

**Brandenburgische Technische Universität
Cottbus - Senftenberg**

**Lehrstuhl für Kommunikationstechnik
Fakultät 3**

Experimente zur akustischen Mustererkennung

Experiments in Acoustic Pattern Recognition

Bachelorarbeit

Author: Johannes Wawra
Matrikel-Nummer: 3100270
Betreuer: Prof. Dr.-Ing. habil. Matthias Wolff
Dipl.-Ing. (FH) Christian Richter

Inhaltsverzeichnis

Abbildungsverzeichnis	IV
Tabellenverzeichnis	V
Symbolverzeichnis	VI
1 Einleitung	1
2 Versuchsaufbau	2
2.1 Das Mikrofonfeld	2
2.2 Vorversuch und Versuch 1	3
2.3 Versuch 2	4
2.4 Versuch 3	4
3 Genutzte Programme	5
3.1 Signalaufnahme: Das vi Mikrofonfeld 3.0	5
3.1.1 Aufnahme- und Wiedergabemodus aktivieren	6
3.1.2 Ordnerpfade und Dateinamen definieren	6
3.1.3 Kanäle aktivieren und Aussteuerung einstellen	7
3.1.4 Wiedergabeparameter	7
3.1.5 Kontinuierlicher und getriggert Modus	8
3.1.6 Dateiformat	8
3.2 Das Hilfsprogramm Fileoperations	9
3.2.1 Filelist erstellen	9
3.2.2 Logdateien auslesen	10
3.3 UASR und dafür angefertigte Skripte	12
3.3.1 Das Skript MAKEFLST.xtp	12

Inhaltsverzeichnis

3.3.2	Konfigurationsdatei	12
3.3.3	FEA Analyse	13
3.3.4	HMM-Training	13
A	Anhang	15
A.1	Quelltexte	15

Abbildungsverzeichnis

2.1	Skizze Versuchsaufbau Version 1: Glasschale	3
2.2	Skizze Versuchsaufbau Version 2, 3: Plexiglasschale	4
3.1	GUI vi Mikrofonfeld	5

Tabellenverzeichnis

3.1	eine kompliziertere Tabelle	14
-----	---------------------------------------	----

Symbolverzeichnis

Allgemeine Symbole

Symbol	Bedeutung
a	der Skalar a
\vec{x}	der Vektor \vec{x}
\mathbf{A}	die Matrix \mathbf{A}

1 Einleitung

Im Rahmen dieser Bachelorarbeit wurden für den Lehrstuhl für Kommunikationstechnik drei Versionen eines Vorführexperiments zur akustischen Mustererkennung erstellt. Das Experiment soll die Möglichkeiten des Experimentiersystems UASR (Unified Approach to Speech Synthesis and Recognition)¹ demonstrieren.

In dem Experiment wird der Aufschlag von Münzen auf verschiedene Oberflächen mit einem Mikrofonfeld aufgenommen. Dabei soll die Nullhypothese widerlegt werden, dass sich die dabei entstehenden Signale der jeweiligen Münzklasse sich nicht signifikant im Frequenzgang unterscheiden.

¹ Das UASR wurde von Professor Wolff (Lehrstuhl für Kommunikationstechnik der btu cottbus-senftenberg) in Kooperation mit Professor Frank Duckhorn (Lehrstuhl Systemtheorie und Sprachtechnologie der TU Dresden) entwickelt.

2 Versuchsaufbau

Die drei Versionen des Experimentes unterscheiden sich vor allem in ihrem Versuchsaufbau. Allen Versionen gemein ist, dass die Münzen in ein Behältnis geworfen werden, welches 30 cm entfernt von einem Mikrofonfeld auf dem Tisch steht, oder dessen Mittelpunkt 30 cm von dem Mikrofonfeld entfernt ist.

2.1 Das Mikrofonfeld

Das Mikrofonfeld ist in allen drei Versionen des Experimentes identisch und besteht aus 4 identischen Kondensatormikrofonen **TODO: MARKE**, die in einer Lochplatte eingefasst sind. Die Löcher der Platte sind rund mit 1 cm Durchmesser ihre Mittelpunkte haben jeweils 5 cm horizontalen sowohl als auch vertikalen Abstand. Die Mikrofone stecken in den Löchern **TODO: Lochnummern**.

TODO: Daraus ergibt sich eine Richtcharakteristik...

Die Messwerte werden von einer **TODO: Soundkarte** der Firma National Instruments, die über Kabel mit den Mikrofonen verbunden ist, aufgenommen.

2.2 Vorversuch und Versuch 1

In der ersten Version des Experimentes werden die Münzen in eine Glasschale geworfen, deren Mittelpunkt 30 cm von der Lochplatte entfernt ist - siehe dazu die Skizze.

Damit der Einwurf der Münzen in die Schale vergleichbare Ergebnisse liefert, wurde für dieses Experiment ein Münzschlitz angefertigt. Im Vorversuch war der Münzschlitz aus Pappe, im Versuch Version 1 aus Plexiglas. Der endgültige Münzschlitz beinhaltet eine Halterung, in welche die Schüssel gestellt wird. Maße hierzu siehe ebenfalls auf der Skizze.

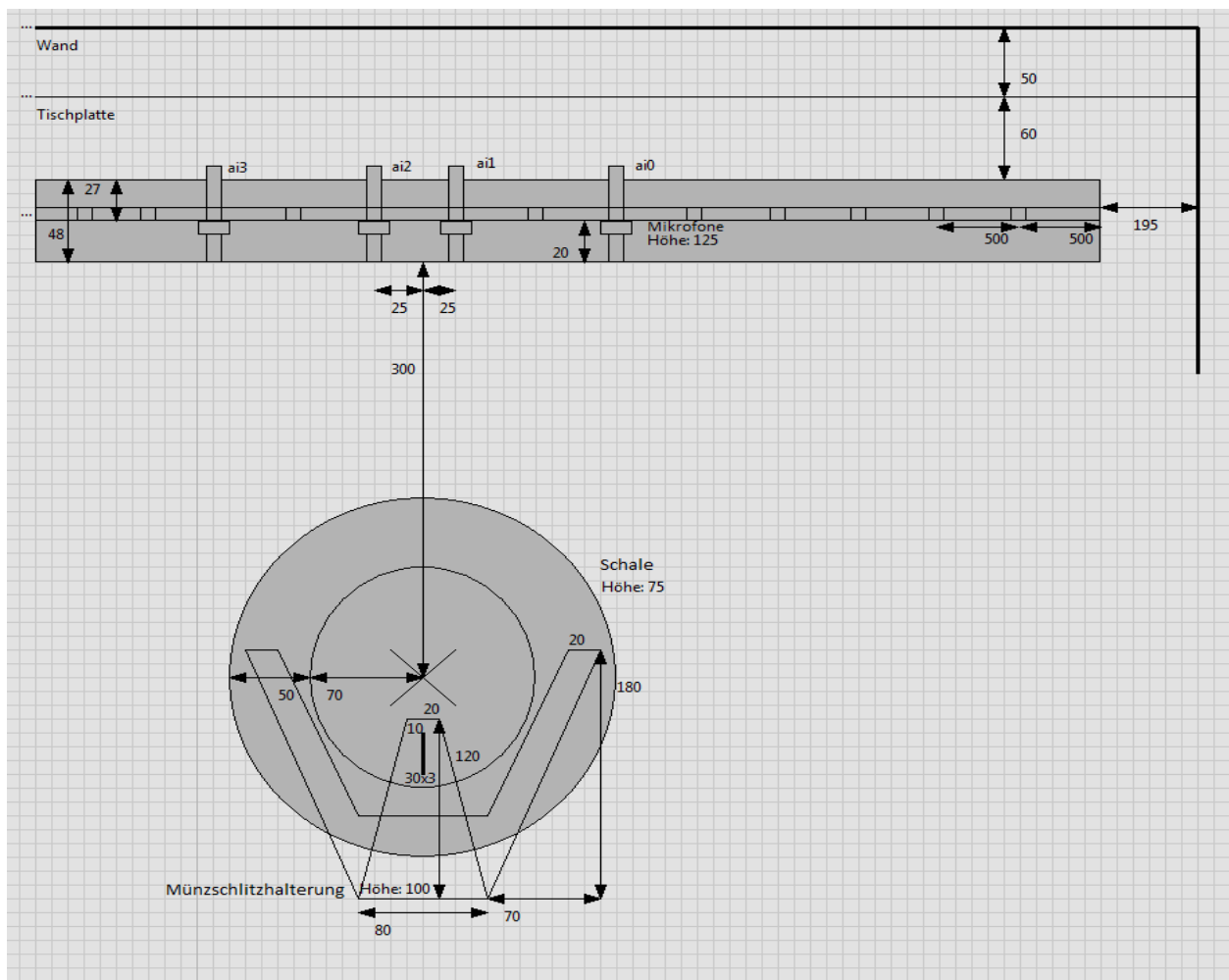


Abbildung 2.1: Skizze Versuchsaufbau Version 1: Glasschale

2.3 Versuch 2

In der zweiten Version wird anstatt der Glasschale eine ebenfalls für dieses Experiment angefertigte rechteckige Schale aus Plexiglas verwendet. In die selbige ist eine Metallplatte eingelegt, an die in jeder Ecke kleine Pads aus gummiartigen Kunststoff geklebt sind, sodass die Platte auf den Pads steht. Maße siehe abermals Skizze.

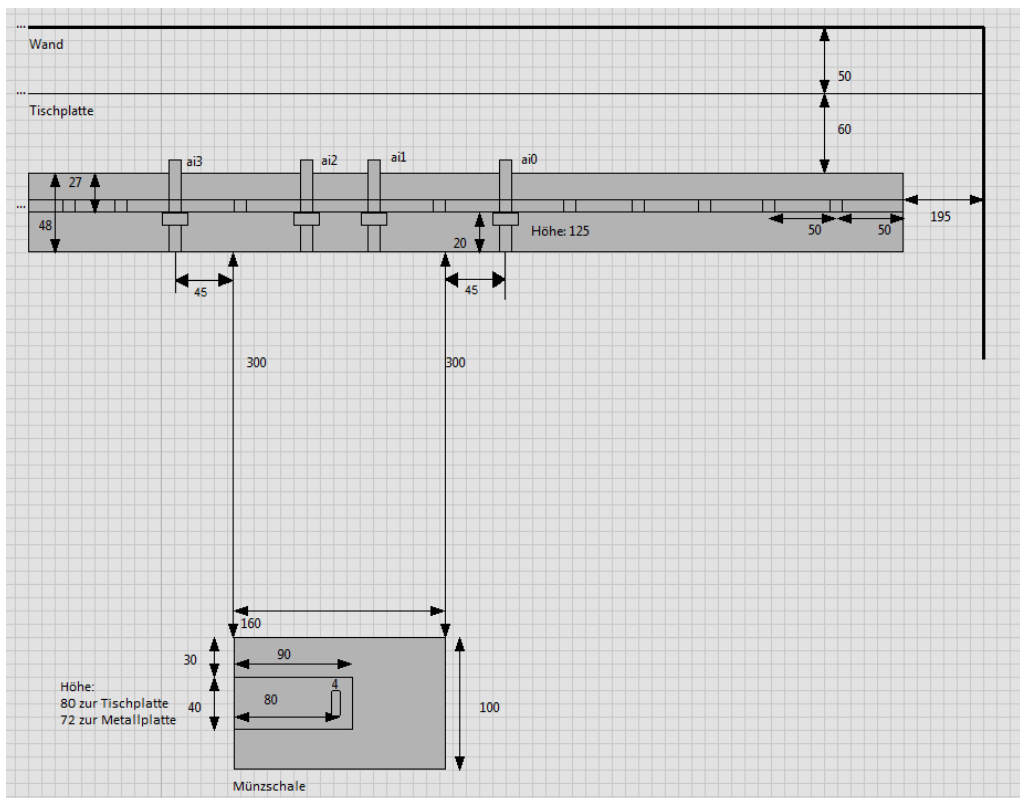


Abbildung 2.2: Skizze Versuchsaufbau Version 2, 3: Plexiglasschale

2.4 Versuch 3

In der dritten Version entspricht der Aufbau dem vorherigen. Nur wird anstatt der Metallplatte eine zugeschnittene unebene Glasplatte ähnlich einem Sichtschutzglas von Türen verwendet. Die obere Oberfläche ist dabei rau, die untere glatt. An der Glasplatte sind keine Pads angebracht, sie liegt direkt mit der glatten Oberfläche auf dem Plexiglas.

3 Genutzte Programme

3.1 Signalaufnahme: Das vi Mikrofonfeld 3.0

Die Signale werden mit einem speziell dafür angefertigten virtuellen instrument (vi) des Programmes "LabVIEW" der Firma National Instruments aufgenommen, welches "Mikrofonfeld 3.0.vi" heißt.

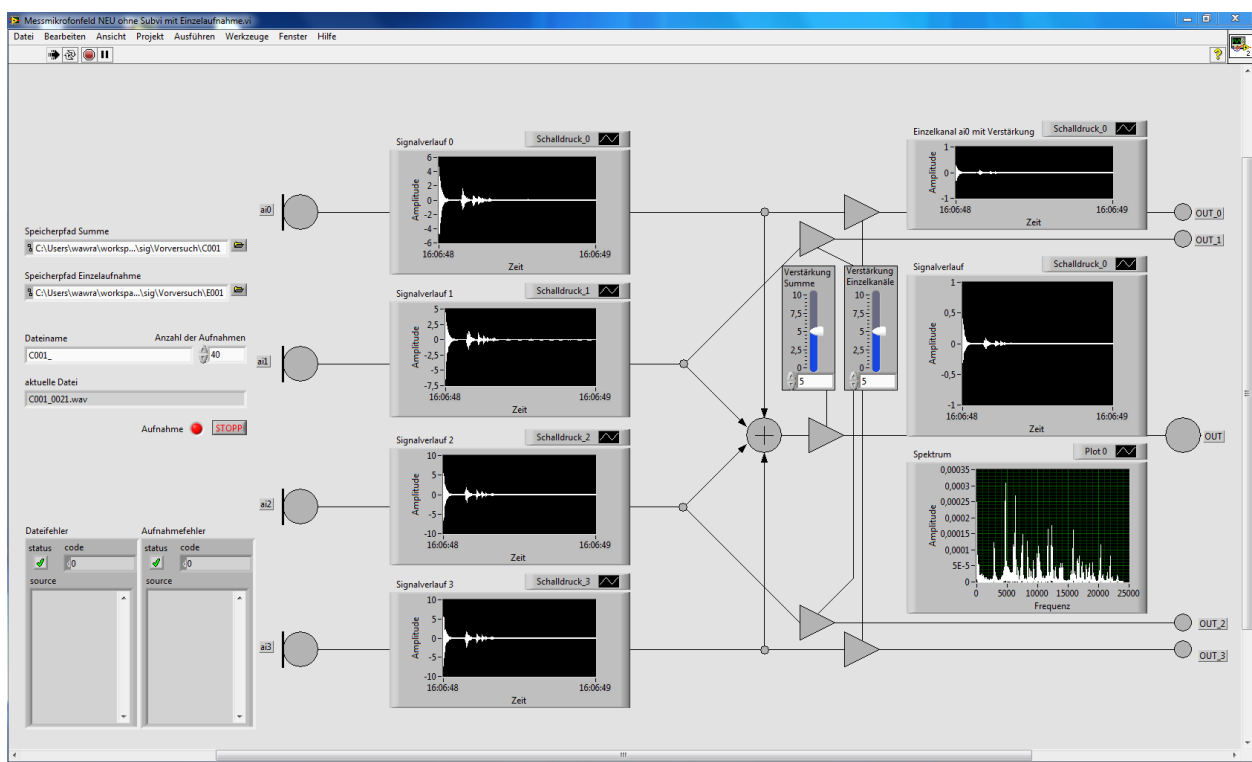


Abbildung 3.1: Benutzeroberfläche des vi Mikrofonfeld

3.1.1 Aufnahme- und Wiedergabemodus aktivieren

Dieses vi erstellt nummerierte Wave-Dateien mit den Aufnahmen des Mikrofonfeldes, kann jedoch auch genutzt werden um das Summen-Signal der Mikrofone am Ausgang der Soundkarte (mit 1 Sekunde Verzögerung) wiederzugeben. Beide Modi können auch kombiniert werden. Für die Auswahl des Aufnahme- oder des Wiedergabe-Modus müssen die jeweiligen blauen Knöpfe aktiviert werden, worauf der Aufnahmeknopf rot und der Wiedergabeknopf grün wird.

3.1.2 Ordnerpfade und Dateinamen definieren

Für eine erfolgreiche Aufnahme muss ein Ordnerpfad für das Summensignal der 4 Mikrofone sowie ein Ordnerpfad für die Aufnahmen der einzelnen Mikrofone angegeben werden. Die Ordner können über das Auswahlménü der Benutzeroberfläche erstellt werden, effektiver können die Ordner allerdings über den systemeigenen Explorer erstellt werden. Für die weitere Nutzung durch Hilfsprogramme ist es angebracht die Ordner für Summensignale mit einem "C" beginnen zu lassen und die Ordner für die Aufnahmen der einzelnen Mikrofone mit einem "E".

Bei diesem Experiment wurde für jede Münzklasse ein Ordner angelegt, was jedoch nicht zwingend nötig ist, da die Klassen durch die Dateinamen definiert werden.

Neben den Ordnerpfaden kann ein Dateinamenprefix angegeben werden. Die Dateien werden durch das Programm automatisch nummeriert, vor die vierstellige Zahl wird die Eingabe des Feldes "Dateiname" eingefügt. Achtung, bestehende Dateien werden ohne Rückfrage überschrieben.

Für die spätere Verwendung durch Hilfsprogramme ist es definiert, dass die Klasse durch eine Zeichenkette vor einem Unterstrich "_" angegeben wird. Darauf können weitere Zeichen folgen, jedoch kein Unterstrich. So wurden bei den späteren Versuchen die einzelnen Münzen innerhalb einer Klasse mit dem Anfangsbuchstaben ihres Landes und ihrem Index nach dem Unterstrich benannt. Als Trennzeichen zu der Nummerierung wurde ein Trennstrich genutzt. Die Endgültige Dateibezeichnung für die 60. Aufnahme einer deutschen 50-Cent Münze mit dem Index 3 war demnach "C050_3D-0059.WAV" wobei alles hinter dem Bindestrich durch das vi "Mikrofonfeld 3.0" eingefügt wurde. Der Dateiname der aktuellen Aufnahme wird im Feld "aktuelle Datei" angezeigt.

Einzelaufnahmen erhalten von dem vi zusätzlich hinter der aktuellen Aufnahmenummer

3 Genutzte Programme

die Bezeichnung des jeweiligen Mikrofons ai0-ai3. Die Speicherung der einzelnen Kanäle kann nicht abgeschaltet werden und dient als Backup, sollte die Summenaufnahme fehlerhaft sein. Die zugehörige Aufnahme des 3. Mikrofons hätte bei oben genannter Datei die Bezeichnung "C050_0059ai2.WAV"

Sollte die Angabe der Ordnerpfade falsch sein, wird ein Dateifehler in dem dazugehörigen Kasten angezeigt, gleiches gilt, wenn die Dateien nicht geschrieben oder überschrieben werden können.

Für den Wiedergabemodus sind keine Ordnerpfade oder Dateinamen nötig, für eine reine Wiedergabe sollten allerdings nicht mehr als 1000 Dateien angefertigt werden, da sonst der Arbeitsspeicher nicht ausreicht.

3.1.3 Kanäle aktivieren und Aussteuerung einstellen

Um eine sinnvolle Aufnahme zu realisieren müssen die gewünschten Kanäle über die Schiebeschalter an den jeweiligen Kanal aktiviert werden. Aktivierte Schiebeschalter haben einen grünen, deaktivierte einen roten Hintergrund. Ob die Kanäle aktiv sind, kann außerdem an den Signalverläufen 0-3 überprüft werden, an welchen bei aktivierten vi die Pegelverläufe der Mikrofone angezeigt werden.

Zusätzlich muss mit den beiden Schiebereglern der Pegel der aufgenommenen Signal angepasst werden. Achtung, zu Beginn ist die Verstärkung auf Null eingestellt, das heißt es werden lediglich 0 Werte aufgenommen. Für die drei Versionen des Versuchs hat sich ein optimaler Wert von 0,05 für die Summe und 0,1 für die Einzelkanäle ergeben, sodass weder Übersteuerung noch ein zu geringes Signal auftraten.

Die Aussteuerung kann an den Signalverläufen Summe sowie Einzelkanal überprüft werden, für eine zuverlässige Überprüfung sollten allerdings die erstellten Wav-Dateien mit einem externen Audio-Bearbeitungsprogramm wie "Audacity" oder "Audacious" geöffnet werden.

3.1.4 Wiedergabeparameter

Für eine Ausgabe des aufgenommenen Signal muss wie bereits erwähnt der Wiedergabemodus aktiviert sein. Für eine erfolgreiche Wiedergabe muss außerdem der Pegel der Ausgabe am Drehregler eingestellt werden, in der Voreinstellung liegt dieser bei 0, ist

3 Genutzte Programme

somit nicht hörbar. Außerdem kann das wiedergebende Audio-Gerät durch einen Index gewählt werden. Nicht vergebene Indexe führen zu einem Wiedergabefehler der in dem entsprechenden Kasten unter dem Drehregler angezeigt wird. In der Regel sind alle vorhandenen Audio-Geräte in den vorderen Indexen 0, 1 und so weiter zu finden.

3.1.5 Kontinuierlicher und getriggelter Modus

Sowohl die Aufnahme als auch die Wiedergabe kann kontinuierlich oder getriggert erfolgen. Beim kontinuierlichen Modus wird jede Sekunde ein Signal erzeugt.

Beim getriggerten Modus wird das Signal ab einem **TODO: bestimmten** Pegel der überschritten wird, z.B. wenn die Münze auf den jeweiligen Untergrund fällt, für eine Sekunde erfasst. Dies ist besonders für die Aufnahme von Messwerten sinnvoll. Es kann durch die Triggerung vorkommen, dass Aufnahmefehler entstehen, da zu lange Zeiträume ohne erkanntes Signal als Fehler gedeutet werden. Werden Dateien aufgenommen entstehen dabei 1 Kilobyte große Dateien ohne Inhalt.

3.1.6 Dateiformat

Die Signale werden mit 48 kHz in 16 Bit abgetastet und werden auf einem Kanal ausgegeben. Außerdem sind alle Signale exakt 1 Sekunde lang, was 48.000 Messwerten entspricht.

$$Dateigroesse = \frac{Messwerte \cdot Bitrate \cdot Kanaele}{BitProByte} = \frac{48.000 \cdot 16bit \cdot 1}{8 \frac{bit}{Byte}} = 96kByte \quad (3.1)$$

Die Dateien sind damit 96 Kilobytes groß, wenn Messwerte erfasst wurden und werden unkomprimiert als wave-Datei gespeichert.

3.2 Das Hilfsprogramm Fileoperations

Nachdem die Dateien aufgenommen wurden müssen sie in Filelisten geschrieben werden um vom Trainingsprogramm des UASR-Systems verarbeitet werden zu können. Dafür wurde das Hilfsprogramm "fileoperations" in JAVA geschrieben.

Das Programm hat zwei Funktionen, zum ersten kann man damit Filelisten erstellen, zum zweiten kann man damit auch die Erkennraten des HMM-Trainings in eine mit Tabstopps und Zeilenumbrüche getrennte Textdatei umwandeln.

3.2.1 Filelist erstellen

Um Filelisten zu erstellen muss man das Programm aufrufen und die erste Option mit "1" auswählen. Man kann sich entscheiden, die Summenaufnahmen, die in den dafür angelegten Ordnern, beginnend mit "C", gespeichert sind zu verwenden, oder die Einzelaufnahmen in den anderen Ordnern, beginnend mit "E". Im Falle der Einzelaufnahmen muss man sich entscheiden, von welchem einem bestimmte Mikrofon man die Antworten verwenden möchte. Ausgewählt wird dies mit den Ziffern "0" bis "3".

Danach kann man den vordefinierten Ordner "\$UASR_HOME-data/coins/common/sig" verwenden oder einen eigenen Ordner definieren. Danach kann man noch einen Unterordner angeben, z.B. der vom jeweiligen Versuch. Das Programm wird dabei nur existierende Ordnerpfade akzeptieren.

Der angegebene Ordner wird nun vom Programm durchsucht und alle gefundenen wave-Dateien (bei Einzelaufnahmen jene, die den jeweiligen Mikrofonindex vor der Dateiendung haben) in den oben beschriebenen Ordnern werden in einer Ausgabezeichenkette aufgelistet. Getrennt werden die einzelnen Dateien mit Zeilenumbrüchen (Steuerzeichen "\n"), was in einigen Editoren nicht erkannt wird.

Wie bereits oben erwähnt wird die jeweilige Klasse aus der Zeichenkette vor dem Unterstrich gelesen. Zusätzlich wird die Dateiendung entfernt und der Pfad relativ zum ersten Ordner im Dateipfad mit dem Namen "sig" angegeben.

3 Genutzte Programme

Hier ein Beispiel für zwei Einträge in der Dateiliste bei Selektion von Einzelaufnahmen von Mikrofon 2:

Beispiel 3.1.

```
\VersuchV1\C050\C050_3D-0059ai2 C050
```

```
\VersuchV1\C050\C050_4F-0001ai2 C050
```

Die so erstellte Zeichenkette sollte danach in den vorgegebenen Ordner "\$UASR_HOME-data/coins/common/flsts" unter dem Namen "all.flst" gespeichert werden. Natürlich kann die Datei auch an einen anderen Ort oder unter einem anderen Namen gespeichert werden, man sollte auch auf jeden Fall eine Sicherheitskopie der Datei anfertigen (dazu muss die Filelist nochmal erstellt werden oder man kopiert sie einfach im Explorer), da diese Datei unter dem oben genannten Verzeichnis die aktuelle Arbeitsdatei ist.

Falls eine Datei überschrieben würde, weist das Programm ausdrücklich darauf hin und falls man das Überschreiben verneint kann man einen anderen Ordnerpfad und Dateinamen angeben.

3.2.2 Logdateien auslesen

Nachdem man das HMM-Training durchgeführt hat - dazu benötigte Vorgänge und Skripte werden im Punkt 3.3.4 HMM-Training (S. 13) beschrieben - erhält man eine sehr lange Ausgabe in der Konsole. Diese muss man, um sie mit diesem Programm auszuwerten, in eine Textdatei speichern. Man kann auch die Ausgabe direkt in eine Textdatei umleiten, weiteres dazu ebenfalls unter dem oben genannten Punkt.

Wenn man das hier beschriebene Programm fileoperations.java startet kann man mit "2" die Option "Logdateien auslesen" auswählen. Dazu gibt man den Dateipfad der Logdatei an oder verwendet den vorgegebenen Pfad "\$UASR_HOME-data/coins/common/log", falls die Logdatei dort gespeichert wurde. Danach sollte man, sofern dies noch nicht geschehen, den Dateinamen der Logdatei inklusive Dateiendung angeben.

Die so geladene Datei wird nun auf Zeilen untersucht, die das Wort "Correctnes" enthalten. In diesen Zeilen sind Angaben zur durchschnittlichen Erkennungsrate enthalten sowie zu dem Konfidenzintervall nach oben und unten. Außerdem ist die jeweilige Splitzahl und Iteration angegeben, die als kombinierter Wert ausgelesen werden. Die Werte werden nacheinander, getrennt durch Tabstopps (Steuerzeichen "\t") in eine Zeile geschrieben. Für jede

3 Genutzte Programme

neue Zeile, die "Correctnes" enthält wird ein Zeilenumbruch (Steuerzeichen "\n") eingefügt.

Eine Logdatei mit 0 Splits und 1 Iteration sähe nach der Umwandlung zum Beispiel so aus:

Beispiel 3.2.

0_0	92,3	4,8	8,3
0_1	89,7	5	9

Diesen Text kann man ebenfalls speichern um sie in einem Tabellenverarbeitungsprogramm z.B. "Libre Office Calc" zu importieren und dort grafisch aufarbeiten zu können.

3.3 UASR und dafür angefertigte Skripte

Das UASR ist wie bereits in der Einleitung beschrieben, das eigentliche Programm für die Mustererkennung. Um die Bedienung zu vereinfachen wurden einige Skripte erstellt, die ich hier zusammen mit dem Programm vorstelle.

3.3.1 Das Skript MAKEFLST.xtp

Mit der Filelist aus Punkt 3.2.1 Filelist erstellen (S.9) kann man noch nicht das HMM-Training starten, man benötigt vorher noch zwei Filelisten, die die Auswahl der Test- und Trainingsdaten beinhalten. Dafür wurde das Skript MAKEFLST.xtp entwickelt, welches die Datei "all.flst" im Pfad "\$UASR_HOME-data/coins/common/flsts" verwendet. Deswegen ist es im oben genannten Punkt so wichtig gewesen, die Datei in diesem Pfad zu speichern. Das Skript erstellt zwei weitere Filelisten in dem Unterverzeichnis "\$UASR_HOME-data/coins/common/flsts/tmp" mit den Namen "test.flst" und "train.flst". Die Anzahl der für das Training genutzten Dateien kann man als Parameter einstellen, es sollten circa 80% - 90% der Testdaten sein. Das Skript nutzt dabei das Unterskript "FLST.xtp" des UASR um die Dateien zufällig aus der "all.flst" auszuwählen und in die neuen Filelisten zu erstellen. Nach der Generierung aller Filelisten sollte sowohl die "all.flst" als auch "train.flst" und "test.flst" in den Ordner "flsts" des jeweiligen Experimentes verschoben werden.

3.3.2 Konfigurationsdatei

Einstellungen wie zum Beispiel Samplerate der wave-Dateien, Anzahl der Split-Punkte oder Anzahl der jeweiligen Iterationen müssen in einer Konfigurations-Datei vorgenommen werden. Eine empfohlene Bezeichnung ist "HMM-trn.cfg", zu speichern sind die Dateien im jeweiligen "info"-Verzeichnis des dazugehörigen Versuchs.

Ist keine Konfigurationsdatei vorhanden wird die globale Konfigurationsdatei "default.cfg" im Verzeichnis "\$UASR_HOME-data/coins/common/info" verwendet.

Spezielle Konfigurationen für den Versuch sollten jedoch in einer eigenen Konfigurations-Datei vorgenommen werden.

3.3.3 FEA Analyse

Bevor das HMM-Training gestartet wird, ist es empfehlenswert eine Merkmalanalyse zu starten. Diese wandelt die große Datenmenge der wave-Dateien in eine kleinere Datenmenge von Merkmalvektoren, die sich schneller verarbeiten lassen.

Gestartet wird die Analyse mit Eingabe 3.3 oder einer entsprechenden RUN-Configuration in Eclipse.

Eingabe 3.3.

```
dLabPro FEA.xtp ana [cfg-Datei]
```

Als cfg-Datei ist die Konfigurationsdatei anzugeben, welche zudem den Namen des Experimentes enthält. Zu finden ist diese Datei in der Regel im Verzeichnis "info" des Versuchs. Als filelist werden die Dateien im Verzeichnis "flists" des genannten Experimentes verwendet, in welchem die Analyse ausgeführt wird. Existieren diese nicht, werden die filelists aus dem Verzeichnis "\$UASR_HOME-data/coins/common/flists" verwendet.

3.3.4 HMM-Training

Nun kann das HMM-Training gestartet werden. Gestartet wird das Programm mit der Eingabe 3.4 oder einer entsprechenden RUN-Configuration in Eclipse.

Eingabe 3.4.

```
dLabPro HMM.xtp trn [cfg-Datei]
```

Genauso wie bei der FEA Analyse ist auch hier in der Konfigurationsdatei das jeweilige Experiment anzugeben, woraus sich die Positionen der verwendeten Dateien ergeben. Es sollten für die Analyse und das Training dieselben Konfigurationsdateien verwendet werden, da sonst die Analyse sinnlos ist.

3 Genutzte Programme

Komplexe Tabellen sind nicht sehr einfach:

		dies			
		von dort	und dort	über hier	zu Los
das	hier	bla	bla	bla	bla
	dort	bla	bla	bla	bla
	da	bla	bla	bla	bla

Tabelle 3.1: textbeschr

A Anhang

A.1 Quelltexte

Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, dass alle Stellen der Arbeit, die wörtlich oder sinngemäß aus anderen Quellen übernommen wurden, als solche kenntlich gemacht und dass die Arbeit in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegt wurde.

Ort, Datum

Unterschrift