



**Bachelor-Thesis      2009**

# **Vokalanalyse im Zeitbereich**

Fachbereich	Informatik
Studierende	Fabrizio Sguaitamatti
Professoren	Dr. Erich Badertscher
Experte	Dr. Federico Flückiger



## Inhaltsverzeichnis

Einleitung.....	2
Aufgabenstellung .....	2
Realisation .....	2
Finden der Merkmale.....	4
Kurvenextrema .....	4
Number of Positives / Negatives .....	5
Aufbereitung des Signals .....	9
Faltung des Ursprungssignals .....	9
Wahl des Faltungskernels .....	9
Der Binomialkernel.....	11
Matching .....	14
Ergebnisse .....	17
Zusammenfassung .....	20
Schlusswort .....	21
Anhang A - Glossar .....	22
Octave.....	22
Anhang B – Verwendete Funktionen.....	23
VBPRConvoluteWhole.m .....	23
VBPRBinary.m .....	23
VBPRBinomFilter.m .....	23
VBPRMonoMixdown.m.....	24
VBPRPosNegGraph.m.....	24
VBPRMaxAverageGraph.m.....	24
VBPRDiffPlot.m.....	24
VBPRFindLocalMaxima.m.....	25
Anhang C – Sourcecode .....	26
VBPRConvoluteWhole.m .....	26
VBPRBinomFilter.m .....	29
VBPRBinary.m .....	30
Verzeichnisse .....	31
Abbildungsverzeichnis .....	31
Literaturverzeichnis .....	31



# Einleitung

## Aufgabenstellung

Thema dieser Thesis ist die Analyse von gesprochenen und aufgezeichneten Vokalen verschiedener Personen im Zeitbereich des Signals. Anhand von unterschiedlichen, aus dem Signal extrahierten Merkmalen soll es möglich sein, eine Person zu identifizieren.

Die Idee zu dieser Arbeit ist aus der Diplomarbeit IBHE1-08 entstanden, wo diese Analyse im Frequenzbereich des Signals vorgenommen wurde.

## Realisation

Für die Realisation dieses Projekts standen bereits einige Rohdaten aus der Diplomarbeit IBHE1-08 zur Verfügung. Diese dienten als Grundlage zur Untersuchung der Merkmale im Zeitbereich der verschiedenen Vokale und Personen. Ausserdem konnten einige Funktionen direkt übernommen werden oder mussten nur minimal angepasst werden.

Untersucht wurden folgende Punkte:

- Welche Merkmale lassen sich im Zeitbereich extrahieren, und welche eignen sich zur Identifikation einer Person?
- Inwiefern beeinflusst die Wahl des Filters das Ergebnis?
- Ist es möglich, eindeutige Statistiken zur Erkennung einer Person zu erstellen?

In der ersten Phase des Projekts wurde nach Möglichkeiten gesucht, Merkmale im Zeitbereich eines Signals zu extrahieren. Zuerst musste untersucht werden, ob es überhaupt solche Merkmale geben könnte. Aus der vorangegangenen Diplomarbeit IBHE1-08 war schliesslich bekannt, dass sich die Perioden eines Signals einer bestimmten Person sehr ähnlich sehen, sich jedoch von Person zu Person manchmal deutlicher, manchmal weniger deutlich unterscheiden.

In der zweiten Phase wurde untersucht, ob sich durch diese extrahierten Merkmale die unterschiedlichen Aufnahmen einer Person zuordnen liessen. Dazu wurden verschiedene Funktionen implementiert, um die statistischen Daten, die zuvor extrahiert und für jede untersuchte Person abgespeichert wurden, auszuwerten und zu veranschaulichen.

In der letzten Phase des Projekts ging es noch darum, die beiden ersten Phasen zu kombinieren und eine Funktion zu schreiben, die darauf abzielt, eine Identifikation der Personen zu automatisieren und ein eindeutiges, binäres Resultat zu liefern.



Realisiert wurde das Projekt in „Octave“<sup>1</sup>.

---

<sup>1</sup> siehe Anhang A

## Finden der Merkmale

### Kurvenextrema

Um einen ersten Eindruck zu erlangen, welche Merkmale im Zeitbereich eine Rolle spielen könnten, wurden zuerst die Durchschnittsperioden, welche in der Diplomarbeit IBHE1-08 verwendet wurden, analysiert.

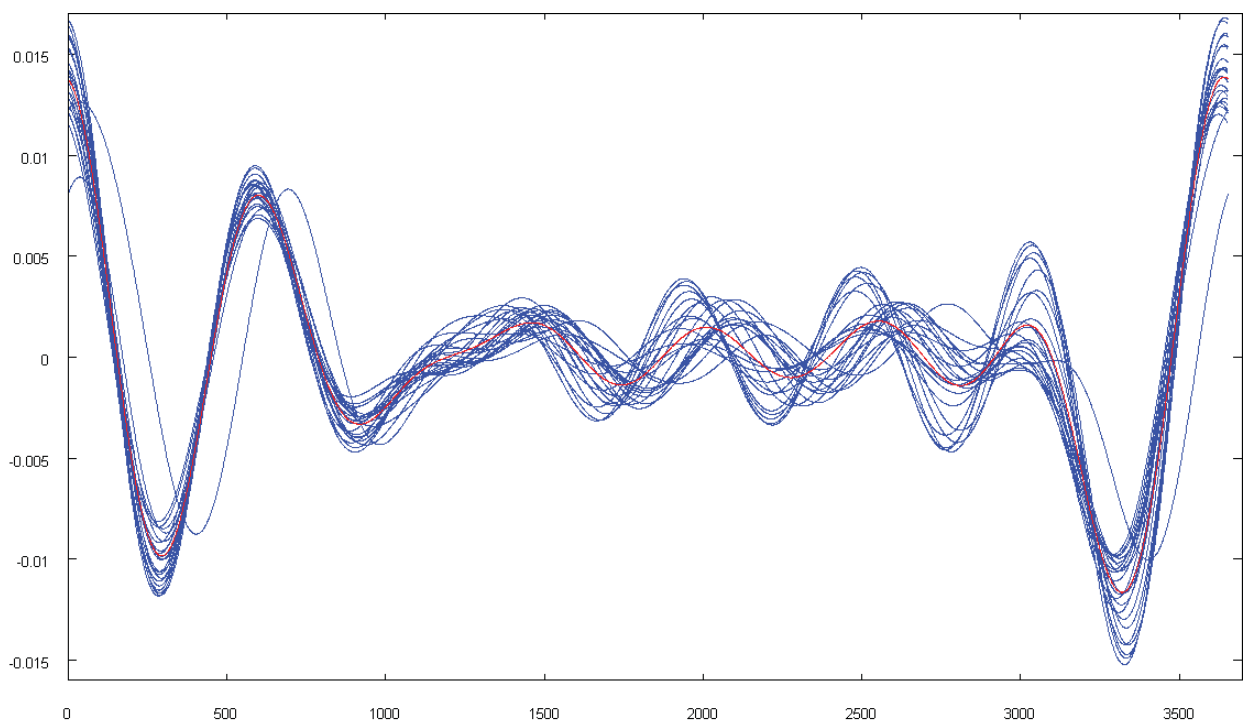


Abbildung 1: Overlay aller Perioden (blau) & Durchschnittsperiode (rot ) der Person „ps“

Wie man auf der Abbildung gut erkennen kann, scheinen die Kurvenextrema vor allem im Bereich von 350 Samples bzw. bei 600 und 3300 Samples sehr konstant zu sein. Bei der Untersuchung weiterer Plots verschiedener Personen wurden diese Ausprägungen intrapersonell - also bei Vergleichen von Aufnahmen derselben Person - als sehr konstant empfunden; im Bereich der interpersonellen Vergleiche – also den Vergleichen zwischen verschiedenen Personen – aber doch als genügend unterschiedlich:

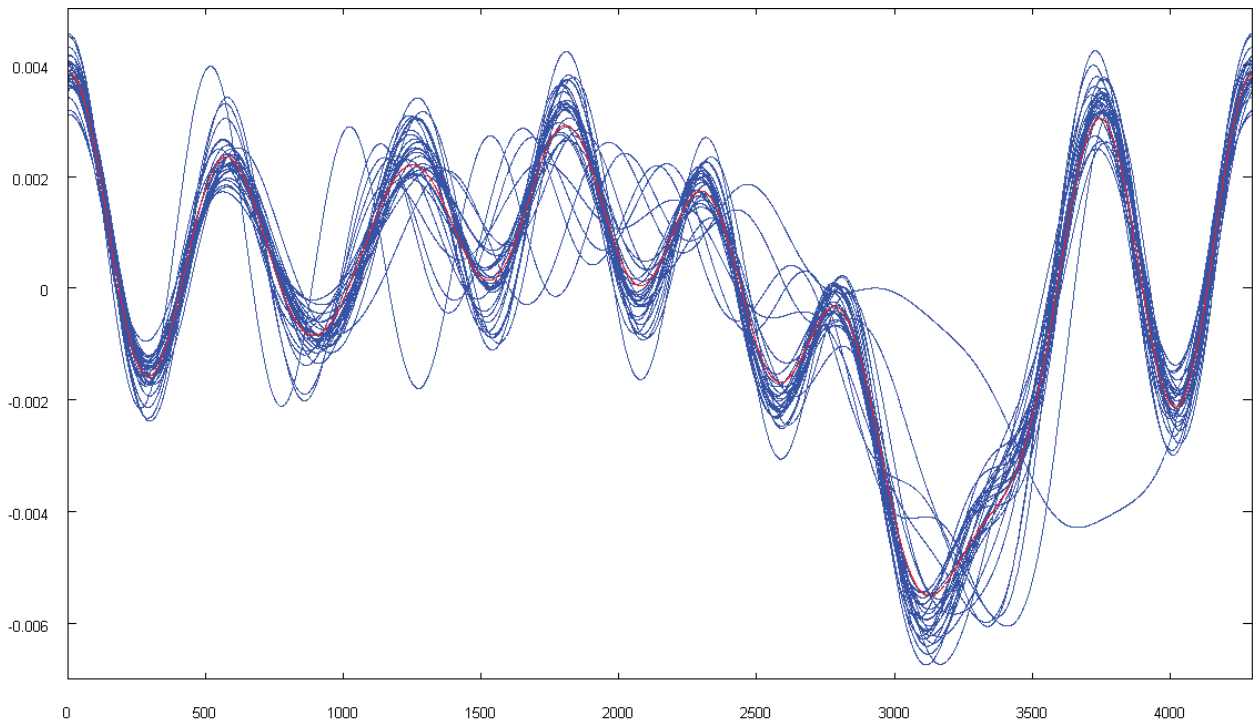


Abbildung 2: Overlay aller Perioden (blau) & Durchschnittsperiode (rot) der Person „br“

Durch die Konstanz der Extrema lassen sich gleich 2 Kenngrößen vermuten:

1. die Position der Extrema in Relation zu der Signal bzw. Periodenlänge (also des Zeitbereichs des Signals in einem bestimmten Abschnitt)
2. die Anzahl der Extrema, die innerhalb einer Periode auftreten

Wie sich aus Abbildung 1 und Abbildung 2 schließen lässt, unterscheiden sich Personen also jeweils durch die Position der Extrema und deren Anzahl voneinander.

Eine weitere Kenngröße lässt sich ausserdem sofort an den Plots ablesen, und zwar handelt es sich dabei um die durchschnittliche Periodenlänge. Wir haben also aus der Arbeit mit der Durchschnittsperiode bereits 3 Kenngrößen ermitteln können, mit welchen weiter gearbeitet wurde.

### Number of Positives / Negatives

Eine weitere Idee entstand aus der Überlegung, das Verhalten des Signals um eine Nulllinie zu beobachten. Dazu war es zuerst einmal nötig, aus der Durchschnittsperiode die Veränderungen über die gesamte Zeitspanne des Signals zu beobachten. Diese Beobachtung resultierte im so genannten „Differenzen-Plot“.

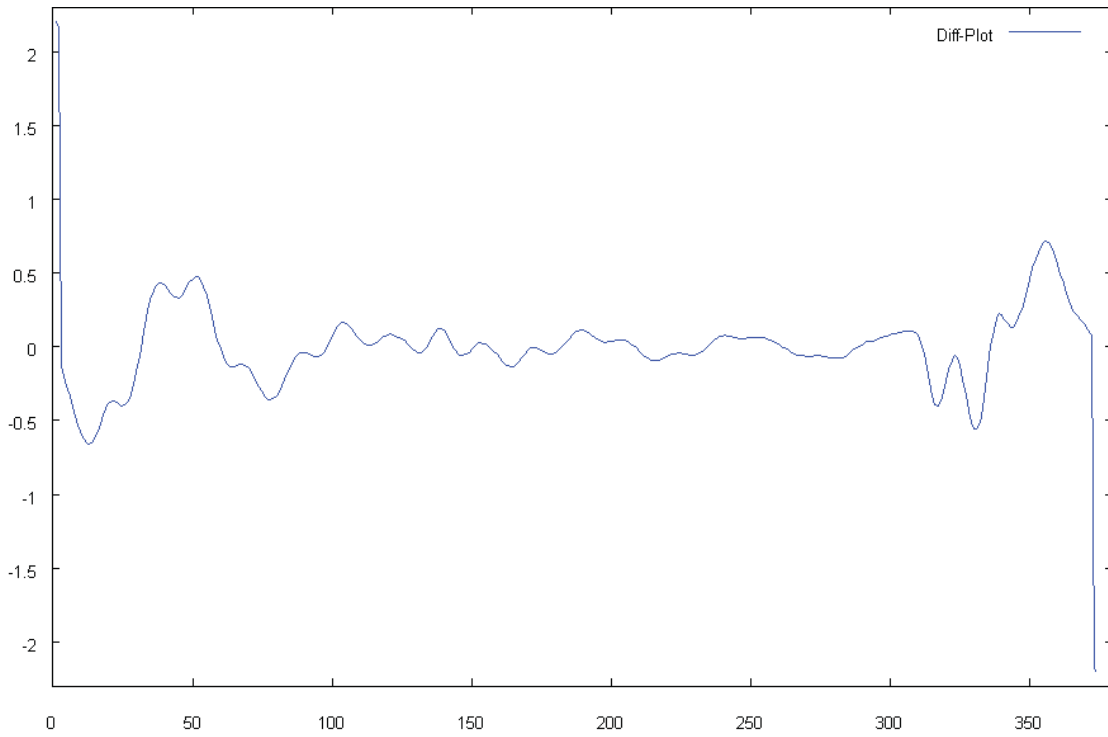


Abbildung 3: Differenzen-Plot der Person "ps", Session 1

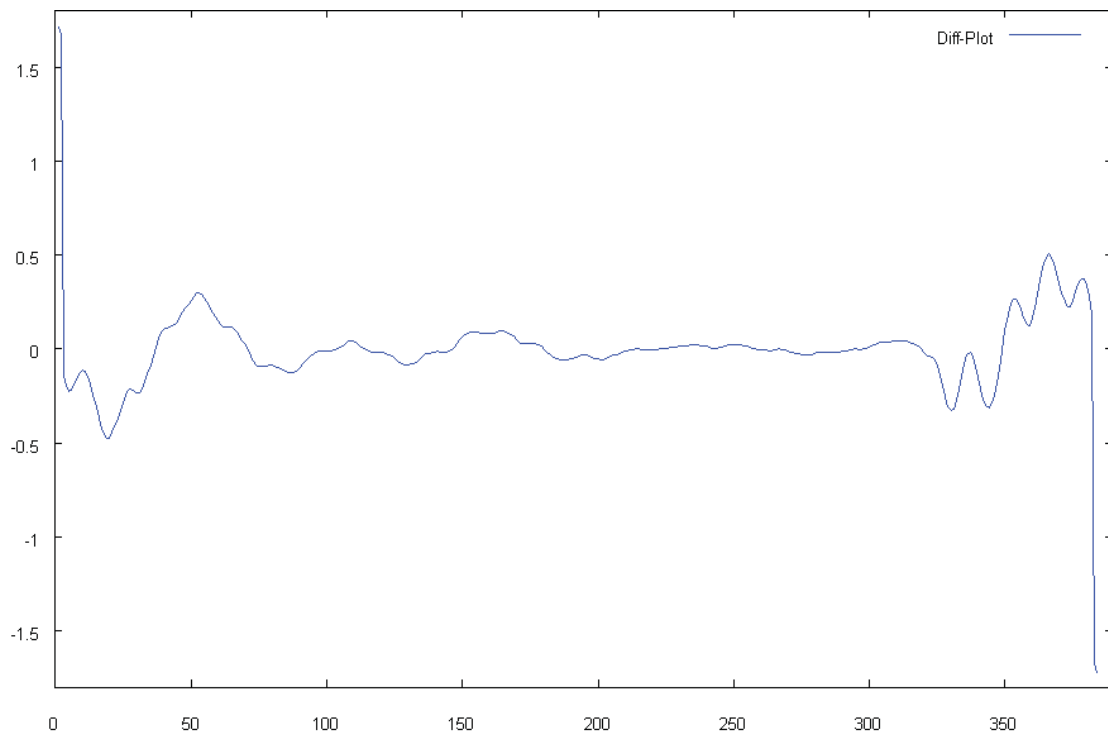


Abbildung 4: Differenzen-Plot der Person "ps", Session 2



Der Vergleich der beiden Plots zeigt auch hier wieder einen schönen Wiedererkennungswert: man hat zu Beginn des Signals negative Ausprägungen, die sich rasch in positive Ausprägungen wandeln.

Ab ca. 100 Samples flacht das Signal zusehends ab und man sieht nur wenige Veränderungen, bis dann in der Gegend von 300 Samples wieder ähnlich den Anfängen des Signals auf eine negative Ausprägung eine positive folgt.

Durch diesen Fluss von positiven und negativen Änderungen wurde schliesslich folgendes binäres Signal erstellt, welches aus der Octave Funktion `VBPRBinaryPlot` kreiert wird:

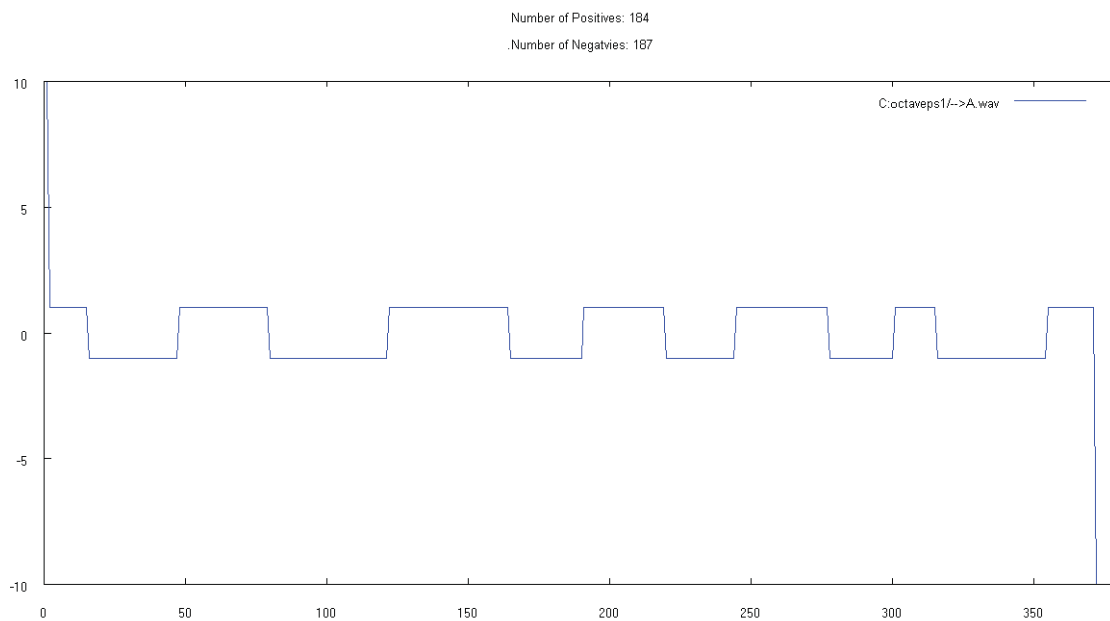


Abbildung 5: Binäres Signal der Person "ps", Session 1



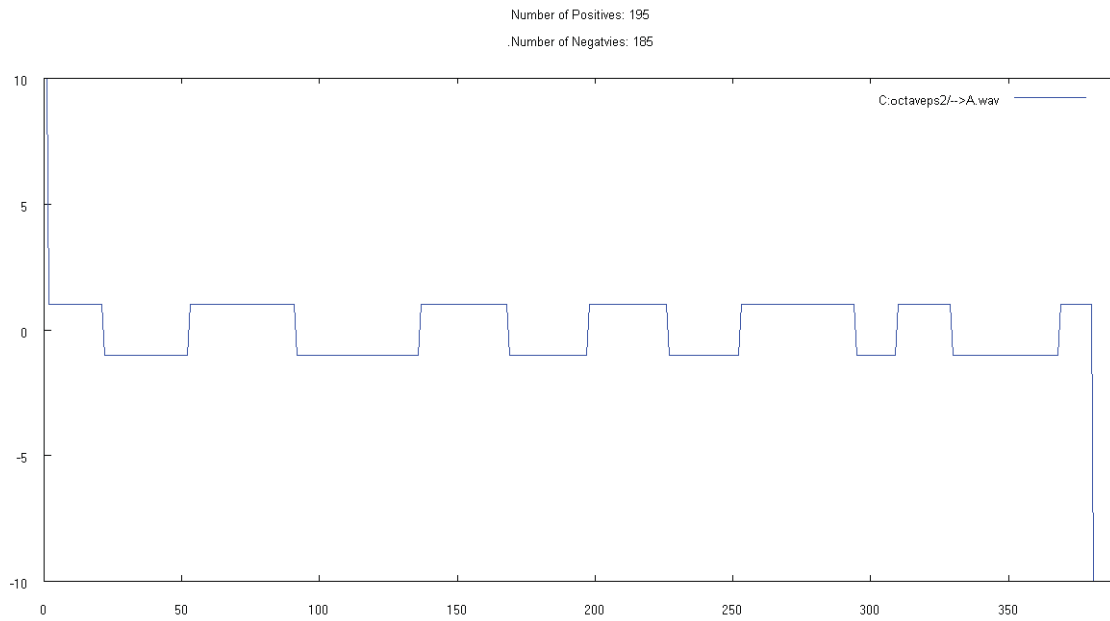


Abbildung 6: Binäres Signal der Person "ps", Session 2

Zusätzlich dazu wurden für die Personenstatistiken die Anzahl der positiven und die Anzahl der negativen Werte gespeichert.

Problematisch wurde die Analyse dieses binären Signals vor allem dadurch, dass die kleineren Veränderungen zwischen 100 und 300 Samples genau gleiches Gewicht hatten wie die doch eher markanten und besser sichtbaren Ausprägungen. Somit konnte durch diese Methode kein eindeutig zuordenbares Ergebnis gefunden werden (siehe Kapitel *Ergebnisse*).



## Aufbereitung des Signals

### Faltung des Ursprungssignals

Damit eine saubere Analyse des Signals erfolgen konnte, musste das Signal erstmal von kleineren Störungen befreit werden. Dies geschah mit Hilfe des Octave-Befehls `conv`, welcher eine Faltung des Signals verursacht. Die Funktion benötigt ein Input-Signal - in unserem Fall also das gesamte Signal einer Vokal-Aufnahme – und einen Kernel, mit welchem das Eingangssignal gefaltet wird. Eine Faltung ist im Grunde nichts anderes als eine Filterung des Signals, wobei im Frequenzbereich von Filterung und im Zeitbereich von einer Faltung gesprochen wird.

Eine Faltung ist ziemlich rechenintensiv; es werden  $N \cdot M$  Multiplikationen benötigt, wobei  $N$  und  $M$  die jeweiligen Längen der zu faltenden Signale bzw. Kernel ist. Eine solche diskrete Faltung ist definiert durch:

$$(f * g)(n) = \sum_{k \in D} f(k)g(n - k)$$

Eine Effiziente Implementierung im Bezug auf die Rechenleistung bietet die „Schnelle Faltung“, welche sich wiederum auf den FFT Algorithmus zur schnellen Berechnung der Fourier -Transformation stützt.

Der Vorteil der Faltung liegt darin, dass bereits akzentuierte Gebiete des Signals noch stärker hervorgehoben werden und weniger akzentuierte Gebiete entsprechen gedämpft werden. Dadurch lassen sich die für unsere Zwecke wichtigen Maxima bzw. Minima noch besser hervorheben und aus dem Signal extrahieren.

### Wahl des Faltungskernels

Bis zur Endgültigen Wahl des Faltungskernels wurden verschiedene Kernel ausprobiert. Anfangs wurden ganz simple Kernel eingesetzt, z.B.

$$[1/8 \ 3/8 \ 3/8 \ 1/8]$$

Diese Kernel glätteten das Ursprungssignal zwar etwas, aber es war in den meisten Fällen nicht genug. Es ergaben sich noch zu viele kleine Ungereimtheiten, welche



die Suche nach den Minima bzw. Maxima in einer Periode unnötig erschweren und zum Teil dann auch nicht eindeutige Resultate liefern.

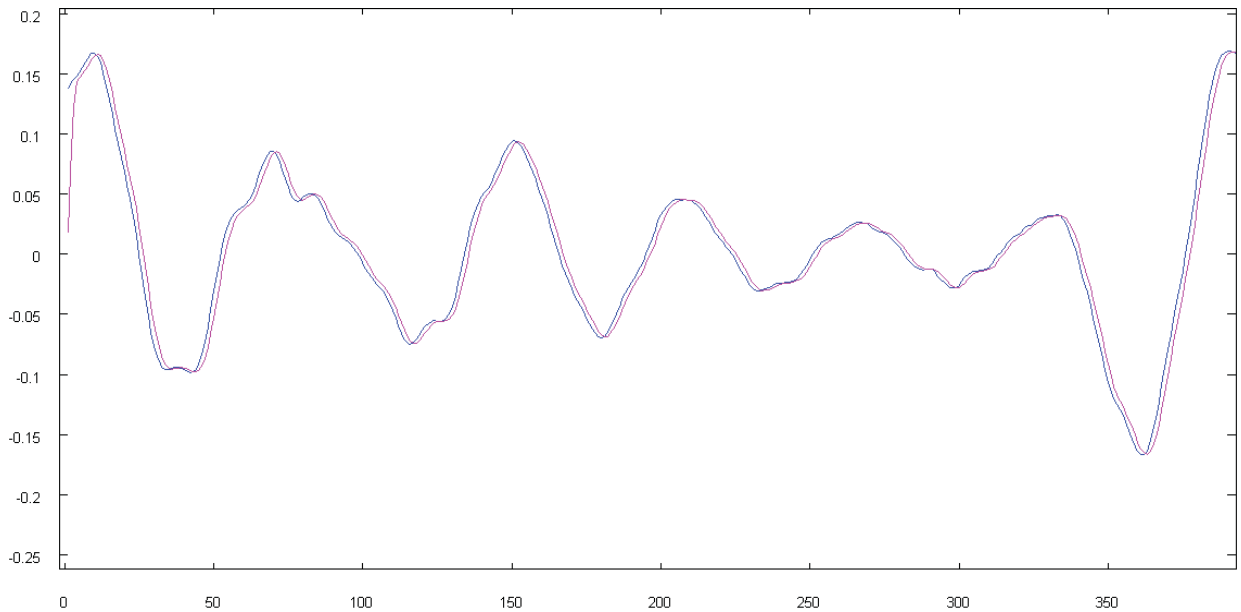


Abbildung 7: Faltung mit  $[1/8 \ 3/8 \ 3/8 \ 1/8]$  Kernel, Original- (blau) & Resultatsignal (Magenta)

Wie man in der Abbildung 3 gut erkennen kann weicht das gefaltete Signal (Magenta) nicht erheblich vom ursprünglichen Signal (blau) ab. Viele der kleinen „Fehler“ im Signal sind immer noch da; es gibt noch zu viele Störungen.

Im Vergleich dazu zeigt sich in Abbildung 4 eine erhebliche Verbesserung der Erkennbarkeit der Maxima bzw. Minima in einer Periode:

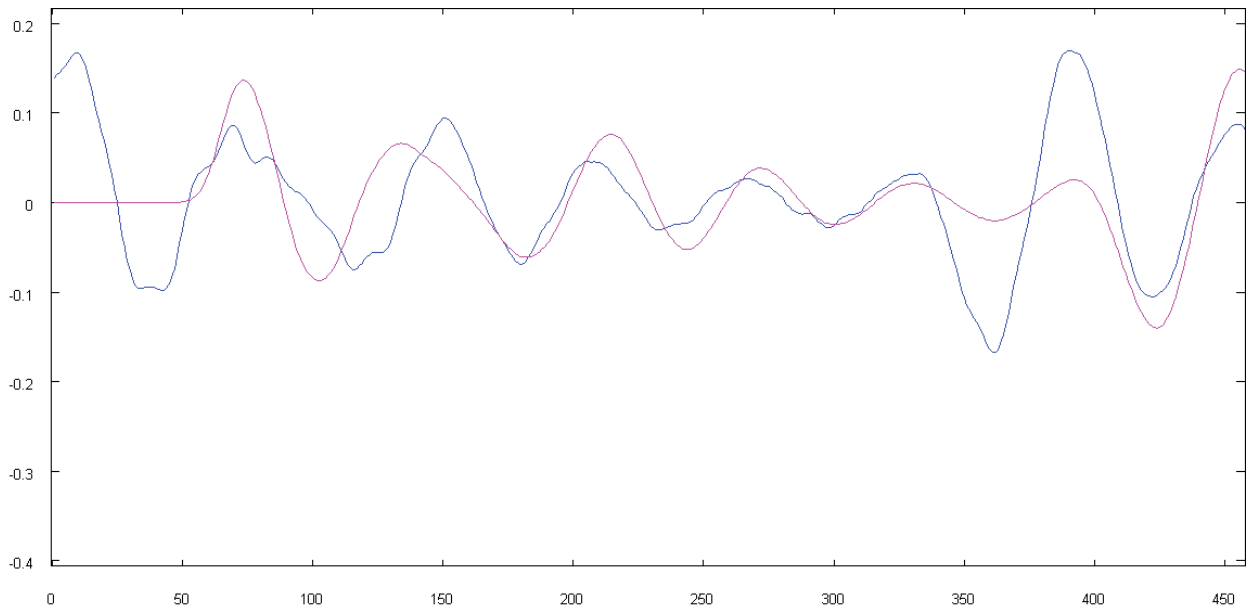


Abbildung 8: Faltung mit Binomial-Kernel, Original- (blau) & Resultatsignal (Magenta)

Die Wahl des Binomial-Filters bzw. Kernels zur Faltung des gesamten Input-Signals war also entscheidend am Prozess zur Verbesserung der Interpretierbarkeit des Signals beteiligt.

### Der Binomialkernel

Wie bereits weiter oben erwähnt trug der Binomialkernel entscheidend dazu bei, dass sich das verrauschte Signal besser interpretieren liess.

Der Binomialkernel wurde in der Octave-Funktion `VBPRBinomFilter` definiert. Er setzt sich wie folgt zusammen:

- zuerst wird ein Ursprungskernel definiert,  $[1/2 \ 1/2]$
- dieser Kernel wird mit sich selbst gefaltet. Es entsteht ein neuer Kernel,  $[1/4 \ 2/4 \ 1/4]$
- der neue Kernel wird wiederum mit sich selbst gefaltet.
- etc.

Wie man durch Multiplikation mit der jeweiligen Iterationsstufe herausfinden kann, entsteht ein Kernel der einer Reihe im Pascal'schen Dreieck entspricht:

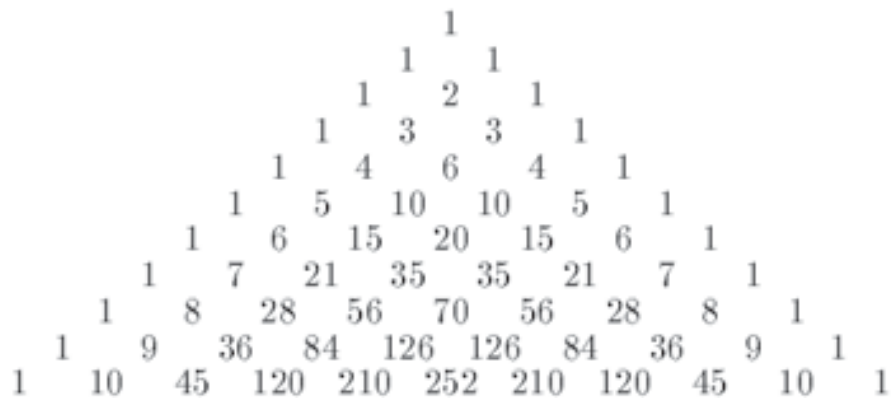


Abbildung 9: Ausschnitt aus einem Pascal-Dreieck

Bei der Iterationsstufe 4 würde man also mit 16 ( $2^4$ ) Multiplizieren, was aus dem Kernel

$$[0.062500 \quad 0.250000 \quad 0.375000 \quad 0.250000 \quad 0.062500]$$

die im Pascal Dreieck auf der vierten Stufe zu findenden Reihe

$$[1 \quad 4 \quad 6 \quad 4 \quad 1]$$

entstehen lässt. Der Binomialkernel lässt sich so beliebig vergrössern und verfeinern. Aber es ist Vorsicht geboten: mit einem zu grossen Kernel kann es passieren, dass das Signal komplett ausgelöscht wird!

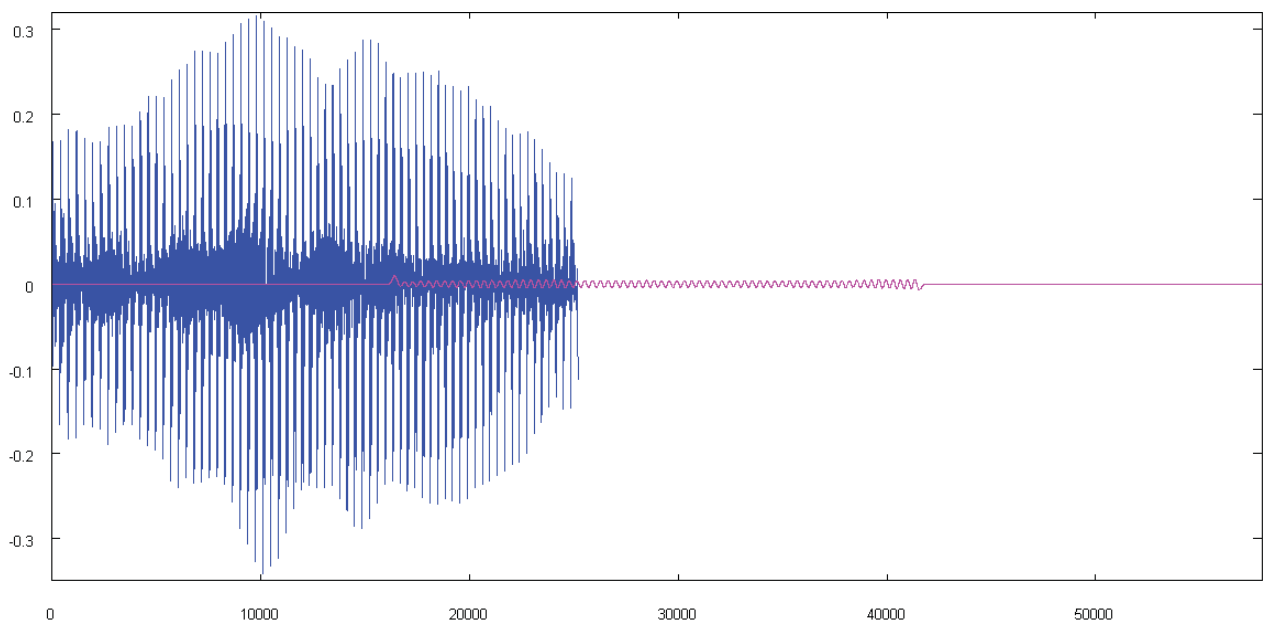


Abbildung 10: Auslöschung des Resultatsignals durch zu grossen Kernel



Dieses Problem tritt allerdings nicht nur bei Binomialkernen auf, sondern generell bei zu grossem Kernel.

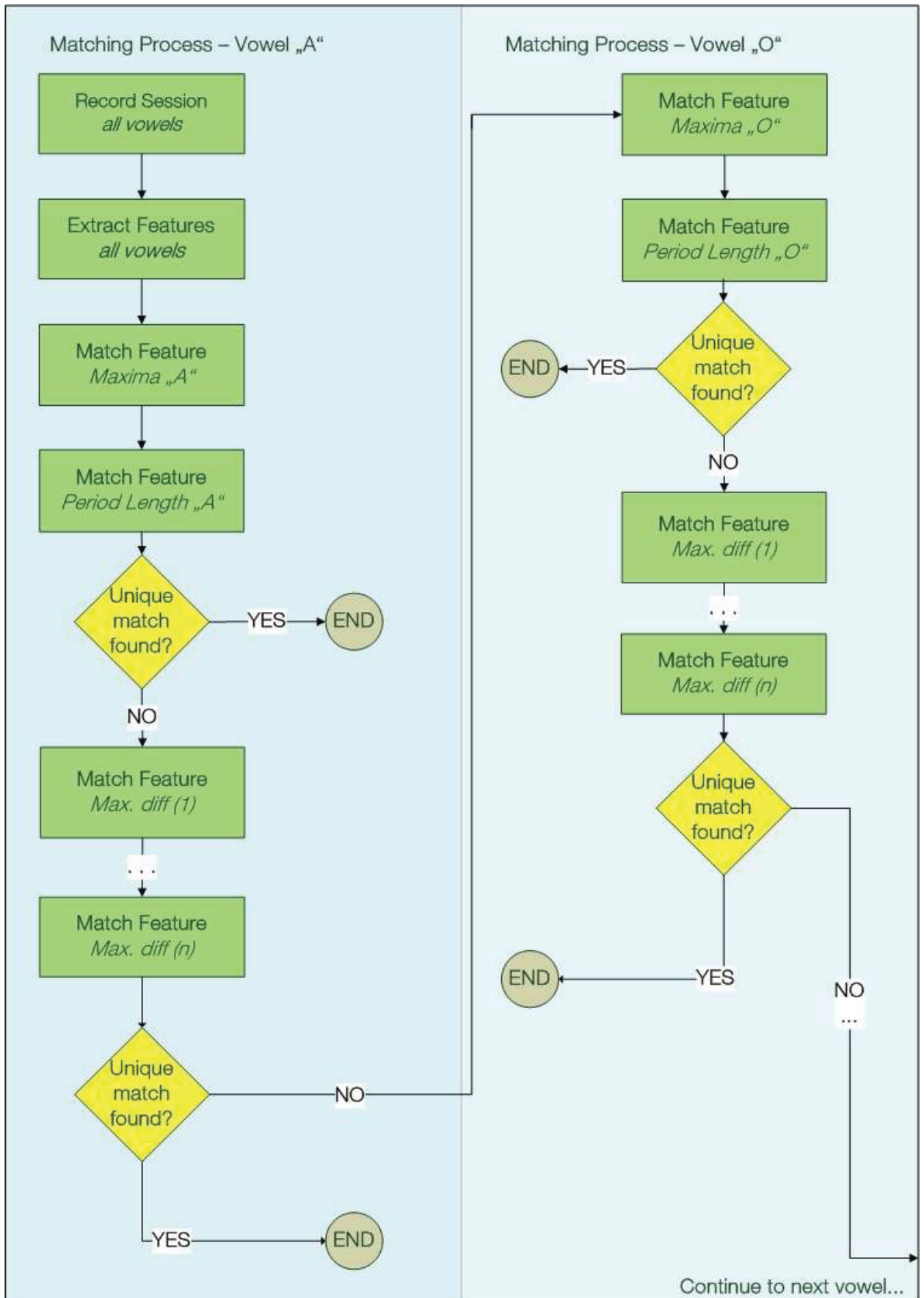
Für die Untersuchung des Vokals „A“, mit welchem ich mich in diesem Projekt hauptsächlich beschäftigt habe, schien eine Iterationsstufe von 8 als guter Wert standzuhalten.



## Matching

Das Matching – also der Prozess, bei welchem – aus dem Vergleich zwischen der zu analysierenden Aufnahme und der gesammelten statistischen Daten aller bisherigen Aufnahmen eine klare Aussage getroffen werden soll, um welche Person es sich handelt – wurde abschliessend aus zeitlichen Gründen nicht mehr fertig implementiert.

Dieser Prozess sieht folgendermassen aus:







Zuerst wird von der zu identifizierenden Person eine Session mit den Vokalen A, E, I, O, und U aufgezeichnet. Aus diesen WAV Dateien werden mit den obigen Methoden alle zur Verfügung stehenden Merkmale extrahiert und gespeichert.

In einem ersten Identifizierungsschritt wird die Anzahl der Maxima verglichen. Dabei fallen alle Personen weg, die keine identische Anzahl an Maxima haben. Es bleiben also nur noch diejenigen zum Vergleich, welche dieselbe Anzahl an Maxima in einer Periode aufweisen.

Aus diesem Personenkreis werden nun die Periodenlängen verglichen. Gibt es da eine Übereinstimmung, und ist diese Übereinstimmung eindeutig oder einzigartig – sprich sie kommt im ganzen Recordset nur einmal vor – so sollte die zu identifizierende Person gefunden worden sein.

Falls noch keine Übereinstimmung gefunden wurde, oder aber mehrere Übereinstimmungen gefunden wurden (non-unique), wird die Suche weiter verfeinert. Dazu wird die Information der Abstände zwischen den einzelnen Maxima verglichen. Da bekanntlich die bereits gefilterte Menge an in Frage kommenden Personen dieselbe Anzahl an Maxima besitzen, können diese nun 1:1 verglichen werden. Sollte sich wider Erwarten noch keine eindeutige Übereinstimmung finden lassen, wiederholt sich die gesamte Prozedur mit dem nächsten Vokal.

## Ergebnisse

Der Erfolg der beiden weiter oben beschriebenen Methoden zur Ermittlung statistischer Daten im Zeitbereich der Aufnahmen war unterschiedlich.

Das Verfahren der Ermittlung der positiven und negativen Werte im Differenzen-Plot der Durchschnittsperiode erzeugte folgendes Ergebnis:

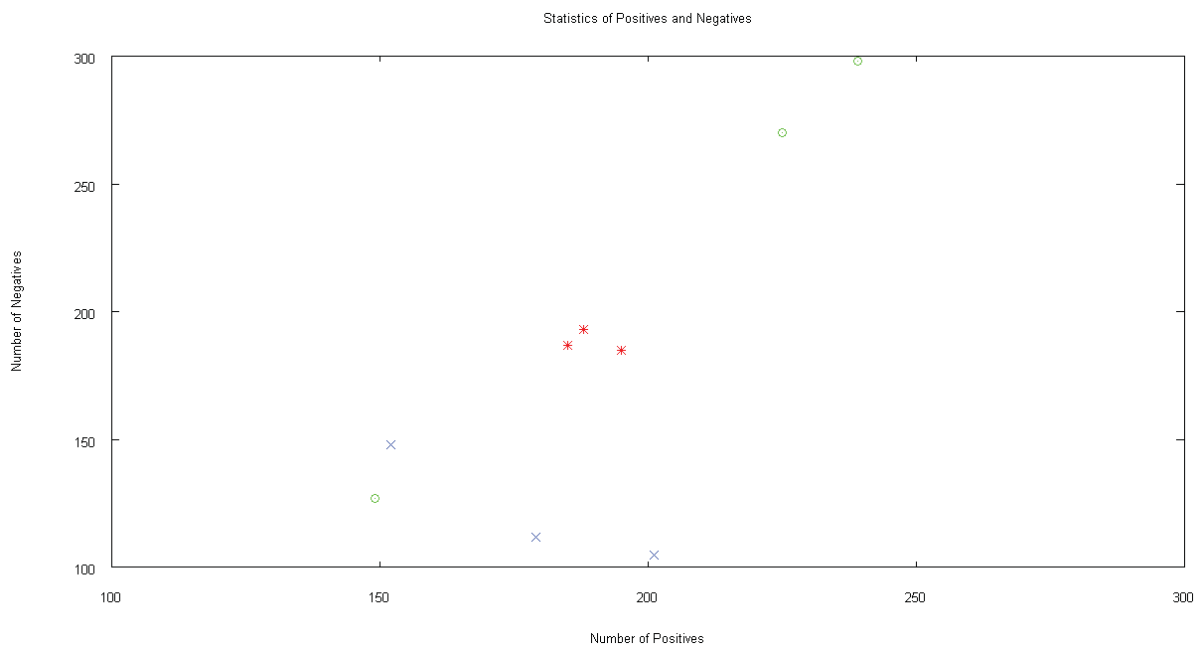


Abbildung 11: Statistik der interpersonellen Verteilung der Anzahl von positiven/negativen Punkte in der Durchschnittsperiode

Man erkennt sehr gut, dass bei der Person „rot“ ein gutes Clustering der Werte stattgefunden hat. Leider verlief dieses Clustering bei den Personen „grün“ und „blau“ nicht ganz so erfolgreich. Vor allem Person „grün“ zeigt eine sehr breite Streuung, sowohl von den positiven wie auch den negativen Werten.

Dies lässt sich wie weiter oben bereits erwähnt dadurch begründen, dass die Mittelabschnitte der Differenzen-Vektoren zwar im Durchschnitt weniger starke Ausprägungen haben, diese aber genau gleich gewichtet sind. Somit haben kleine Veränderungen dieselbe Auswirkung wie grosse Veränderungen. Gepaart mit der unterschiedlichen Aufnahmequalität und den dadurch auch unterschiedlichen Mengen and Rauschen ist es praktisch nicht möglich, alleine über diese Parameter eine Unterscheidung der Personen zu bewerkstelligen.

Das bessere Ergebnis wurde durch den Vergleich der Maxima einer Person mit der Länge der Durchschnittsperiode erzielt:

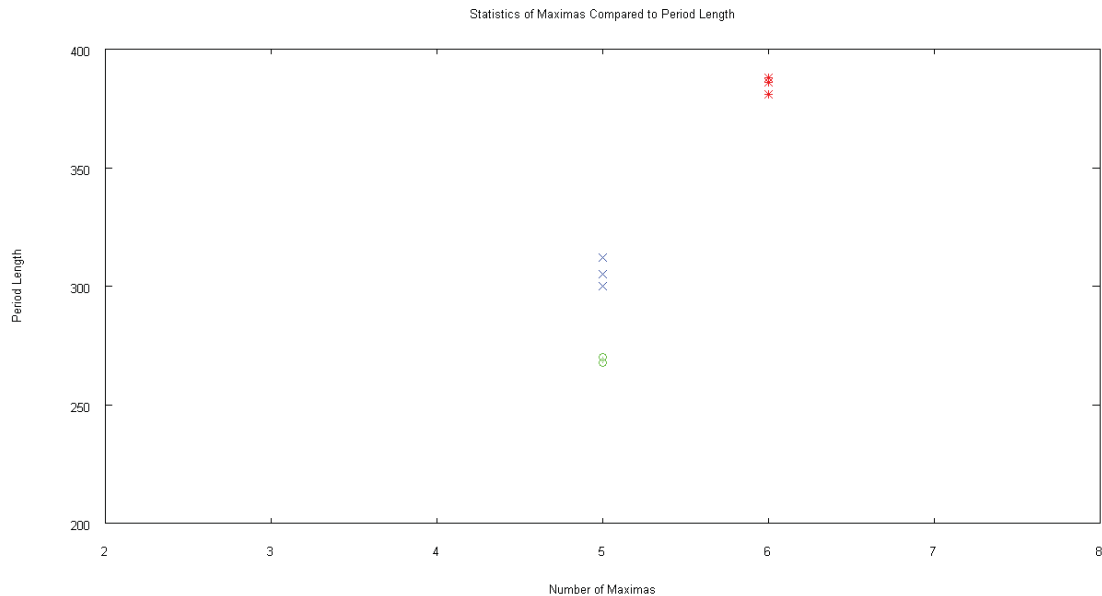


Abbildung 12: Statistik der interpersonellen Verteilung der durchschnittlichen Anzahl Maximas und der Periodenlänge

Wie man auf der Abbildung sehr schön erkennen kann haben wir von den im obigen Beispiel bereits bekannten Personen „grün“, „rot“ und „blau“ ein viel besseres Clustering der Werte. Sowohl im Bereich der Maximas als auch im Bereich der Periodenlänge bleiben die Aufnahmen konstant. Zusätzliches Rauschen hat hier keinen Einfluss und somit ist diese Lösung auch robust gegenüber etwas schlechteren Aufnahmen, sollten diese vorkommen.

Natürlich besteht die Gefahr, dass mehrere Personen sowohl dieselbe Anzahl Maximas wie auch eine annähernd identische Periodenlänge aufweisen. Für diesen Fall stehen noch weitere statistische Daten zur Verfügung:

1. der Abstand zwischen den einzelnen Maximas innerhalb einer Periode
2. der Vergleich der statistischen Werte der Personen mit einem weiteren Vokal, z.B. „O“

Vergleicht man z.B. den Abstand zwischen den auftretenden Maximas einer Person (Max.diff) so fällt auf, dass diese intrapersonell genügend konstant sind und interpersonell zum Teil recht unterschiedlich sind:

<b>Max. pos</b>	74	134	214	272	331	392
<b>Max. diff</b>	60	80	58	59	61	64
<b>Ratio</b>	1.33	0.73	1.02	1.03	1.05	0.98

Abbildung 13: Tabelle der 1. Periode von Person „ps“ mit Abstand zwischen den Maxima und dem zugehörigen Verhältnis

<b>Max. pos</b>	2000	2062	2145	2200	2252	2312
<b>Max. diff</b>	62	83	55	52	60	68
<b>Ratio</b>	1.34	0.66	0.95	1.15	1.13	0.93

Abbildung 14: Tabelle der 6. Periode von Person „ps“ mit Abstand zwischen den Maxima und dem zugehörigen Verhältnis

<b>Max. pos</b>	2084	2143	2190	2230	2282	
<b>Max. diff</b>	59	47	40	52	90	
<b>Ratio</b>	0.80	0.85	1.30	1.73	0.67	

Abbildung 15: Tabelle der 9. Periode von Person „vg“ mit Abstand zwischen den Maxima und dem zugehörigen Verhältnis

<b>Max. pos</b>	4080	4141	4188	4230	4282	
<b>Max. diff</b>	61	47	42	52	84	
<b>Ratio</b>	0.77	0.89	1.24	1.62	0.73	

Abbildung 16: Tabelle der 17. Periode von Person „vg“ mit Abstand zwischen den Maxima und dem zugehörigen Verhältnis

Diese Kennzahlen könnten in einem weiteren Schritt als gute Unterscheidungsmerkmale dienen, wenn man z.B. über die eher grobe und Filtrierung durch die Anzahl der Maxima und der Periodenlänge auf Situationen treffen sollte, die noch kein eindeutiges Ergebnis liefern.



## Zusammenfassung

Wie aus der Lektüre der obigen Erkenntnisse zu erkennen ist, war die Qualität des Matchings der Personen über Vokale im Zeitbereich ungefähr auf demselben Niveau wie dies im Frequenzbereich der Fall war. Fairerweise muss man dazu aber auch sagen, dass sich im Frequenzbereich meiner Meinung nach noch mehr detaillierte Informationen herausholen lassen, als das im Zeitbereich der Fall war.

Mit dem Fortschreiten der Arbeit hat sich mein Eindruck gefestigt, dass eine ideale Lösung für eine Spracherkennung von Personen auf Basis von gesprochenen Vokalen in einer Kombination der beiden Methoden fassen sollte, und dass daraus sicher eine weitere interessante Thesis entstehen könnte.

Auch immer deutlicher wurde, wie wichtig die Filterung bzw. die Faltung der Signale ist: ein verrauschtes Signal benötigt eine viel intensivere Bearbeitung als ein sozusagen „perfektes“ Signal, weshalb sich teilweise auch die Resultate entsprechend darstellten. Ein weiterer Punkt der im weitesten Sinne damit zusammenhängt ist die Auswahl des Aufnahmeorts und der Ausrüstung. Je besser die Hardware und je „klinischer“ die Umgebung, in welcher die Aufnahmen entstehen, desto besser sind die daraus resultierenden WAV-Files. Beispielsweise waren die Aufnahmen, die ich an meinem Arbeitsplatz aufgenommen habe zum Teil unbrauchbar, da die Hintergrundgeräusche (surrende Computer, sprechende Mitarbeiter etc.) sich zwar beim Anhören der Aufnahme nicht bemerkbar gemacht hatten, jedoch bei der Analyse des Signals eindeutigen Einfluss darauf hatten.



## Schlusswort

Bereits in der Schlussphase der Diplomarbeit IBHE1-08 war vermutet worden, dass sich ein Matching von Personen auch im Zeitbereich realisieren lassen könnte, und dass eine Arbeit darüber eine Interessante und vor allem lehrreiche Angelegenheit werden würde.

Als ich mich dazu entschloss, diese Arbeit alleine zu machen, war ich mir noch nicht bewusst, wie wertvoll die Unterstützung anderer Kollegen in diesem Umfeld sein kann. Sie wurde manchmal sehr vermisst. Glücklicherweise hatte ich mit Herrn PD Prof. Dr. Erich Badertscher einen Betreuer an der Seite, der mich bei meinem Vorhaben jederzeit mit vollem Einsatz unterstütz hat. Dafür möchte ich mich an dieser Stelle nochmals recht herzlich bedanken.

Ausserdem möchte ich auch meiner Familie und vor allem meiner Frau danken, die mich immer bedingungslos unterstützt haben und so manches Wochenende auf mich verzichten mussten.



## Anhang A - Glossar

### Octave

GNU Octave ist eine Hochsprache, die hauptsächlich für numerische Berechnungen verwendet wird. Das Freeware-Tool „Octave“ lässt sich mit Matlab vergleichen. Es bringt viele eingebaute Funktionen mit, welche sich bestens für die Signalanalyse bewährt haben. Das Schreiben eigener Funktionen geschieht mittels „.m“-Files, die den Source der Funktion beinhalten. Funktionen können auch untereinander verknüpft und aufgerufen werden.

Ideal war in diesem Zusammenhang auch das Tool GNU-Plot, welches mit Octave mitgeliefert wird. Es lassen sich so sehr einfache aber ansehnliche Plots erstellen, exportieren, drucken usw.

Octave läuft sowohl unter Linux als auch unter Windows, wobei im Rahmen der Bachelorthesis ausschliesslich unter Windows gearbeitet wurde. Erwähnenswert ist zudem noch, dass die Linux Version, welche in der vorangegangenen Diplomarbeit verwendet wurde, wesentlich stabiler lief.



## Anhang B – Verwendete Funktionen

Es folgt eine Auflistung der wichtigsten benutzten Funktionen. Einige wurden im Rahmen der Diplomarbeit IBHE1-08 bereits erstellt, weshalb sie hier nicht explizit aufgelistet sind. Folgende Funktionen sind erweiterte oder neu hinzugekommene Funktionen:

### **VBPRConvoluteWhole.m**

Input: Pfad zu einem beliebigen Stereo WAV-Signal eines Vokals  
Output: Plot des Ursprungssignals und des gefalteten Signals

Diese Funktion verwendet als Input das ganze Vokalsignal. Es bearbeitet dabei den Input mit einem Binomial-Kernel und faltet es zu einem neuen Signal, welches anschliessend im Vergleich zum Ursprungssignal geplottet wird.

Ausserdem werden verschiedene Merkmale des so entstandenen Signals dem jeweiligen Sprecher bzw. Vokal zugeordnet (Maxima, durchschnittliche Periodenlänge etc.)

### **VBPRBinary.m**

Input: Vektor, der die Durchschnittsperiode eines Vokals beschreibt  
Output: Plot eines binären Signals und die Anzahl der positiven / negativen Werte

Diese Funktion nimmt als Input ein Vektor mit den Werten, aus welchen schliesslich die Durchschnittsperiode entsteht. Für jeden Wert des Vektors prüft die Funktion, ob er positiv oder negativ ist. Diese Werte werden anschliessend für die Session in Variablen geschrieben und auf dem Plot ausgedruckt. Sie werden ausserdem für die Weiterverwendung gespeichert.

### **VBPRBinomFilter.m**

Input: -  
Output: Ein binomialer Kernel

Diese einfache Funktion erstellt durch einen definierten Startkernel  $\rightarrow [1/2 \ 1/2]$  durch Iteration einen grösseren binomialen Kernel. Je höher die Iterationsstufe, desto grösser der Kernel.





### **VBPRMonoMixdown.m**

Input: Stereo WAV-Signal  
Output: Der Vektor des Monosignals

Diese Funktion überprüft, ob das Eingangssignal ein Stereo-Signal ist. Falls dies der Fall ist, wird aus den beiden Input-Vektoren (für jeden Stereokanal einer) ein Durchschnittsvektor erzeugt, welcher gerade dem Monosignal aus den beiden Stereovektoren entspricht.

### **VBPRPosNegGraph.m**

Input: Anzahl positiver und negativer Werte  
Output: 2-dimensionaler Graph mit interpersoneller Statistik

VBPRPosNegGraph vergleicht die Anzahl der positiven und negativen Werte einer Person und illustriert diese anhand eines 2-dimensionalen Graphen.

### **VBPRMaxAverageGraph.m**

Input: Anzahl Maxima und Länge der Durchschnittsperiode  
Output: 2-dimensionaler Graph mit interpersoneller Statistik

VBPRMaxAverageGraph vergleicht die Anzahl der Maxima einer Person gepaart mit der Länge der Durchschnittsperiode und illustriert diese anhand eines 2-dimensionalen Graphen.

### **VBPRDiffPlot.m**

Input: Pfad zu einem beliebigen Stereo WAV-Signal eines Vokals  
Output: Differenzen-Plot des Input-Signals

Aus dieser Funktion entsteht der Differenzen-Plot. Dieser beschreibt die durchschnittliche Änderung des Signals über die gesamte Zeitspanne in der Durchschnittsperiode.



## VBPRFindLocalMaxima.m

Input: Ein beliebiger Signalvektor

Output: Ein Vektor mit den Positionen der gefundenen lokalen Maxima

Diese Funktion überprüft in einem frei definierbaren Fenster für jeden Wert des Input-Vektors die umliegenden Werte (links und rechts). Ist ein Wert innerhalb dieses gegebenen Fensters der höchste Wert, wird er als lokales Maxima betrachtet und in den Maxima-Vektor eingetragen. Diese Prozedur wird solange wiederholt bis jeder Wert des Vektors überprüft wurde.

## Anhang C – Sourcecode

In diesem Abschnitt sei der Sourcecode der wichtigsten Funktionen näher beschrieben. Da es keinen Sinn macht, den Sourcecode von allen Funktionen in diesem Dokument zu platzieren, wurden 3 Funktionen ausgewählt:

### VBPRConvoluteWhole.m

```

%
% Convolutes the whole vowel with a binomial kernel, extracts
the period length
% of each period and the number of maximas in any given period
% VERSION 1
%
% Returns: number of maximas, period lengths, plots it
% Input: path to file and wavfile of one vowel (44100 Hz)
%

function av = VBPRConvoluteWhole (path, filename)

% clear console and close old plot windows
close all;
clc;
page_screen_output(false);           % supresses paging on screen
output
tic;                                   % store starttime

% if there is no / at the end of the path, we append it
if(!strcmp(path(length(path)), "/"))
    path = strcat(path, "/"); endif;
file = strcat(path, filename);

% check if we have a valid file
[ststruct, err, msg ] = stat(file);
if (err != 0)
    printf("-Error: no such file: %s\n",file);

    return;
endif;

textout = "";
plots = true;

% read wavfile and convert it to mono
[wave,fs,bit] = wavread(file);
if (fs != 44100)
    printf("-Error: not 44,1kHz!\n");
    return;

```

```

    endif;
    signal = VBPRMonoMixdown(wave);

    % give some additional information of wavefile
    len = length(signal);
    len_s = len/44100;
    textout = strcat(textout,sprintf("Length of Mono Signal [s]:
%f\n", len_s));
    textout = strcat(textout,sprintf("Length of Mono Signal
[samples]: %i\n",
        len));

    % cut away noise (threshold is 15 * max of a window)
    noise = signal(1:1000); % first 1000
samples as noise reference
    minpeak = abs(min(noise));
    maxpeak = abs(max(noise));
    if (minpeak > maxpeak) threshold = minpeak*15;
    else threshold = maxpeak*15; endif;
    if (threshold*15 > max(signal)) % to prevent errors :-
        threshold = max(signal)*1/3;
    endif;
    threshold_m = zeros(rows(signal), 1); % used to plot
noise threshold
    threshold_m = threshold_m + threshold; % as well here

    % search signal start and end and set ROI
    break1 = break2 = false;
    sigstart = 1;
    sigend = len;
    i = 1;
    while(!(break1 & break2) && i < len)
        %search from the beginning and mark position of first
threshold overrun
        if (abs(signal(i)) > threshold_m & !break1)
            sigstart = i;
            break1 = true;
        endif;
        %search from the back and mark position of first
threshold overrun
        if (abs(signal(len-i)) > threshold_m & !break2)
            sigend = len -i;
            break2 = true;
        endif;
        i = i + 100; % reduce step to 1 makes it more accurate,
but much slower
    endwhile;
    textout = strcat(textout,sprintf("Signal start [index]: %i end:
%i\n",
        sigstart,sigend));

```

```

    vowel = signal(sigstart:sigend);           % cut away noise

    % take middle of signal f. average calculation => ROI
    roi = round(length(vowel)/2);
    vowel(round(length(vowel)*1/4):round(length(vowel)*3/4));
    roi_len = length(roi);
    textout = strcat(textout,sprintf("Length of ROI [samples]:
%i\n",
                                   roi_len));

    % call the binomial kernel function
    VBPRBinomFilter;

    % now convolute
    convsig = conv(binomfilter,vowel);
    dasFile = strcat(path, filename, ".txt");

    % find all the maximas and save them to a file
    maximaindex = VBPRFindLocalMaxima (convsig,5)
    save ("-ascii", dasFile, "maximaindex")

%VBPRExtractPeriod(vowel)

    % plot win1 (NOISE,ROI,thresholds)
    if(plots)
        % prepare marker vector to visualize indices and
thresholds
        marker = zeros(rows(signal), 1);
        marker(sigstart, 1) = max(signal);
        marker(sigend, 1) = max(signal);

    plottitle = strcat("Monosignal in time domain: ", path,
filename);

    figure;
    grid;
    plot(vowel);
    xlabel('Samples');
    title(plottitle);
    hold on;
    %plot(1:rows(threshold_m), threshold_m, "r");
    %plot(1:rows(threshold_m), (threshold_m*-1), "r");
    %plot(1:rows(marker), marker, "g");
    plot(convsig, "-14");
    %hold off;
    axis([0 4400 -0.5 0.5])

    %print(strcat(path,"data_tech1_",filename, ".gif"), "-dgif");
    endif;

```



```
endfunction;
```

## VBPRBinomFilter.m

```
%  
% Creates a simple binomial kernel for use with convolution  
function  
% VERSION 1  
%  
% Returns: a binomial kernel  
% Input: none  
%  
  
% set initial kernel size  
binomfilter = [1/2 1/2];  
  
% set amount of iterations  
limit = 8;  
  
% now iterate and create the binomial kernel  
for i = 1:limit-1;  
binomfilter = conv(binomfilter,binomfilter);  
i++;  
endfor  
  
binomfilter;
```

## VBPRBinary.m

```

%
% Analyzes the average vowel signal and counts the positive and
negative
% values in the vector
% VERSION 1
%
% Returns: the binary plot of the average vector and the number
of positive
% and negative values in said vector
% Input: path to file and wavefile of one vowel (44100 Hz)
%

function binarysignal = VBPRBinary (average)
numberofpositives=0;
numberofnegatives=0;

% iterate on the average vector and deem a position positive or
negative
for p=2:(length(average)-1)
    if (average (p) > 0)
        binarysignal(p) = 1;
        ++numberofpositives;
    else
        binarysignal(p) = -1;
        ++numberofnegatives;
    endif
endfor

% for better readability
binarysignal(1) = 10;
binarysignal((length(average)-1)) = -10;

% convert to strings
pos = int2str(numberofpositives)
neg = int2str(numberofnegatives)
figure(1);

% plot it
plot (binarysignal)
plottitle = strcat("Number of Positives: ",pos,"\n.,"Number of
Negatvies: ",neg);
title (plottitle);
endfunction

```



## Verzeichnisse

### Abbildungsverzeichnis

[1]	Abbildung 1: Overlay aller Perioden (blau) & Durchschnittsperiode (rot) der Person „ps“ .....	4
[2]	Abbildung 2: Overlay aller Perioden (blau) & Durchschnittsperiode (rot) der Person „br“ .....	5
[3]	Abbildung 3: Differenzen-Plot der Person "ps", Session 1 .....	6
[4]	Abbildung 4: Differenzen-Plot der Person "ps", Session 2 .....	6
[5]	Abbildung 5: Binäres Signal der Person "ps", Session 1 .....	7
[6]	Abbildung 6: Binäres Signal der Person "ps", Session 2 .....	8
[7]	Abbildung 7: Faltung mit $[1/8 \ 3/8 \ 3/8 \ 1/8]$ Kernel, Original- (blau) & Resultatsignal (Magenta).....	10
[8]	Abbildung 8: Faltung mit Binomial-Kernel, Original- (blau) & Resultatsignal (Magenta).....	11
[9]	Abbildung 9: Ausschnitt aus einem Pascal-Dreieck .....	12
[10]	Abbildung 10: Auslöschung des Resultatsignals durch zu grossen Kernel.....	12
[11]	Abbildung 11: Statistik der interpersonellen Verteilung der Anzahl von positiven/negativen Punkte in der Durchschnittsperiode .....	17
[12]	Abbildung 12: Statistik der interpersonellen Verteilung der durchschnittlichen Anzahl Maxima und der Periodenlänge.....	18
[13]	Abbildung 13: Tabelle der 1. Periode von Person „ps“ mit Abstand zwischen den Maxima und dem zugehörigen Verhältnis .....	19
[14]	Abbildung 14: Tabelle der 6. Periode von Person „ps“ mit Abstand zwischen den Maxima und dem zugehörigen Verhältnis .....	19
[15]	Abbildung 15: Tabelle der 9. Periode von Person „vg“ mit Abstand zwischen den Maxima und dem zugehörigen Verhältnis .....	19
[16]	Abbildung 16: Tabelle der 17. Periode von Person „vg“ mit Abstand zwischen den Maxima und dem zugehörigen Verhältnis .....	19

### Literaturverzeichnis

[1]	Steven W. Smith: Digital Signal Processing, Newnes 2003, ISBN 0-7506-7444-X
[2]	<a href="http://en.wikipedia.org/convolution">http://en.wikipedia.org/convolution</a>
[3]	<a href="http://en.wikipedia.org/Binomial_coefficient">http://en.wikipedia.org/Binomial_coefficient</a>
[4]	<a href="http://wavemetrics.com">http://wavemetrics.com</a>
[5]	<a href="http://www.bores.com">http://www.bores.com</a>