimension of 35 is proposed as an upper limit for good accuracy.

3. In all cases, the difference between word accuracy values for different LDA dimensions is not very high. Compared to the differences between various decomposition levels (figure 3), the LDA dimension does not seem to be a pivotal parameter for tuning the system.

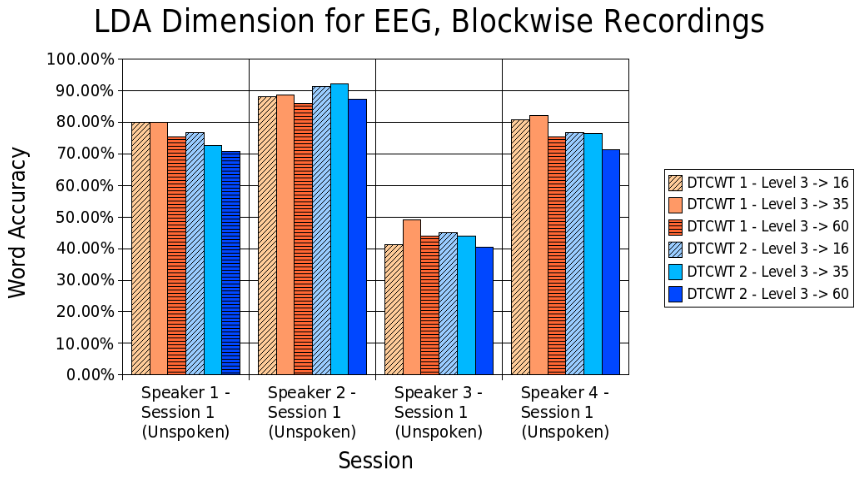

Figure 8: Comparison of LDA Dimensionality for Blockwise EEG Recordings of Unspoken Words

4.3 Unspoken Words, Sequential Recordings

In this section, we deal with sequential recordings of unspoken words.

Just as above, I first took a look at the decomposition level which yields optimal classification results. As usual, I used both detail and lowpass coefficients for classification, and the resulting feature vectors were processed by an LDA which reduced their dimension to 35.

The central result of this section is that the classification requires *both high-frequency and low-frequency* coefficients to work. The optimal decomposition level was at or around 8, see table 5 and the corresponding figure 9.

These results differ completely from those we got with blockwise recordings. Therefore, when investigating the optimal LDA dimension, I chose to use a decomposition level of 8 as a basis. Since I used (as always) both detail and lowpass coefficients, this means that each feature vector had $8 \cdot 2 \cdot 16 = 256$ coefficients. By LDA, this number was reduced to 16, 30, 35, 40 or 60. The results are found in table 6 and charted in 10.

We see that the best LDA dimension is always less or equal to 35; in most cases it is around 35. This is again consistent with [19].

4.4 Comparison Of Blockwise and Sequential Recordings

The main result so far is that blockwise and sequential recordings behave differently. Not only are the recognition results much better in the blockwise case, but also the optimal parameters for the recognition are very different. Figure 11 visualizes the different recognition results. I chose the optimal result achieved by any of the DTCWT filters, but with a fixed LDA of 35.

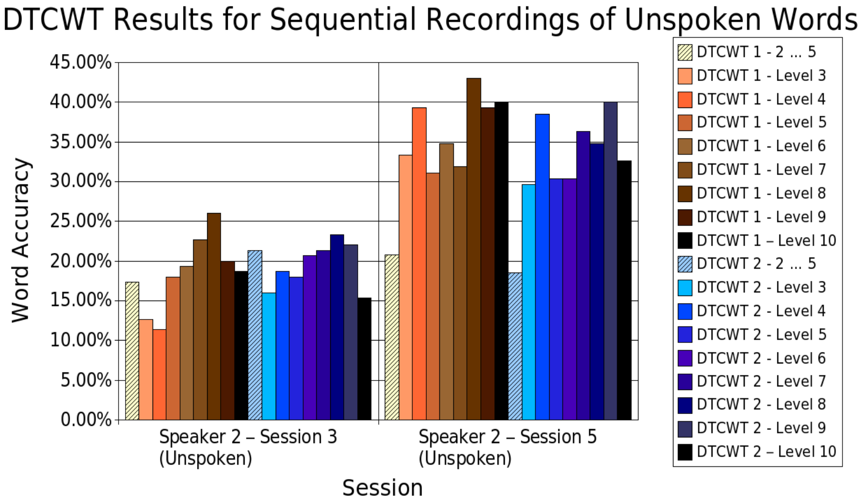

Figure 9: Comparison of DTCWT Decomposition Levels for Sequential EEG Recordings of Unspoken Words

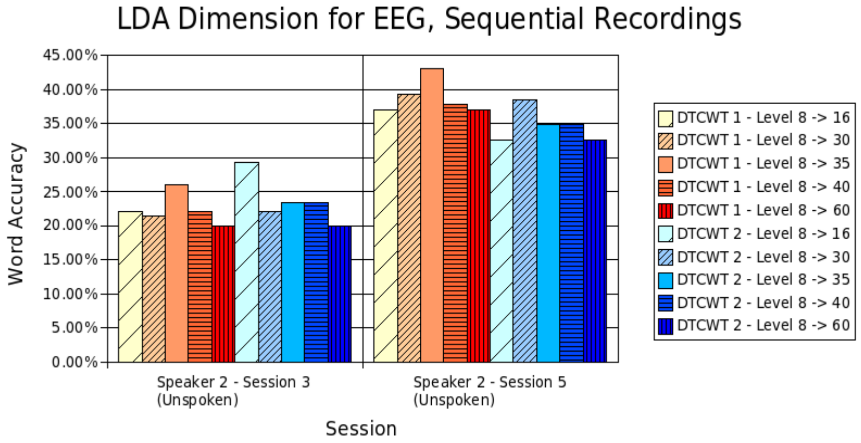

Figure 10: Comparison of LDA Dimensionality for Sequential EEG Recordings of Unspoken Words

[3] conjectures that in the blockwise case, long temporal changes of the EEG signal which are then aligned with the blocks of different words cause the training algorithm to classify not the underlying words, but rather the temporal closeness of utterances.

My results may support this conjecture: The fact that the optimal recognition parameters differ in the two cases suggests that somewhat different things are classified. We see that in the case of sequential recordings, where such a

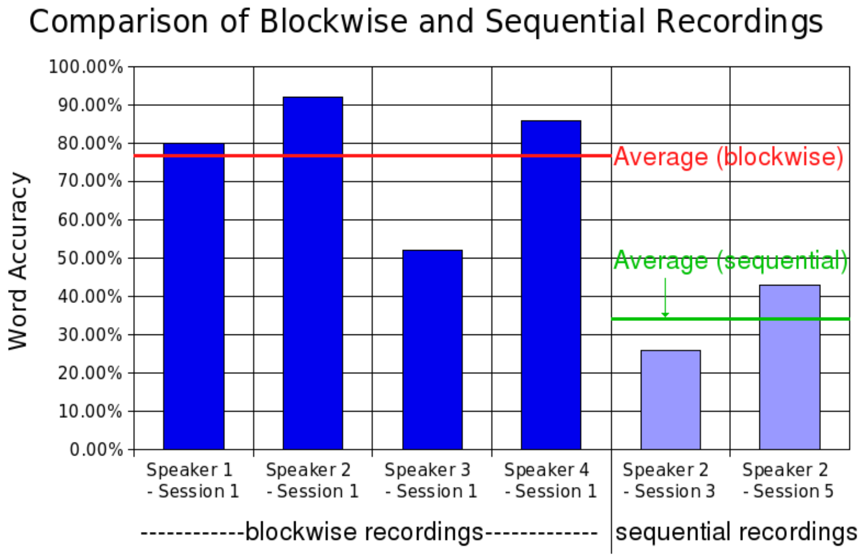

Figure 11: Comparison of Blockwise and Sequential Recordings at their Optimal Parameters

misleading classification cannot take place, we need details and approximation coefficients up to a level of 8. Since our recordings are sampled with a sampling rate of 300 Hz, this means that the length of details covered by coefficients at this level is approximately $2^8/300 \approx 0.85$ seconds. This result is consistent with [18]: In that work, bandpass filters are used to extract the signal of interest for EEG speech recognition, and it is shown that the low-pass limit of these filters often lies in the range of 1 Hz - 5 Hz. As figure 9 shows, the recognition rate first increases with the level of decomposition, and decreases again after the optimal level of approximately eight is reached. The fact that too much data lowers the recognition rate is a well-known phenomenon (which we also see in the LDA tuning) and merits no further explanation.

However, we see that in the blockwise case, the recognition rate is best at very low decomposition levels and sinks already beyond the level three! How is this possible when the upper decomposition levels seem to contain the very data we are trying to classify?

I propose that this shows that data related to the long temporal changes of the EEG signal described above resides in the low decomposition levels, which correspond to high frequencies of brain waves. The additional data in the high decomposition levels seems to contain more artifacts which make recognition difficult—the relatively bad results for the classification of sequential recordings prove this.

It remains a subject for further investigation how exactly information is distributed at different scales in the EEG recordings. The results described so far may hopefully give a grounding for future research on this topic.

4.5 Spoken Words

In this section, we deal with recordings of spoken words. Table 7 and chart 12 depict the results of the evaluation of the optimal decomposition level.

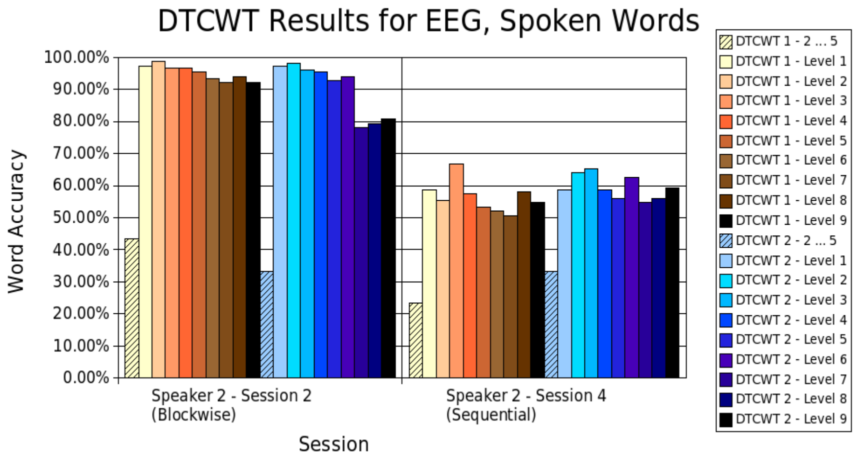

Figure 12: Comparison of DTCWT Decomposition Levels for EEG Recordings of Spoken Words

First of all, and unsurprisingly, the results for the blockwise recording are way better than the results for the sequential recording. However, the relative performance differences between different decomposition levels are very similar:

- The optimal decomposition level is between 2 and 3.
- For the longer filter DTCWT 2 and maximum decomposition level greater than 6, there is a huge reduction in performance. It remains unknown what causes this result.
- Leaving out the coefficients of the first decomposition step (i. e. the ones referring to small details), the performance gets very bad (see the columns marked "DTCWT x - 2 .. 5").

Thus, in the case of spoken words, the difference between blockwise and sequential recordings is *quantitatively* measurable, but there seems to be no *qualitative* difference.

To complete the analysis, I did further experiments with different LDA dimensions for these sessions. The results may be found in table 8 and figure 13.

In the sequential case, the performance decreases for an LDA dimension of 60. In the blockwise case, there is no clear result, the LDA dimension hardly influences the recognition performance at all.

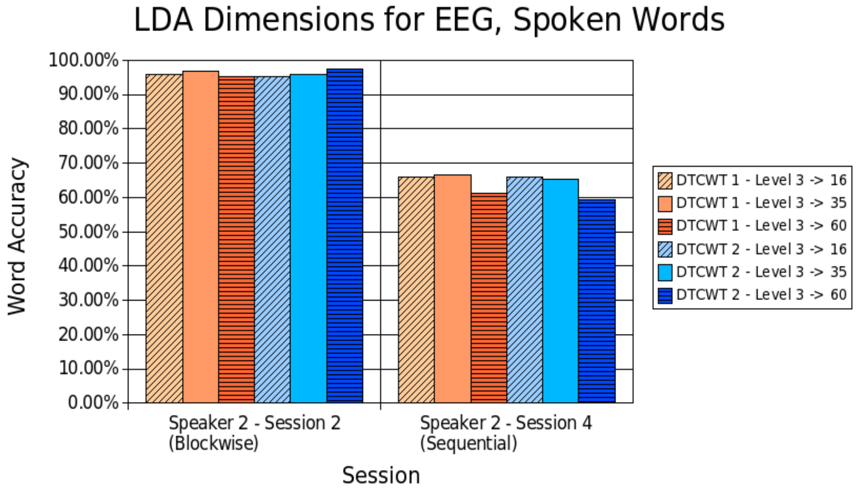

Figure 13: Comparison of LDA Dimensionality for EEG Recordings of Spoken Words

Again, the question is why the results for spoken words differ from those for unspoken words: Blockwise recordings still perform better than sequential ones, but why is the optimal decomposition level for sequential recordings of spoken words relatively low? Since the only difference between spoken and unspoken words is the actual movement of the facial muscles, these artifacts must make the difference. In general, we know that muscular movement generates strong artifacts, indeed we use this to mark the beginning and the end of recordings by eyeblinks.

This naturally leads to considering whether these muscular artifacts may themselves be used for classifying speech. This is the domain of electromyography, which is the second part of this thesis and described quite thoroughly in section 5.

5 Speech Recognition by Electromyography

The fact that classifying EEG recordings of spoken speech proved much more successful than classifying EEG recordings of unspoken speech hints to an influence of the artifacts of the movements of the facial muscles on our recognition task. This inevitably leads to the question whether our preprocessing methods may be applied to electromyographic data as well.

Electromyography (EMG) denotes the measuring of electric signals generated by the human muscles. For the task of speech recognition, we used signals measured on the surface of the human skin which were generated by the articulatory muscles.[5]

[5] This is more accurately described by the term *surface electromyography*, however to keep

On the surface, EEG and EMG speech recognition are unrelated beyond the fact that they both pose a classification task. However, they have got several things in common:

- Both tasks aim at enhancing traditional speech recognition by using not-acoustic information. In particular, they both do not require words to be pronounced loudly, thereby suggesting applications in similar settings. Some of these possible settings are described in section 1.1.
- Both EEG and EMG record electric signals from the human body by means of electrodes.
- In both cases, one of the major obstacles is finding a suitable preprocessing method for this kind of signal.
- As described above, EEG recordings of spoken words may contain artifacts of the movement of the facial muscles.

Since evaluating preprocessing methods was the main goal of my research, this gives a strong motivation to deal with both EEG and EMG recordings. Knowledge gained in one field may easily prove suitable for solving another task.

5.1 Related Work

There have been several attempts on classifying EMG data. Most of these were directed at distinguishing isolated words [7, 11]. The main result of Jorgensen and Binsted [7] is that for word-based recognition, the DTCWT yields better word accuracy than several other classical preprocessing methods, in particular the windowed Fourier transform and the FWT (as defined in section 2). Jorgensen and Binsted achieved a maximum word accuracy of 92%.

More recently, Jou et al. presented their research on a phoneme-based EMG recognition in [8]. The main result of this article is that classical spectral features perform badly in this recognition task. A set of special EMG preprocessing methods is presented which increases the word accuracy to a maximum of 68%.

My EMG experiments were based on the work of Jou et al. In particular, I used exactly the same set of training and evaluation data. Therefore, my results are directly comparable to [8]. In the following sections, I will describe the experimental setup and data acquisition as well as the results of my experiments.

5.2 Experimental Setup

It is a well-known fact that the recognition of EMG recordings is *session-dependent* [11]. This does not only mean that the recognition rates vary between different sessions, but also that the recognition task itself gets much more difficult if data from different sessions is used: the properties of the signal change between sessions, maybe due to a slightly different electrode positioning or a different conductivity of the subject's skin.

the notation brief, I will continue using the word electromyography.

In order to circumvent these problems, I only used data recorded in one session with one speaker. In a quiet room, the speaker read English sentences in a normal tone, which were recorded simultaneously by an EMG recording device and an audio microphone. The recording borders were marked manually by the subject, who pressed a button to start and stop the recording.

The corpus consisted of 38 phonetically balanced sentences and 12 sentences from news articles, each of which was read out 10 times. The 380 phonetically balanced utterances with a total duration of 45.9 minutes were used as the training set, and the 120 news article utterances with a total duration of 10.6 minutes were used for testing. Ten "silence" utterances with a duration of about 5 seconds each were also recorded.

The recording setup for both the audio data and the EMG data is cited after [8]:

> The format of the speech recordings is 16 kHz sampling rate, two bytes per sample, and linear PCM, while it is 600 Hz sampling rate, two bytes per sample, and linear PCM for the EMG signals. The speech was recorded with a Sennheiser HMD 410 close-talking headset.
>
> The EMG signals were recorded with six pairs of Ag/Ag-Cl surface electrodes attached to the skin1, as shown in Fig. 1. Additionally, a common ground reference for the EMG signals is connected via a self-adhesive button electrode placed on the left wrist. The six electrode pairs are positioned in order to pick up the signals of corresponding articulatory muscles: the levator angulis oris (EMG2,3), the zygomaticus major (EMG2,3), the platysma (EMG4), the orbicularis oris (EMG5), the anterior belly of the digastric (EMG1), and the tongue (EMG1,6) [3, 6]. Two of these six channels (EMG2,6) are positioned with a classical bipolar configuration, where a 2cm center-to-center inter-electrode spacing is applied. For the other four channels, one of the electrodes is placed directly on the articulatory muscles while the other electrode is used as a reference attaching to either the nose (EMG1) or to both ears (EMG 3,4,5).
>
> [. . .]
>
> EMG responses were differentially amplified, filtered by a 300 Hz low-pass and a 1 Hz high-pass filter and sampled at 600 Hz.

A Broadcast News speech recognizer trained with the Janus Recognition Toolkit (JRTk) as described in [8] was applied to the audio data. The forced-aligned labels created by this recognizer were used to bootstrap the EMG recognizer. While the audio and EMG recordings were created simultaneously, [8] describes an *anticipatory effect* of the EMG signal, i. e. the EMG signal is ahead of the audio signal. This effect was taken into account by delaying the EMG recordings for a certain time ranging from 0 ms to 90 ms. The recognition rate for different delays varies considerably. Each graphic and each table gives recognition results for each delay.

In [8], several preprocessing methods for EMG are tested, and I use those values as baseline in this paper. They are given in the table below. Note that I use the *Word Accuracy Rate* to measure all EMG results in this paper, whereas [8] gives the *Word Error Rate*.

Preprocessing	Result (Word Accuracy Rate) with optimal delay
Spectral (pure FFT)	13%
Spectral with stacking filter	36%
Optimal EMG feature "*E4*"	68%

5.3 Preprocessing Methods—Wavelets for EMG?

Just as for EEG, the first step is to evaluate in general whether wavelet-based methods may be used for EMG classification. Therefore, on the basis of the corpus described above, I tested four preprocessing setups:

- The RDWT (Redundant Discrete Wavelet Transform) described in section 2.5
- The FWT with a Coiflet-4 filter
- The DTCWT with the 14-tap q-shift filter according to [9]
- A classical windowed Fourier transform (STFT) with 9 subbands.

For all wavelet methods, both detail and approximation coefficients were used according to the experience with EEG recognition, where this proved most useful. The result is found in table 9 in the appendix and charted in figure 14 below.

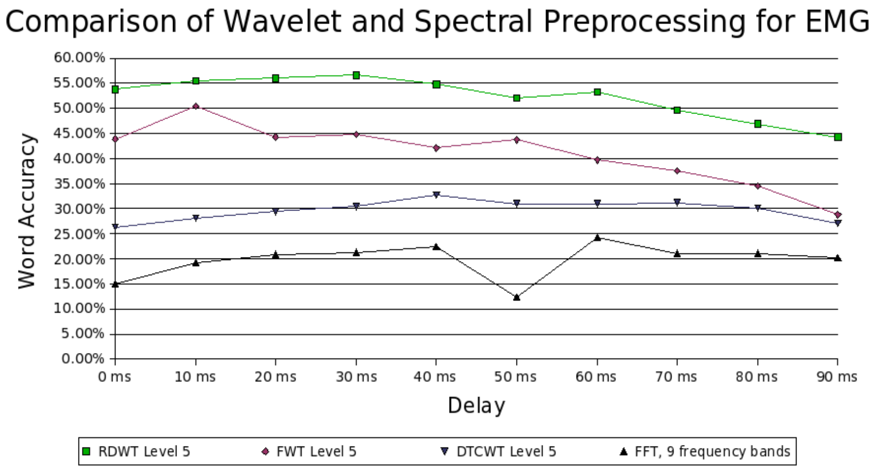

Figure 14: Comparison of Spectral and Wavelet Features for EMG

We see that all three variants of the discrete wavelet transform which we applied performed better than the FFT: The RDWT was best, then comes the FWT and after this the DTCWT. The results are consistent throughout the different delays. The optimal delay depends on the transform used, but does not exceed 40 ms.

The result is encouraging: The best word accuracy rate achieved with the pure RDWT (56.60%) is not very much below the baseline performance (68%) of the specially designed EMG feature E4 from [8]. However, the results are also surprising in that the DTCWT performs wose than both the FWT and the RDWT: In fact, ordering the transforms by their average performance, the results are completely different from the EEG case.

My result also differs from the findings of Jorgensen and Binsted [7], who do not experiment with the RDWT, but assert that the DTCWT performs better than the FWT. The table below shows the preprocessing methods sorted by their performance in different tasks ("$\succ$" signifies "is better than").

Recognition Task	Preprocessing methods ordered by average performance
EEG	DTCWT $\succ$ FWT, STFT $\succ$ RDWT
EMG	RDWT $\succ$ FWT $\succ$ DTCWT $\succ$ STFT
EMG (Jorgensen & Binsted)	DTCWT $\succ$ FWT, STFT

It is beyond the scope of this work to determine what incurs this inconsistency. Clearly, EEG and EMG recognition are not at all the same. It is also noteworthy that the EMG task described in this paper was a *phoneme-based* recognition, whereas Jorgensen and Binsted evaluate *word-based* recognition of EMG recordings. The EEG task described in this paper also is word-based.

However, it is by no means clear that the inconsistency is caused by this difference. Another approach might be to scrutinize the classification algorithm: Jorgensen and Binsted use a neural network, whereas I used a HMM based on context-free phonemes. The HMM topology itself might also have to be adapted to the underlying preprocessing method, see section 6.4 for suggestions on this matter.

5.4 RDWT-based Features

In [8], a major increase in the word accuracy is achieved by means of three *contextual filters* which may be applied to any feature and generate a new feature. In [8], as well as in my research, these filters are used on a preprocessed feature *before* the LDA is applied.

Let $(\mathbf{f}_j)$ be a sequence of feature vectors, where the index $j \in \mathbb{N}$ stands for the time. The filters are the following:

- The *Delta Filter*:

$$D(\mathbf{f}_j) = \mathbf{f}_j - \mathbf{f}_{j-1}$$

- The *Trend Filter*:

$$T(\mathbf{f}_j, k) = \mathbf{f}_{j+k} - \mathbf{f}_{j-k}$$

- The *Stacking Filter*:

$$S(\mathbf{f}_j, k) = \begin{bmatrix} \mathbf{f}_{j-k} \\ \vdots \\ \mathbf{f}_{j+k} \end{bmatrix}$$

So in this section, I describe the results which were yielded by the application of these contextual filters.

To keep consistent with the notation of [8], I define a set of RDWT-based features. In all cases, we start with a sequence of preprocessed feature vectors $X = (\mathbf{x}_j)$ which is the result of a Coiflet-4 RDWT transform till level 5, where both lowpass and highpass coefficients were used (as usual). Thus, each $\mathbf{x}_j$ has got 10 coefficients.

These are the features I used:

$$\begin{aligned}
\mathbf{W0} &= X \\
\mathbf{W1} &= S(X, 1) \\
\mathbf{W2} &= S(X, 5) \\
\mathbf{WD0} &= [X, D(X)] \\
\mathbf{WD1} &= S(\mathbf{WD0}, 1) = S([X, D(X)], 1) \\
\mathbf{WD2} &= S(\mathbf{WD0}, 5) = S([X, D(X)], 5) \\
\mathbf{WT0} &= [X, T(X, 3)] \\
\mathbf{WT1} &= S(\mathbf{WT0}, 1) = S([X, T(X, 3)], 1) \\
\mathbf{WT2} &= S(\mathbf{WT0}, 5) = S([X, T(X, 3)], 5) \\
\mathbf{WDT0} &= [X, D(X), T(X, 3)] \\
\mathbf{WDT1} &= S(\mathbf{WDT0}, 1) = S([X, D(X), T(X, 3)], 1)
\end{aligned}$$

(Since **WDT1** showed worse results than **WDT0**, I refrained from defining a corresponding $\mathbf{WDT2} = S(\mathbf{WDT0}, 5)$ feature).

The results of all these experiments are given in table 11. The charts below emphasize two aspects of these experiments.

At the first run, I solely evaluated the consequences of using the Stacking Filter of the RDWT data. The respective features are **W0**, **W1** and **W2**. The results are seen in figure 15.

The result is not as consistent as expected, but it can be seen that stacking improves the recognition rate at least by several percent points compared to no stacking at all. The result shows that under certain circumstances (which are to be evaluated), too much context information makes recognition results worse. In any case, the best result was yielded by the **W2** feature with a delay of 10 - 20 ms.

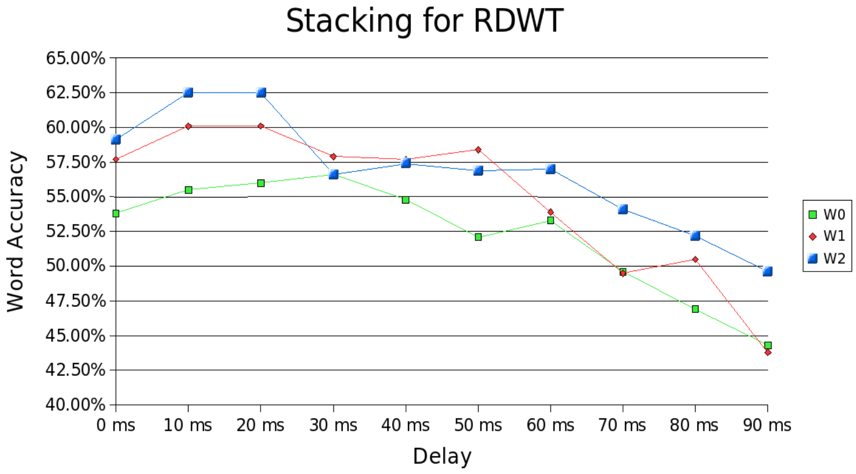

Figure 15: Stacking Filter after the RDWT for EMG Recognition

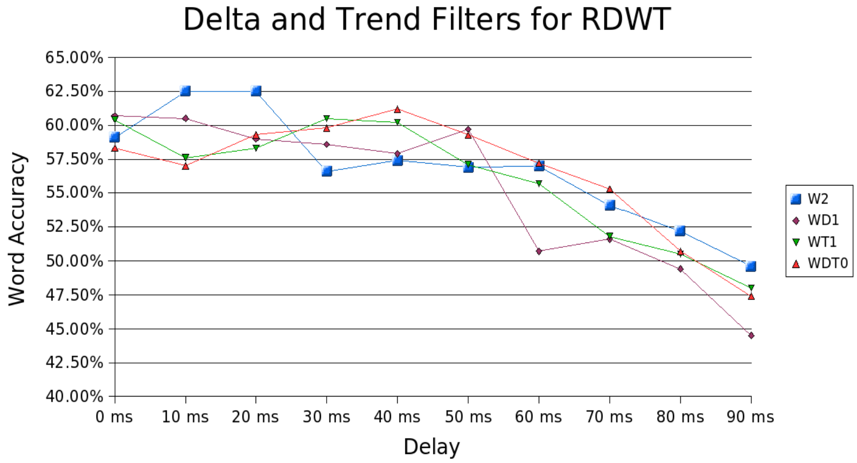

Figure 16: Contextual Filters with Optimal Stacking after the RDWT for EMG Recognition

The next graphic 16 shows the results of applying the Delta and Trend filters separately or together with optimal stacking. The optimal features for this were **W2**, **WD1**, **WT1** and **WDT0**.

Again, **W2** gives the best results with a low, nonzero delay—but not by much. Indeed, it seems that increasing the contextual information initially improves the classification results up to am approximate maximum, but that further increasing the contextual information actually makes the results worse, where the exact nature of the context filters *does not greatly influence the results.*

5.5 Special EMG Features and the Wavelet Transform

We have seen that the recognition rate with RDWT and context information alone may not be increased without limits. Thus at last, we deal with extensions of the "special EMG features" from [8]. In that paper, the original signal is split into a low-pass part and a high-pass part. The definition is as follows: For the original signal x_j, the *nine-point double-averaged signal* w_j is defined as

$$w_j = \frac{1}{9} \sum_{n=-4}^{4} v_{j+n}, \qquad \text{where} \qquad v_j = \frac{1}{9} \sum_{n=-4}^{4} x_{j+n}.$$

Then, the *high-frequency signal* p_n is defined as

$$p_j = x_j - w_j$$

and its rectified version is

$$r_j = |x_j - w_j|.$$

This split signal is processed frame-based, where the width of a frame is 27ms and the frame shift is 10ms. In the following formulas, the time indices 0 and J represent the beginning and end of a frame, respectively.

For both w_j and r_j, the frame-based power and average are taken: For any frame, it is defined

$$\bar{w} = \frac{1}{J} \sum_{j=0}^{J} w_j \qquad \text{and} \qquad P_w = \sum_{j=0}^{J} |w_j|^2.$$

and similarly

$$\bar{r} = \frac{1}{J} \sum_{j=0}^{J} r_j \qquad \text{and} \qquad P_r = \sum_{j=0}^{J} |r_j|^2.$$

In addition, z is defined to be the zero-crossing count of the high-frequency signal $p_0, \dots, p_J$. All of these features are normalized.

Finally, the definition of **E4** is

$$\mathbf{E4} = S([P_w, P_r, \bar{w}, \bar{r}, z], 5).$$

Eventually, LDA is applied to the feature with the usual dimensionality reduction to 32.

My approach is using this processing (i. e. the calculation of power, average and zero-crossing rate) on RDWT-preprocessed data. This means that we do a RDWT transform based on a Coiflet-4 wavelet as usual, this yielding several data rows $\rho^{(k)}$, where each row corresponds to the high-pass or low-pass coefficients on one specific scale. Then for each row, we calculate the frame-based power $P_{\rho^{(k)}}$, the average $\bar{\rho}^{(k)}$ and the zero-crossing rate $z_{\rho^{(k)}}$ with the same definition of "frame" as above.

The difference between the features lies in the RDWT preprocessing, and this difference yields some important insights into the structure of the EMG data. In particular, for some of these features, *only* the highpass coefficients of the Wavelet Transform were used.[6] Those features are marked with “HP only”, whereas features where both highpass and lowpass coefficients were used are marked “LP/HP”. The features are as follows

X1 = RDWT Level 2, LP/HP - 4 rows per channel
X2 = RDWT Level 2, HP only - 2 rows per channel
X3 = RDWT Level 4, HP only - 4 rows per channel
X4 = RDWT Level 5, HP only - 5 rows per channel
X5 = RDWT Level 5, LP/HP - 10 rows per channel.

The results are charted in figure 17, the numbers are found in table 12.

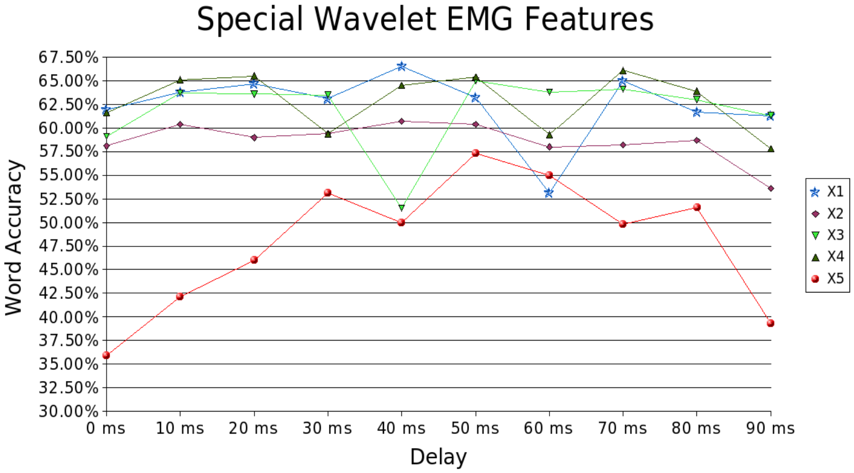

Figure 17: Special Wavelet EMG Features

The results of these experiments are inconsistent across the different delays. However, it can be said that the best result of my EMG experiments was achieved with the **X1** feature at a delay of 40 ms. The recognition rate of 66.50% is very close to the optimal recognition rate in [8], which was 68%. We also see that the features **X2** and **X5** generally perform worse than the other features, whereas **X1**, **X3** and **X4** are quite equal in terms of the recognition rate. The most striking fact is that in the case of an RDWT to level 2, the highpass coefficients increase the recognition rate, whereas for an RDWT till level 5, they significantly reduce the performance.

This fact may be explained by the same reasoning as in the section above—that there is a certain level of “saturation” which the RDWT features may

[6] This means that some information was irretrievably lost, since the reconstruction of the original data requires the highpass (wavelet) coefficients and the lowest-scale approximation coefficient.

reach or exceed, and that further information reduces the recognition rate. But further scrutiny might also give hints to the location of relevant information in the different scales the original EMG data is split into.

5.6 Summary

By these experiments, we have seen that the Redundant Discrete Wavelet Transform (RDWT) is a powerful tool in preprocessing EMG signals for speech reocgnition. Even a simple RDWT with no further processing significantly outperforms ordinary spectral features. [8] asserts that the pure FFT is inadequate for this kind of processing, so it is remarkable that the RDWT performs so much better.

I did not succeed for now in outperforming the best results of [8], where the **E4** feature yielded a word error rate of 32%, i. e. a word accuracy of 68%. However, a combination of RDWT and the smoothing applied in [8] came very close to those results.

The main point, however, is that we now have a lever to better understanding the nature of the EMG signal as well as the preprocessing behavior. For example, the fact that context-based features improve the recognition results up to a certain limit *where the exact nature of these features plays a minor role* gives us hints where to direct further research (see also section 6.3).

6 Conclusion

In this section, I will lay out which conclusions may be drawn from the experiments described above.

6.1 The DTCWT is Feasible for EEG Recognition

This work shows, in general, that the DTCWT is a good lever for increasing the performance of EEG-based speech recognition. It must, however, also be noted that there apparently were "good" and "bad" recordings in terms of recognition performance, i. e. a recording which yielded bad results under a standard Fourier preprocessing (or indeed a FWT-based preprocessing) would not yield excellent results when using the DTCWT. In other words, the DTCWT performs better than the other preprocessing methods, but is not more or less robust than those.

If this effect is due to artifacts in the recording, more refined methods of artifact removal may solve this problem; if the problem is caused by a generally poor signal (for example because of badly placed electrodes, bad connectivity of the skin at the time of recording or lack of concentration on the side of the subject), then a change of the preprocessing method will not be helpful, instead the recording methods would have to be reevaluated.

I tried two different filter setups for the DTCWT. Based on my experiments, none of these setups is best in all circumstances—the differences between these filters only amount to a few percent points, the decomposition level has got a much greater influence on the result.

Nonetheless, given a particular task, it may be suitable to experiment with different filter setups (there are way more possibilities than I employed here) in order to obtain best results. Filters can also be designed specifically for a task at hand ([16] and the references therein).

Finally, an LDA should always be applied, with a dimensionality reduction to approximately 35 dimensions. Beyond this, the recognizer performance suffers.

6.2 Comparison of EEG Recordings of Different Modalities

Based on my experiments, one cannot say that under all circumstances, one specific filter setup gives best results. However, it has turned out (as described in the Results section) that the recordings may be grouped according to their modality and sequencing mode, and that for recordings which fall into the same group, similar properties hold.

Before discussing each group separately, I give here a comparison of the average performance levels for each combination of modalities:

	Blockwise	**Sequential**	**ø**
Unspoken	77.46%	36.15%	56.81%
Spoken	98.67%	66.67%	82.67%
ø	88.07%	51.41%	69.74%

6.2.1 Spoken Words Are Recognized Better than Unspoken Words

The most striking result is that the performance of the EEG recognizer for *Spoken Language* is significantly better than for unspoken language. This is a result which remains unchanged throughout all recordings, and in some cases even if recordings of spoken and unspoken speech were made during one single session (i. e. without the ElectroCap being removed in-between), the performance for spoken words is superior to the performance for unspoken words. In particular, this can be seen with the sessions 1 and 2 (blockwise) as well as 3 and 4 (sequential) of speaker 2. I chart this effect in figure 18.

Since all other causes can be ruled out, the difference in speech modality must be the influential factor for this gap. This gap exists no matter whether the recording sequence mode was blockwise or sequential.

6.2.2 Blockwise Recordings are Better than Sequential Recordings

A similarly striking result is that blockwise recordings perform much better than sequential ones, *but only in the case of unspoken words.* Again, this even occurs when the two recordings to be compared were done as a “Double Session”, i. e. without removing the ElectroCap in-between.

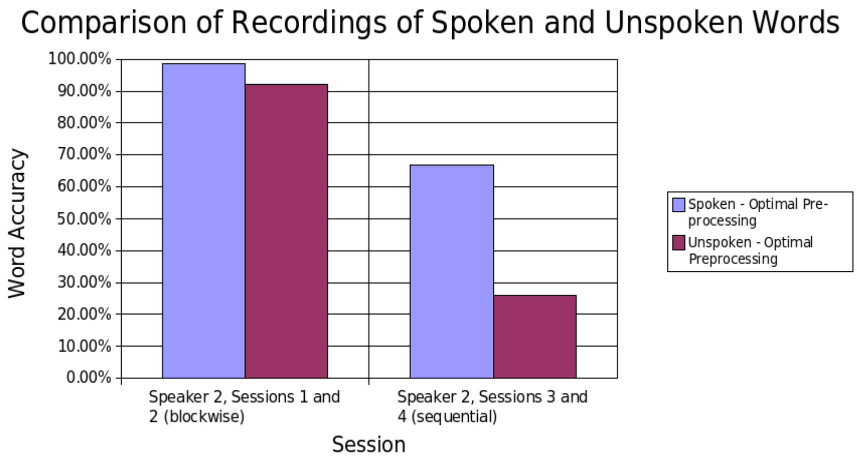

Figure 18: Comparison of Recordings of Spoken and Unspoken Words

This has got far-reaching consequences: First of all, it means that during evaluations of any EEG recognizer, the sequence of the recorded data *must* be observed in order to get comparable results.

Mainly, however, it means that in blockwise mode, a kind of artifact related to this mode enters classification and improves recognition performance, which implies that this artifact changes in accordance to the currently recorded block.

One explanation could be that a word which is recorded multiple times in a row gets increasingly "fixed" in the subject's mind, so that the thought which we are trying to recognize becomes increasingly stronger and more recognizable. However, since between the recording of two samples, several seconds pass (at least), and before each recording the current word is shown to the subject, this seems not too convincing. The fact that the words do not have connotations for the subject further reduces this probability.

The most probable explanation is that the EEG recognizer classifies a slow temporal change of the overall shape of the EEG—independent of the exact word which was thought at a point in time. Indeed, I took a *sequential* recording, rearranged the samples in temporal order and relabeled them accordingly:

- The first 30 samples got the label "First".
- Samples 31-60 got the label "Second".
- Samples 61-90 got the label "Third".
- Samples 91-120 got the label "Fourth".
- Samples 121-150 got the label "Fifth".

Using the EEG recognizer on these relabeled sessions yielded results comparable to those for blockwise recordings, even though in these cases, the classification could *not* be based on the actual words that were being thought.

As I outlined in section 4.5, for spoken words the results for blockwise recordings are still better than those for sequential recordings, but even in the latter case the performance is significantly above chance. Further conclusions of this fact are discussed in the next section.

6.2.3 Conclusions by Modality

Comparing the different modalities and sequencing modes, the following results can be inferred: For **blockwise recordings of unspoken words**, the optimal decomposition level of the DTCWT is very low, between 2 and 3.

For **sequential recordings of unspoken words**, the optimal decomposition level is about 8.

The difference between these results must be rooted in the time-related artifacts entering the classification—a conjecture already posed in [3] which my results confirm. These artifacts seem to get weaker when the decomposition level increases, i. e. details of larger scales (lower frequencies) enter the feature vectors. At the same time, we see that information about a currently thought word is located in the lower frequencies which correspond to decomposition levels around 8. At a sampling rate of 300 Hz, decomposition level 8 represents details of maximal size $2^8/300 = 0.85$ seconds, which corresponds to a frequency of about 1.17 Hz. This result is consistent with previous work on EEG word recognition: Suppes et al. found that ideal bandpass filters for a recognition task similar to this one had low frequencies usually ranging from 1 Hz to 5 Hz ([18], table 2).

For **spoken words**, the optimal decomposition level was again relatively low, namely around 3. More importantly, sequential recordings of spoken words behaved similar to blockwise recordings of spoken words and showed a much better performance than sequential recordings of unspoken words. It is clear that this is related to the presence of muscular artifacts in the recordings, though further research is needed to show whether the actual muscular movement or the commands sent by the brain to the muscles make the difference.

6.3 The RDWT is Feasible for EMG Recordings

We have seen that the RDWT significantly outperforms all spectral features in the task of EMG recognition. However, we also have seen that the performance of the "special EMG features" described in [8] is not easily exceeded. It is particularly noteworthy that the context filters D, T and S from section 5.4 increase the word accuracy up to a certain limit, but not further. It remains a question for future research why this happens.

6.4 Ideas fur Further Research

6.4.1 EEG

So far, the work in EEG speech recognition at InterACT [19, 3] has laid an important cornerstone for future research. In [19], Marek Wester created a working recognizer for EEG data and applied it to data based on several different corpora, recorded with various modalities. It turned out that the choice of the corpus does not measurably influence recognition results.

However, the recording modality greatly influences the recognition performance. Wester uses Normal, Silent, Mumbled and Whispered Speech (all incurring facial muscle movements) as well as Unspoken Speech. The fact that in the case of Unspoken Speech, the original recording setup proved susceptible to artifacts related to the temporal closeness of utterances is one of the major results of Jan Calliess in [3].

The next step must be to eliminate this susceptibility in the case of Unspoken Speech, which is probably most interesting due to the absence of muscular artifacts. My own work as well as [18] affirm that word recognition by EEG is well possible, but needs even more refined methods.

One major hindrance I encountered was the lack of session data containing a sufficient number of utterances in sequential (or random) ordering.[7] [18] use 700 utterances in random ordering for each recognition task, in my work I was limited to maximally 150 utterances. In order to be able to sufficiently generalize results, more recordings particularly of unspoken speech, restricted to one or two of the corpora mentioned in section 3.1 should be taken. Early experiments, which I do not describe in detail in this paper, indicate that an increase in the number of training utterances greatly improves the word accuracy.

At this point, it should also be mentioned that due to the smallness of the recording session data, we did not use independent test/crossvalidation sets, but rather tuned our preprocessing to the same data on which it was tested. The "Leave-One-Out" method we used (described in section 3.4) may therefore have influenced the genericity of the results.

When a suitable corpus of training, testing and evaluation data has been created, the preprocessing methods must be refined. The following ideas come to mind:

- Test a variety of different filters instead of only two ones
- Leave out certain scales (i. e. decomposition levels) from the feature vectors in order to exactly determine at which scales relevant information is found. Compare this with spectral methods.
- Implement and try the Wavelet Packet Transform [4].
- Try context-based features like the ones described in the EMG section 5.4.

6.4.2 EMG

We have seen that context information is very important for EMG recognition. It would be a natural continuation of this work to introduce phonemes including context (i. e. polyphones). However, this requires a large body of data, which is

[7] The time limit imposed on my work made it unfeasible to do recording sessions on my own.

currently difficult to obtain because of the session-dependency of EMG recordings. Finding a way to do EMG recognition across different recording sessions, maybe even across different persons, would be a great step forward.

The good results achieved with the RDWT can give us further hints on how to refine the EMG preprocessing. The results should be cross-checked with different data; this might empower us to answer the question how EMG data is structured. My special EMG features somewhat indiscriminately computed three variables (power, mean and zero-crossing-rate) for each row created in the RDWT processing. Refining this method might improve the results.

Beyond the mere preprocessing, it might also prove valuable to scrutinize the classification algorithm. Even when sticking to a left-to-right HMM model, the number of states and the transition weights might have to be adapted. The caveat is best shown by example: The original frame-based computation of feature vectors in [8] created 100 feature vectors per second (because the frame shift was 10 ms), the RDWT creates about 600 feature vectors per second (one for each recorded value, and the sampling rate was 600 Hz). Tuning the HMM to reflect this difference might give a better model of the EMG signal and therefore a better recognition rate.

Appendices

A Software Documentation

This section is intended to serve as a guide to those who want to continue experiments in EEG recognition at InterACT. It describes the software used for the recognition process and outlines the required configuration as well as the process which has to be initiated by the user.

My description is based on the environment I used for most of my work at the Interactive Systems Lab at CMU. Given paths refer to the common NFS space managed by the lab.

A.1 JANUS

Janus is the state-of-the-art speech recognizer developed by the CMU and the University of Karlsruhe. It provides a complete framework for tasks in the field of signal processing and pattern recognition based on the most recent techniques used in speech recognition research. A multitude of algorithms and all building blocks for a speech recognizer system are available, e. g. HMM models, decision trees, path algorithms, Fourier transforms etc.

JANUS has been implemented in two parts: A core containing time-consuming algorithms has been written in plain C. This core is accessed by a set of TCL functions found in the *tcl-lib*. A documentation of this structure is available. Several routines are written in TK to provide a graphical user interface, however, there are not necessary for the recognition task. In fact, since actual recognition tasks are run without direct user supervision, the use of a GUI would be contraproductive, and therefore the GUI part is often left out when compiling JANUS. For details refer to the JANUS documentation.

The functions from the C core mostly appear in TCL as methods of pseudo-classes. All wavelet transform functions are part of the FeatureSet class.

A.2 The Wavelet Transform in JANUS

I implemented the Wavelet Transform in a separate codebase of the JANUS core. This codebase is to be merged with the main JANUS branch in the future.

The implementation of the wavelet transform variants may be found in the file `$JANUSBASE/src/features/featureDWT.c`[8]and its respective header file. The implementation has got three access functions, each of which has got a C name and a JANUS name.

Function in C	Function in JANUS	Description
`fesFWTItf`	`fwt`	Performs the FWT
`fesDTCWTItf`	`dtcwt`	Performs the DTCWT
`fesRDWTItf`	`rdwt`	Performs the RDWT

[8] In my setup, `$JANUSBASE` is `/people/mwand0/ibis-014`.

All these functions take the following parameters (in this order):

Parameter	**Meaning**
`TO`	The name of the feature which will be newly created.
`FROM`	The name of the feature which has to be processed.
`Filter`	The filter(s) which will be used. See below.
`Level`	The maximum decomposition level.
`-useLowpass L`	(Optional) The number of low-pass coefficients which should go into the new feature. Must be smaller than `Level`, defaults to 0.

The parameter `Filter` takes the following form: For the FWT and the RDWT, it is one single `Filter` object. For the DTCWT, four filters must be passed: First the two low-pass filters for the first stage of the transform, then the two low-pass filters for the second stage. Refer to section 2.4 for details.

These functions appear as member function of the `FeatureSet` class, so for performing e. g. the FWT, one needs to use the following code. Here we assume that `fs` is a `FeatureSet` having a feature `ADC`. `WV` will be the new feature containing the wavelet coefficients. `myLevel` must be a positive integer. For a more complete example see below. `myFilter` can be a filter object or an expression like { `offset` { `...coefficients ...`} }.

```
fs fwt WV ADC myFilter $myLevel [-useLowpass $myLevel]
```

A.3 The Training and Recognition Process

In the current setup, (almost) all files relevant for the EEG recognition are found under `/people/mwand0/cluster`, and any user wishing to use them should copy the entire directory to his/her own NFS space. Before actually starting the work, the following setup has to be done:

- These shell variables have to be set *appropiately.* I give the commands here in csh syntax:

  ```
  setenv CLUSTER_DIR /people/mwand0/cluster
  setenv JANUSNX /people/mwand0/cluster/bin/janusNX
  ```

  The right place for this would be the `~/.cshrc` if csh is used.

- Apart from the above, several base directories must be set in `$CLUSTER_DIR/mainconfig.tcl` *and* `$CLUSTER_DIR/mainconfig.csh` (this is due to the fact that the EEG recognizer contains both csh and TCL code).

- Here comes an example for a `~/.janusrc`:

  ```
  puts "janusrc executed by [info nameofexecutable]"
  puts "at [clock format [clock seconds]]"
  ```

```
set base "/people/mwand0/ibis-014"

set TclLibraryPath $base/tcl-lib
set JANUSLIB $base/gui-tcl
lappend auto_path $base/library $base/tcl-lib $base/gui-tcl

# This is the only nonstandard part, and only necessary
# if any filter in one of these files
# will be used. filters.tcl defines some basic filters.
source /people/mwand0/library/filters.tcl
source /people/mwand0/library/tcltools.tcl
```

The structure of the `$CLUSTER_DIR` directory tree is as follows:

`bin` contains the necessary binaries.

`config` collects all configuration data necessary for a training job. As soon as the training job has been started, these files may be changed to perpare another job. The meaning of the files will be explained below.

`eeg_recognizer` contains an EEG recognizer. This directory serves as a template for actual tasks.

`pool` is where the work is done—this directory contains adapted copies of the `eeg_recognizer` directory.

`tools` contains additional TCL scripts.

For a training run, first of all the job database containing all adc files[9], their labeling etc. has to be prepared. The main difficulty in this is to determine the signal bounds, which according to section 3.2 are marked by eyeblinks.

When this has been done, a subdirectory of `pool` is created which contains all files which the recognizer uses. This subdirectory has got the default name `SPEAKERID_SESSIONID_MODALITY`, for example `13_01_think`.

The JANUS process is highly parallelized. At CMU, it may run on a Condor cluster [1]. The process will only access files inside the respective subdirectory of `pool`. The file controlling the entire following process is `$CLUSTER_DIR/master.tcl`. It may be called with several options, but only three of them are usually needed. These I will describe in the following list.

A full training run works like this (paths are given assuming `$CLUSTER_DIR` is the current directory):

- Edit `config/dbaseparam.tcl` appropiately. Set the data folder which should be worked with, the vocabulary and the speech modality. The variable `offsetForRecordings` gives the amount of space at the beginning

[9] This is the format for recorded data

of a recording which should be ignored by the eyeblink recognizer. Two seconds seems to be a good choice to avoid errors which may stem from irregularities at the beginning of the recording.

- The job database is created then by a call to

  ```
  bin/janusNX tools/prepareDatabse.tcl
  ```

  This database is saved to `eeg_recognizer/desc/janus/db-utt.⋆`.

- Edit `config/baseDesc.tcl` and `config/featDesc_eeg.tcl` according to the desired preprocessing. Make sure that in `config/baseDesc.tcl`, `${SID}(codebookDimN)` is set to the correct dimensionality of the feature vector before the LDA application.

- For building the subdirectory of pool, call

  ```
  bin/janusNX master.tcl build <directory> -domain <dom>
  ```

  where `<directory>` will have a value like `13_01_think`, and `<dom>` represents the vocabulary, like `alpha` or `digit5` (for a full list, see the source code of `master.tcl`).

- When this is done, you may optionally rename the subdirectory in order to accomodate several runs of JANUS on the same data, or to represent specific settings. However, the name or this directory *must* be set both in `pool/.../condorDesc` and `pool/.../desc/baseDesc.tcl`.

- Start the recognition run by calling

  ```
  bin/janusNX master.tcl start <directory>
  ```

  where `<directory>` should be e. g. `13_01_think`, as above. This command returns immediately, and Condor takes control over the process. You must, however, make sure that a Condor scheduler is installed on your computer.

- Use the standard Condor command `condor_q` to check the status of your task. The result should look approximately like this:

  ```
  -- Submitter: proconsul.is.cs.cmu.edu: <128.2.219.46:32775>: proconsul.is.cs.cmu.edu
   ID      OWNER            SUBMITTED     RUN_TIME ST PRI SIZE CMD
  3590.0   mwand          12/1  09:02   0+00:00:00 I  0   1.9  janusNX DO.crossVa
  ```

  The important thing here is that the basic script to be called is `DO.crossValidation.tcl`. I recommend reading it to see how the process works in detail.

  Refer to the Condor documentation for more information. `master.tcl` also incorporates some Condor handling.

- When `condor_q` does not show the JANUS process any more, it is finished, and the results may be found in the directory `pool/.../log/`. If everything ran smoothly, the file `condor.out` contains the overall recognition rate at the end. The results may be represented as a confusion matrix by the call `bin/janusNX master.tcl results directory`.

B Detailed Results of the Experiments

B.1 EEG Recognition Results

These are the results from my EEG experiments. See section 4 for an explanation and the corresponding charts.

	Speaker 1	**Speaker 2**	**Speaker 2**	**Speaker 2**
	Session 1	**Session 1**	**Session 5**	**Session 2**
	Unspoken	*Unspoken*	*Unspoken*	*Spoken*
	blockwise	*blockwise*	*sequential*	*blockwise*
STFT - 12 subbands	64.00%	76.00%	36.30%	83.33%
DTCWT 1 - Level 4	77.34%	88.00%	39.26%	96.67%
DTCWT 2 - Level 4	70.00%	90.00%	38.52%	95.33%
FWT with Daub4 - Level 4	31.33%	76.00%	34.81%	92.00%
RDWT with Daub4 - Level 4	33.33%	54.00%	31.11%	86.00%

	Speaker 2	**Speaker 3**	**Speaker 4**
	Session 4	**Session 1**	**Session 1**
	Spoken	*Unspoken*	*Unspoken*
	sequential	*blockwise*	*blockwise*
STFT - 12 subbands	52.67%	30.08%	44.97%
DTCWT 1 - Level 4	57.33%	42.67%	77.33%
DTCWT 2 - Level 4	58.67%	34.13%	69.33%
FWT with Daub4 - Level 4	50.00%	27.73%	52.30%
RDWT with Daub4 - Level 4	46.67%	23.73%	47.00%

Table 2: Preprocessing Results for EEG

	Speaker 1	**Speaker 2**	**Speaker 3**	**Speaker 4**
	Session 1	**Session 1**	**Session 1**	**Session 1**
DTCWT 1 - 2 ... 5	70.67%	77.33%	19.51%	29.53%
DTCWT 1 - Level 1	71.33%	88.67%	39.47%	85.83%
DTCWT 1 - Level 2	76.67%	87.33%	52.00%	84.00%
DTCWT 1 - Level 3	80.00%	88.67%	49.07%	82.00%
DTCWT 1 - Level 4	77.34%	88.00%	42.67%	77.33%
DTCWT 1 - Level 5	70.00%	84.67%	38.21%	69.13%
DTCWT 1 - Level 6	62.67%	85.33%	38.13%	71.33%
DTCWT 1 - Level 7	50.67%	78.00%	43.09%	60.40%
DTCWT 1 - Level 8	48.00%	78.00%	43.90%	31.54%
DTCWT 1 - Level 9	50.67%	78.67%	36.53%	44.63%
DTCWT 2 - 2 ... 5	30.67%	53.33%	18.70%	38.26%
DTCWT 2 - Level 1	71.33%	88.67%	39.47%	85.83%
DTCWT 2 - Level 2	76.67%	82.67%	46.13%	85.33%
DTCWT 2 - Level 3	72.67%	92.00%	44.00%	76.50%
DTCWT 2 - Level 4	70.00%	90.00%	34.13%	69.33%
DTCWT 2 - Level 5	62.00%	88.00%	39.02%	67.11%
DTCWT 2 - Level 6	57.30%	81.33%	32.53%	64.00%
DTCWT 2 - Level 7	58.67%	80.00%	20.33%	30.20%
DTCWT 2 - Level 8	58.67%	79.33%	20.23%	31.54%
DTCWT 2 - Level 9	50.67%	75.33%	19.73%	32.21%

Table 3: DTCWT Results for EEG, Blockwise Recordings

	Speaker 1	**Speaker 2**	**Speaker 3**	**Speaker 4**
	Session 1	**Session 1**	**Session 1**	**Session 1**
DTCWT 1 - Level 3 → 16	80.00%	88.00%	41.33%	80.67%
DTCWT 1 - Level 3 → 35	80.00%	88.67%	49.07%	82.00%
DTCWT 1 - Level 3 → 60	75.30%	86.00%	44.00%	75.33%
DTCWT 2 - Level 3 → 16	76.67%	91.33%	45.07%	76.67%
DTCWT 2 - Level 3 → 35	72.67%	92.00%	44.00%	76.50%
DTCWT 2 - Level 3 → 60	70.67%	87.33%	40.53%	71.17%

Table 4: LDA Dimension for EEG, Blockwise Recordings

	Speaker 2	**Speaker 2**
	Session 3	**Session 5**
DTCWT 1 - 2 ... 5	17.33%	20.74%
DTCWT 1 - Level 3	12.67%	33.33%
DTCWT 1 - Level 4	11.34%	39.26%
DTCWT 1 - Level 5	18.00%	31.11%
DTCWT 1 - Level 6	19.33%	34.81%
DTCWT 1 - Level 7	22.67%	31.85%
DTCWT 1 - Level 8	26.00%	42.96%
DTCWT 1 - Level 9	20.00%	39.26%
DTCWT 1 - Level 10	18.67%	40.00%
DTCWT 2 - 2 ... 5	21.33%	18.52%
DTCWT 2 - Level 3	16.00%	29.63%
DTCWT 2 - Level 4	18.67%	38.52%
DTCWT 2 - Level 5	18.00%	30.37%
DTCWT 2 - Level 6	20.67%	30.37%
DTCWT 2 - Level 7	21.33%	36.30%
DTCWT 2 - Level 8	23.33%	34.81%
DTCWT 2 - Level 9	22.00%	40.00%
DTCWT 2 - Level 10	15.33%	32.59%

Table 5: DTCWT Results for EEG, Sequential Recordings

	Speaker 2	**Speaker 2**
	Session 3	**Session 5**
DTCWT 1 - Level 8 → 16	22.00%	37.04%
DTCWT 1 - Level 8 → 30	21.33%	39.26%
DTCWT 1 - Level 8 → 35	26.00%	42.96%
DTCWT 1 - Level 8 → 40	22.00%	37.78%
DTCWT 1 - Level 8 → 60	20.00%	37.04%
DTCWT 2 - Level 8 → 16	29.33%	32.59%
DTCWT 2 - Level 8 → 30	22.00%	38.52%
DTCWT 2 - Level 8 → 35	23.33%	34.81%
DTCWT 2 - Level 8 → 40	23.33%	34.81%
DTCWT 2 - Level 8 → 60	20.00%	32.59%

Table 6: LDA Dimension for EEG, Sequential Recordings

	Speaker 2	**Speaker 2**
	Session 2	**Session 4**
DTCWT 1 - 2 ... 5	43.33%	23.33%
DTCWT 1 - Level 1	97.33%	58.67%
DTCWT 1 - Level 2	98.67%	55.33%
DTCWT 1 - Level 3	96.67%	66.67%
DTCWT 1 - Level 4	96.67%	57.33%
DTCWT 1 - Level 5	95.33%	53.33%
DTCWT 1 - Level 6	93.33%	52.00%
DTCWT 1 - Level 7	92.00%	50.67%
DTCWT 1 - Level 8	94.00%	58.00%
DTCWT 1 - Level 9	92.00%	54.67%
DTCWT 2 - 2 ... 5	33.33%	33.33%
DTCWT 2 - Level 1	97.33%	58.67%
DTCWT 2 - Level 2	98.00%	64.00%
DTCWT 2 - Level 3	96.00%	65.33%
DTCWT 2 - Level 4	95.33%	58.67%
DTCWT 2 - Level 5	92.67%	56.00%
DTCWT 2 - Level 6	94.00%	62.67%
DTCWT 2 - Level 7	78.00%	54.67%
DTCWT 2 - Level 8	79.33%	56.00%
DTCWT 2 - Level 9	80.67%	59.33%

Table 7: DTCWT Results for EEG, Spoken Words

	Speaker 2	**Speaker 2**
	Session 2	**Session 4**
DTCWT 1 - Level 3 → 16	96.00%	66.00%
DTCWT 1 - Level 3 → 35	96.67%	66.67%
DTCWT 1 - Level 3 → 60	95.33%	61.33%
DTCWT 2 - Level 3 → 16	95.33%	66.00%
DTCWT 2 - Level 3 → 35	96.00%	65.33%
DTCWT 2 - Level 3 → 60	97.33%	59.33%

Table 8: LDA Results for EEG, Spoken Words

B.2 EMG Recognition Results

Delay	0 ms	10 ms	20 ms	30 ms	40 ms	50 ms	60 ms	70 ms	80 ms	90 ms	AVG
RDWT Level 5	53.80%	55.50%	56.00%	56.60%	54.80%	52.10%	53.30%	49.60%	46.90%	44.30%	52.29%
FWT Level 5	43.90%	50.40%	44.30%	44.80%	42.10%	43.70%	39.70%	37.60%	34.50%	28.80%	40.98%
DTCWT Level 5	26.20%	28.10%	29.40%	30.40%	32.70%	31.00%	31.00%	31.20%	30.10%	27.10%	29.72%
FFT, 9 frequency bands	15.00%	19.30%	20.90%	21.30%	22.40%	12.30%	24.20%	21.10%	21.10%	20.20%	19.78%

Table 9: Comparison of Spectral and Wavelet Features for EMG

Delay	**0 ms**	**10 ms**	**20 ms**	**30 ms**	**40 ms**	**50 ms**	**60 ms**	**70 ms**	**80 ms**	**90 ms**	**AVG**
W0	53.80%	55.50%	56.00%	56.60%	54.80%	52.10%	53.30%	49.60%	46.90%	44.30%	52.29%
W1	57.70%	60.10%	60.10%	57.90%	57.70%	58.40%	53.90%	49.50%	50.50%	43.80%	54.96%
W2	59.10%	62.50%	62.50%	56.60%	57.40%	56.90%	57.00%	54.10%	52.20%	49.60%	56.79%
W3	56.90%	59.30%	57.40%	59.10%	58.90%	54.90%	54.20%	52.40%	48.70%	45.10%	54.69%
W4	60.70%	60.50%	59.00%	58.60%	57.90%	59.70%	50.70%	51.60%	49.40%	44.50%	55.26%

Table 10: RDWT with Delta and Stacking Filters

Delay	**0 ms**	**10 ms**	**20 ms**	**30 ms**	**40 ms**	**50 ms**	**60 ms**	**70 ms**	**80 ms**	**90 ms**	**AVG**
W0	53.80%	55.50%	56.00%	56.60%	54.80%	52.10%	53.30%	49.60%	46.90%	44.30%	52.29%
W1	57.70%	60.10%	60.10%	57.90%	57.70%	58.40%	53.90%	49.50%	50.50%	43.80%	54.96%
W2	59.10%	62.50%	62.50%	56.60%	57.40%	56.90%	57.00%	54.10%	52.20%	49.60%	56.79%
WD0	56.90%	59.30%	57.40%	59.10%	58.90%	54.90%	54.20%	52.40%	48.70%	45.10%	54.69%
WD1	60.70%	60.50%	59.00%	58.60%	57.90%	59.70%	50.70%	51.60%	49.40%	44.50%	55.26%
WD2	52.00%	47.70%	56.80%	51.40%	50.70%	52.30%	56.80%	41.70%	45.00%	44.60%	49.90%
WT0	57.30%	60.00%	58.50%	58.30%	57.80%	57.30%	53.60%	54.80%	49.30%	46.10%	55.30%
WT1	60.40%	57.60%	58.30%	60.50%	60.20%	57.10%	55.70%	51.80%	50.50%	48.00%	56.01%
WT2	57.30%	58.40%	53.70%	57.20%	58.30%	57.30%	59.50%	50.20%	56.50%	50.80%	55.92%
WDT0	58.30%	57.00%	59.30%	59.80%	61.20%	59.30%	57.20%	55.30%	50.70%	47.40%	56.55%
WDT1	52.50%	51.50%	55.80%	59.40%	58.00%	50.60%	54.50%	55.50%	50.80%	49.80%	53.84%

Table 11: Contextual Filters with the RDWT for EMG Processing

Delay	0 ms	10 ms	20 ms	30 ms	40 ms	50 ms	60 ms	70 ms	80 ms	90 ms	AVG
X1	61.90%	63.80%	64.70%	63.10%	66.50%	63.20%	53.10%	65.00%	61.70%	61.30%	62.43%
X2	58.10%	60.40%	59.00%	59.40%	60.70%	60.40%	58.00%	58.20%	58.70%	53.60%	58.65%
X3	59.10%	63.70%	63.60%	63.50%	51.50%	65.00%	63.80%	64.10%	63.00%	61.30%	61.86%
X4	61.60%	65.10%	65.50%	59.40%	64.50%	65.40%	59.30%	66.10%	63.90%	57.80%	62.86%
X5	35.90%	42.10%	46.00%	53.10%	50.00%	57.30%	55.00%	49.80%	51.60%	39.30%	48.01%

Table 12: Special Wavelet EMG Features

References

[1] Condor Cluster. http://www.cs.wisc.edu/condor/.

[2] Niels Birbaumer. The Thought Translation Device (TTD) for Completely Paralyzed Patients. IEEE, 2000.

[3] Jan Calliess. Further Investigations on Unspoken Speech. Studienarbeit, 2006.

[4] Mac A. Cody. The wavelet packet transform - extending the wavelet transform. Dr. Dobb's Journal, 1994.

[5] Matthias Honal and Tanja Schultz. Identifying User State using Electroencephalographic Data. In *Proceedings of the International Conference on Multimodal Input (ICMI), Trento, Italy*, October 2005.

[6] Herbert H. Jasper. The Ten-Twenty Electrode System of the International Federation. *Electroencephalography and Clinical Neurophysiology. EEG Journal*, (10):371375, 1958.

[7] Chuck Jorgensen and Kim Binsted. Web Browser Control Using EMG Based Sub Vocal Speech Recognition. In *Proceedings of the 38th Hawaii International Conference on System Sciences*, 2005.

[8] Szu-Chen Jou, Tanja Schultz, Matthias Walliczek, Florian Kraft, and Alex Waibel. Towards Continuous Speech Recognition using Surface Electromyography. In *Proc. Interspeech*, Pittsburgh, PA, Sep 2006.

[9] Nick G. Kingsbury. A Dual-Tree Complex Wavelet Transform with Improved Orthogonality and Symmetry Properties. In *Proc. IEEE Conf. on Image Processing, Vancouver*, 2000.

[10] Alfred Karl Louis, Peter Maaß, and Andreas Rieder. *Wavelets*. Teubner, 2. edition, 1998.

[11] Lena Maier-Hein, Florian Metze, Tanja Schultz, and Alex Waibel. Session Independent Non-Audible Speech Recognition Using Surface Electromyography. In *Proc. ASRU*, 2005.

[12] T. Malina, A. Folkers, and Ulrich G. Hofmann. Real-time EEG processing based on Wavelet Transformation. In *12^{th} Nordic Baltic Conference on Biomedical Engineering and Medical Physics, Reykjavik*, June 2002.

[13] Stephane G. Mallat. A Theory for Multiresolution Signal Decomposition: The Wavelet Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):pp. 674–693, July 1989.

[14] C. Mayer. UKA EMG/EEG Studio v2.0.

[15] Todd Rowland. L^2-Space. From MathWorld–A Wolfram Web Resource, created by Eric W. Weisstein. http://mathworld.wolfram.com/L2-Space.html.

[16] Ivan W. Selesnick, Richard G. Baraniuk, and Nick G. Kingsbury. The Dual-Tree Complex Wavelet Transform. *IEEE Signal Processing Magazine*, 22(6):123–151, November 2005.

[17] Mark J. Shensa. The Discrete Wavelet Transform: Wedding the Trous and Mallat Algorithms. *IEEE Transactions on Signal Processing*, 40:2464 – 2482, October 1992.

[18] Patrick Suppes, Zhong-Lin Lu, and Bing Han. Brain Wave Recognition of Words. *Proc. Natl. Acad. Sci. USA*, 94:14965–14969, December 1997.

[19] Marek Wester. Unspoken Speech - Speech Recognition Based On Electroencephalography. Master's thesis, Lehrstuhl Prof. Waibel Interactive Systems Laboratories Carnegie Mellon University, Pittsburgh, PA, USA and Institut für Theoretische Informatik Universität Karlsruhe (TH), Karlsruhe, Germany, 2006.

Hiermit versichere ich, dass ich die vorliegende Arbeit selbständig erstellt habe und keine anderen als die angegebenen Hilfsmittel verwendet habe.

Ort, Datum Unterschrift