



Ludwig-Maximilians-Universität München  
Institut für Informatik  
Lehr- und Forschungseinheit für Datenbanksysteme



# **Probabilistische Ähnlichkeitssuche für Audiosequenzen zur Unterstützung von Query-by-Humming Anwendungen**

## **Diplomarbeit**

bearbeitet von: **Markus Feil, Angerweg 10, 83071 Stephanskirchen**

Bearbeitungszeitraum: **18.8.2008 bis 17.2.2009**

Betreuer: **Dr. Matthias Renz, Andreas Züfle**

Aufgabensteller: **Prof. Dr. Hans-Peter Kriegel**

## Abstrakt

Eine Query-by-Humming Anwendung ist ein System, das einer gesummten Melodie die ähnlichsten Lieder aus einer Menge von Liedern, die in einer Datenbank gespeichert sind, zuordnet.

Im Rahmen dieser Diplomarbeit werden verschiedene neue Algorithmen für die einzelnen Module eines QbH Systems entwickelt und verglichen. Ein Hauptfokus liegt auf der automatisierten Erkennung von Noten.

Wissenschaftliche Fortschritte konnten dabei in folgenden Bereichen erreicht werden:

- Für die Vorverarbeitung wurde ein Filter entwickelt, das die Notenerkennungsrate für gesummtes Audiomaterial bei männlichen Sängern um 39% steigert.
- Für die Grundfrequenzsuche wurde die AKF+COMB Transformation zur Erzeugung von „Grundfrequenz Spektrogrammen“ entwickelt. Im Gegensatz zum klassischen, Fourier-basierten Spektrogramm wird hier eine lineare Zeitreihe nicht nach Sinusschwingungen, sondern nach beliebig geformten Zyklizitäten durchsucht. Durch die AKF+COMB Methode konnte die Grundfrequenzerkennungsrate der Autokorrelation für gesummtes Material um 28% gesteigert werden. Der Algorithmus erreicht in Kombination mit dem Filter aus der Vorverarbeitung eine Notenerkennungsrate von 98%.
- Es konnte ein Algorithmus auf Basis der Autokorrelation gefunden werden, der die automatisierte Takterkennung bei synthetischen Audiomaterial effizient löst. Dabei wird ein Algorithmus vorgestellt, der auf einer linearen Zeitreihe die Phasenlage und Amplitude einer beliebigen Frequenz in linearer Laufzeit berechnet.
- Für die automatisierte Melodieerkennung wird ein Algorithmus vorgestellt, der zuverlässig kontinuierliche Grundfrequenzverläufe in diskrete Notenwerte quantisiert.
- Für den Melodievergleich konnte eine Distanztabelle gefunden werden, welche die Erkennungsqualität steigert.
- Es wurde ein QbH System entwickelt, das den Titel von Melodien automatisiert erkennen kann.

# Aufgabenstellung

## Probabilistische Ähnlichkeitssuche für Audiosequenzen zur Unterstützung von Query-by-Humming Anwendungen

### Motivation:

Query by Humming Anwendungen sollen ein altbekanntes Problem lösen: Der Sänger hat eine Melodie im Kopf, aber kennt dabei den Interpreten und den Titel des Liedes nicht. Durch das QbH Systems können Musiktitel auf Grundlage gesungener Melodien automatisiert erkannt werden.

Im ersten Schritt wird die Melodie in ein Mikrofon gesummt. Der Gesang wird digitalisiert und in eine Folge von Noten umgewandelt. Die Melodie wird dann mit Hilfe eines Verfahrens zum Melodievergleich mit allen in einer Datenbank gespeicherten Melodien verglichen und ein Ähnlichkeitswert berechnet. Die zur gesungenen Eingabe ähnlichsten Melodien werden in Form einer sortierten Liste ausgegeben.

### Aufgabenstellung:

In diesem Projekt sollen verschiedene Methoden zur Verbesserung von Query-by-Humming Systemen entwickelt werden. Der Schwerpunkt der Arbeit liegt auf der automatisierten Erkennung von Melodien. Die zu entwickelnden Methoden sollen eine hohe Effektivität und Robustheit aufweisen. Eventuell ist es sinnvoll, auf Methoden und Werkzeuge zurückzugreifen, die bereits am Lehrstuhl entwickelt wurden.

## **Selbstständigkeitserklärung**

**Ich erkläre hiermit, dass ich die vorliegende Arbeit selbstständig angefertigt habe, alle Zitate als solche kenntlich gemacht, sowie alle benutzten Quellen und Hilfsmittel angegeben habe.**

Rosenheim, den 20.12.2008

---

Markus Feil

# Inhaltsverzeichnis

1	Einleitung.....	1
2	Gesamtkonzept.....	2
3	Aufnahme.....	4
3.1	Die Wahl des richtigen Mikrophons.....	4
3.2	Sampling.....	4
3.3	Dateiformat.....	5
3.4	Import von gesumnten Melodien.....	5
4	Vorverarbeitung.....	6
4.1	Abstrakt.....	6
4.2	Problemstellung.....	6
4.3	Vorverarbeitung von gesumnten Melodien.....	7
4.3.1	Evaluierung des Filters.....	11
4.4	Vorverarbeitung von gesungenen Melodien.....	13
5	Grundfrequenzextraktion.....	14
5.1	Abstrakt.....	14
5.2	Problemstellung.....	14
5.3	Grundfrequenzsuche mit Hilfe der Fourier Transformation.....	16
5.3.1	Erstellung eines Spektrogramms mit Hilfe der Fourier Transformation.....	16
5.3.2	Logarithmieren des Spektrogramms.....	17
5.3.3	Durchsuchen des Spektrums nach kammartigen Strukturen.....	17
5.3.4	Die Wahl der Kreuzkorrelationsfunktion.....	19
5.3.5	Faltung.....	20
5.3.6	Komplexität.....	20
5.3.7	Schwächen der Methodik.....	20
5.4	Grundfrequenzsuche mit dem AKF+COMB Algorithmus.....	22
5.4.1	Spektrale Eigenschaften des zu suchenden Signals.....	22
5.4.2	Die Wahl eines Testsignals .....	25
5.4.3	Der AKF+ Algorithmus - die verbesserte Autokorrelation.....	26
5.4.4	Vorverarbeitung.....	26
5.4.5	Autokorrelation.....	27
5.4.6	Autokorrelation mit Fensterung.....	28
5.4.7	Autokorrelation mit Fensterung und Offsetkorrektur (AKF+).....	28
5.4.8	Komplexität und Optimierung der AKF+.....	30
5.4.9	Autokorrelation mit Fensterung, Offsetkorrektur und Kammfilterung (AKF+COMB).....	30
5.4.10	Vergleich zwischen AKF und AKF+COMB Spektrogrammen.....	33
5.4.11	Komplexität und Echtzeitfähigkeit.....	36
5.4.12	Evaluierung .....	36
6	Automatisierte Rhythmuserkennung.....	38
6.1	Abstrakt.....	38
6.2	Problemstellung.....	38
6.3	Rhythmuserkennung auf FFT Basis.....	39
6.3.1	Schwächen der Methodik.....	40
6.4	Takterkennung auf Basis der Autokorrelation.....	41
6.4.1	Bestimmung der Taktgeschwindigkeit.....	41
6.4.2	Bestimmung des Taktzeitpunkts.....	43
6.4.3	Komplexität.....	44

6.4.4	Anwendung in der Praxis.....	44
7	Melodieextraktion.....	48
7.1	Abstrakt.....	48
7.2	Die Auswirkung der Unschärferelation.....	48
7.3	Unsichere Noten.....	49
7.4	Rasterung.....	50
7.5	Logarithmierung der Wahrscheinlichkeiten.....	51
7.6	Die richtige Wahl des k-Wertes .....	52
7.7	Export.....	52
7.8	Weitere Verbesserungsmöglichkeiten.....	52
8	Datenbank.....	53
8.1	Abstrakt.....	53
8.2	Midi.....	53
8.3	Import von Midi Daten.....	54
8.3.1	Realisierung.....	54
9	Melodievergleich.....	56
9.1	Abstrakt.....	56
9.2	Problemstellung.....	56
9.3	Timestretching .....	57
9.4	Ähnlichkeitsberechnung.....	57
9.5	Matchingalgorithmen.....	58
9.6	Binäres Matching.....	59
9.6.1	Abstrakt.....	59
9.6.2	Funktionsweise.....	59
9.6.3	Komplexität.....	59
9.6.4	Schwächen der Methodik.....	60
9.7	Matching mit Ableitungsfunktion.....	60
9.7.1	Abstrakt.....	60
9.7.2	Funktionsweise.....	60
9.7.3	Komplexität.....	61
9.7.4	Schwächen der Methodik.....	61
9.8	Matching mit gewichteter Distanzfunktion.....	62
9.8.1	Abstrakt.....	62
9.8.2	Funktionsweise.....	62
9.8.3	Distanztabelle.....	63
9.8.4	Komplexität.....	65
9.8.5	Schwächen der Methodik.....	66
9.9	Vergleich der Matchingalgorithmen.....	66
9.9.1	Binäres Matching.....	66
9.9.2	Matching mit Ableitungsfunktion.....	67
9.9.3	Matching mit gewichteter Distanzfunktion.....	67
10	Sortierung und Ranking.....	68
10.1	Abstrakt.....	68
10.2	Funktionsweise.....	68
11	„Hummel“ - eine praktische Umsetzung eines QbH Systems.....	69
11.1	Probleme bei der Implementierung von QbH Systemen in der Praxis.....	70
12	Zusammenfassung.....	71
13	Ausblick.....	72
14	Glossar.....	73
15	Inhaltsverzeichnis der beigelegten CD.....	76

16 Literaturverzeichnis.....77



# 1 Einleitung

Alltägliche Dinge, die einem Menschen trivial erscheinen, stellen bisweilen für Maschinen eine große Herausforderung dar.

Das automatisierte Erkennen von Melodien ist eine dieser Herausforderungen, die bis heute noch nicht zufriedenstellend gelöst wurden.

Query by Humming (QbH) liefert die Lösung für ein altbekanntes Problem: Man hat eine Melodie im Kopf, kann jedoch keinen Interpreten oder Titel zuordnen. Mit Hilfe des Melodieerkennungssystems QbH können Musiktitel auf Grundlage gesungener oder anderer monophoner Melodien identifiziert werden.

Die Software analysiert dann die melodischen und rhythmischen Eigenschaften der digitalisierten Melodie und durchsucht eine Datenbank nach möglichen Stücken, aus denen die Melodie stammen könnte.

Die Erforschung und Entwicklung von Query-by-Humming Systemen ist ein interdisziplinäres Gebiet.

Es erfordert Fachkenntnisse in folgenden Bereichen:

- Medieninformatik
- Digitale Signalverarbeitung
- Hardwarenahe Programmierung
- Phonetik
- Musiktheorie
- Statistik
- Effiziente Algorithmen
- Datenbanken

## 2 Gesamtkonzept

Im Umfang dieser Diplomarbeit sollten einzelne Module zur Unterstützung von QbH Systemen entwickelt werden. Diese lassen sich zwar separat implementieren, aber um deren Leistungsfähigkeit hinsichtlich der Erkennungsrate sinnvoll testen zu können, ist es nötig, ein komplettes System zu betrachten.

Daher musste, bis auf das Sampling Modul, ein komplettes, experimentelles QbH System mit dem Projektnamen „Hummel“ implementiert werden. Ein Hauptfokus dieser Diplomarbeit liegt auf der automatisierten Erkennung von Noten. Die Notendetektion beeinflusst die Erkennungsqualität von QbH Systemen stark, da sie die Basisdaten für die weiteren Verarbeitungsmodule liefert.

Im folgenden werden die einzelnen Module und die dafür neu entwickelten Algorithmen beschrieben.

Das QbH System „Hummel“ lässt sich grob in diese Module zerteilen:

- Vorverarbeitung
- Grundfrequenzerkennung
- Rhythmuserkennung
- Melodieextraktion
- Melodievergleich
- Sortierung und Ausgabe

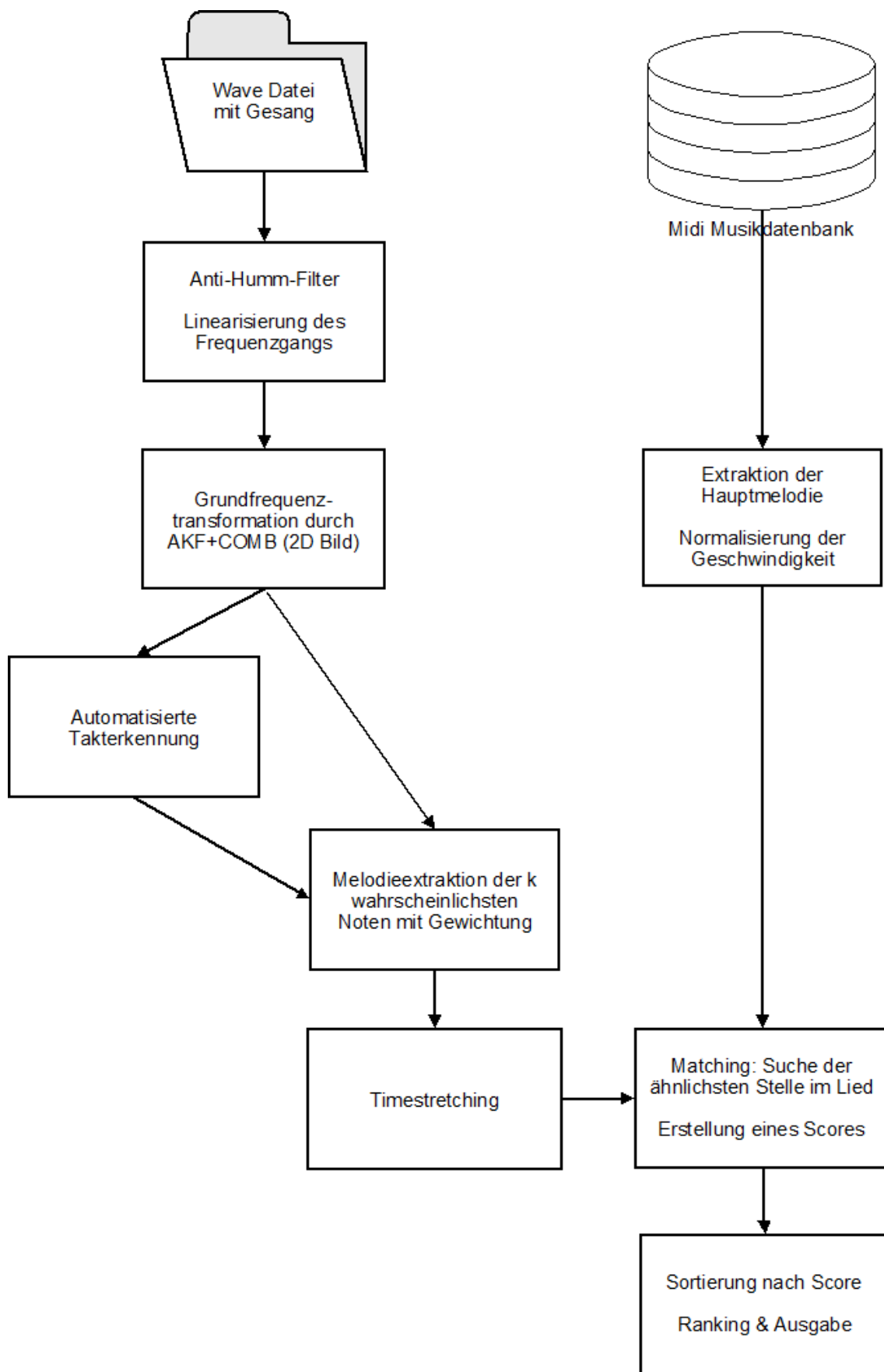


Abbildung 2.1: Gesamtkonzept des Query-by-Humming Systems „Hummel“

## **3 Aufnahme**

Die Qualität der Aufnahme ist von entscheidender Bedeutung. Hierbei können eine große Anzahl von Störungen auftreten: Rauschen, Hintergrundgeräusche, Reflexionen, Rückkoppelungen, Verzerrungen, Übersteuern, nicht-Linearität des Frequenzgangs und Atemgeräusche.

### **3.1 Die Wahl des richtigen Mikrophons**

Bereits die Wahl des richtigen Mikrophons hat deutliche Auswirkungen auf die spätere Erkennungsqualität. Ziel ist es hierbei, die Störgeräusche so gering wie möglich zu halten und die größtmögliche Linearität in der Aufnahme zu erreichen.

Bei den Versuchen erreichten wir gute Ergebnisse mit Headset- und Popschirmmikrophonen.

Atem- und Popgeräusche bei der Aufnahme verschlechtern vor allem die später erfolgende Rhythmuserkennung.

Nichlinearitäten im Frequenzgang können zu Oktavfehlern in der Grundfrequenzerkennung führen. Hier kann ein Equalizer helfen, die Mängel im Frequenzgang auszugleichen.

### **3.2 Sampling**

Die Person summt eine etwa 10 Sekunden lange Melodie in das Mikrophon. Dabei wird die akustische Schallenergie in elektrische Spannung gewandelt. Die elektrische Spannung wird anschließend mit einem DA-Wandler in diskrete Zahlenwerte gewandelt (Sampling).

Die Abtastung muss mit mindestens 44kHz erfolgen, um alle für das menschliche Ohr hörbaren Frequenzen in der Messung zu erfassen. Eine saubere Umrechnung auf niedrigere Abtastraten zu Optimierungszwecken kann gegebenenfalls später erfolgen. Eine gängige Bit-Tiefe für das Sampling beträgt 16 Bit. Sie sollte nicht niedriger gewählt werden, weil sonst schwer filterbares Quantisierungsrauschen auftritt. Da das Signal aus dem Kehlkopf, einer einzelnen Schallquelle entstammt, reicht ein mono Recording aus (1 Kanal).

### **3.3 Dateiformat**

Als Dateiformat zur Speicherung der gesumnten Melodie eignet sich die WAV-Datei, weil sie von einem Großteil der am Markt erhältlichen Audiosoftware unterstützt wird, keine rechtlichen Probleme bestehen und sie sehr etabliert und gut dokumentiert ist. In unserem System wurde in 16 Bit Datentiefe mit einer Abtastrate von 44Khz in verlustfreier PCM mono Codierung gespeichert. Zur Speicherung von einer 10 Sekunden langen Melodie sind 900 kByte nötig.

Verlustbehaftete Codierung kann die Extraktionsqualität in der Weiterverarbeitung negativ beeinflussen.

### **3.4 Import von gesumnten Melodien**

Der Import der gesumnten Melodie kann theoretisch aus jedem beliebigen samplingbasierten Dateiformat erfolgen. Da der Import nur einmalig pro Suche erfolgt, spielt die Performance und Komplexität hier eine untergeordnete Rolle. Beim Importieren ist die Abtastrate, die Kanalzahl und Bittiefe auf ein einheitliches Zielformat umzurechnen.

In unserem System erfolgt eine Umwandlung auf ein 44kHz mono Format mit 32 Bit float Datentyp. Das float Format eignet sich gut für die digitale Signalverarbeitung, weil es nicht anfällig für Overflows ist und einen sehr großen Wertebereich mit ausreichender Genauigkeit erfasst.

Bei der Umrechnung der Abtastrate ist darauf zu achten, dass das Shannonsche Abtasttheorem nicht verletzt wird, also keine Aliasing-Artefakte auftreten. Um die größtmögliche Qualität zu erzielen, kann das Signal für das Resampling mit einem steilen, bandbreitenbegrenzten FIR Filter vorgefiltert werden.

Um die Weiterverarbeitung zu erleichtern, wird der Pegel beim Import auf +/- 1 normiert. +1 entspricht dem maximalen erreichbaren positiven Signalpegel und -1 dem maximalen negativen Signalpegel. Zur Normierung des Signals wird das gesamte Signal nach dem Absolutwert des maximalen Ausschlags durchsucht. Alle Werte werden dann durch diesen Maximalwert dividiert.

Beim Import von Stereodaten wird der Mittelwert aus linkem und rechtem Kanal gebildet.

## 4 Vorverarbeitung

### 4.1 Abstrakt

Für die Vorverarbeitung wurde ein Filter zur spektralen Einebnung entwickelt, das die Frequenzantwort von gesummtem Material glättet. Dadurch steigert sich die Notenerkennungsrate bei männlichen Sängern um 39%.

### 4.2 Problemstellung

Im Vorverarbeitungsschritt sollen möglichst viele Störgeräusche aus dem Signal entfernt werden. Nur die, für die Weiterverarbeitung relevanten Informationen sollen erhalten bleiben.

*„Es bietet sich an, den Analyseprozess in verschiedene Stufen zu unterteilen. In den einzelnen Stufen wird a priori Wissen für die jeweilige Verarbeitung auf einem bestimmten Abstraktionsgrad genutzt. Die erste Stufe benutzt bekannte Algorithmen der Signalverarbeitung, z. B. zur Filterung des Signals. Die zweite Stufe führt eine syntaktische Analyse durch.“ [WIL 03]*

In der Vorverarbeitung kann die Hinzunahme von Weltwissen helfen, die Datenqualität zu verbessern:

- Die Information, ob die Melodie gesummt oder gesungen ist
- Das Geschlecht des Sängers
- Den maximalen Grundfrequenzbereich der menschlichen Stimme
- Information über Messfehler in der Frequenzantwort und Linearität des Mikrophons und der Hardware
- Wissen über die musikalische Kompetenz des Sängers

*„Ein wesentliches Problem der GFB ist die Empfindlichkeit gegenüber stark ausgeprägten ersten Formanten, wenn diese mit der 2. oder 3. Teilschwingung zusammenfallen. Diesem Problem kann man am ehesten durch das Verfahren der spektralen Einebnung entgegenwirken.“ [Hes 05]*

Unser Sprechapparat wird durch das Quelle-Filter Modell ausreichend genau modelliert. Dabei dient die Glottis (Stimmritze im Kehlkopf) als Schallquelle. Durch die Formung des Rachenraumes wird die Frequenzantwort dieser Schallquelle verändert.

### 4.3 Vorverarbeitung von gesummen Melodien

Beim Summen ist die Formung des Rachenraumes und somit auch die Frequenzantwort (Formantfrequenzen) statisch. Es wird ein stimmhafter Laut oder kein Laut gebildet.

Die starke Formantfärbung beim Summen beeinflusst die Qualität der Grundfrequenzextraktion negativ. Bei 200 Hz verstärkt der erste Formant (F1) das Signal um bis zu 10dB mit einer Bandbreite von 50Hz. Bei der Grundfrequenzextraktion führt dies häufig zu Oktavfehlern. Es wird daher nicht die Grundfrequenz, sondern die doppelte Grundfrequenz erfasst, was der ersten harmonischen Oberschwingung entspricht. Bei 1000Hz ist das Spektrum mit einer Bandbreite von 200Hz um 6dB eingedellt. Bei 2kHz wird das Signal um 10dB mit einer Bandbreite von 800Hz angehoben. Die Ermittlung dieser Werte erfolgte dadurch, dass von drei männlichen und weiblichen Sängern ein Sweep von 400Hz bis 90Hz gesungen wurde. Anschließend wurde die spektrale Energieverteilung visualisiert und analysiert.

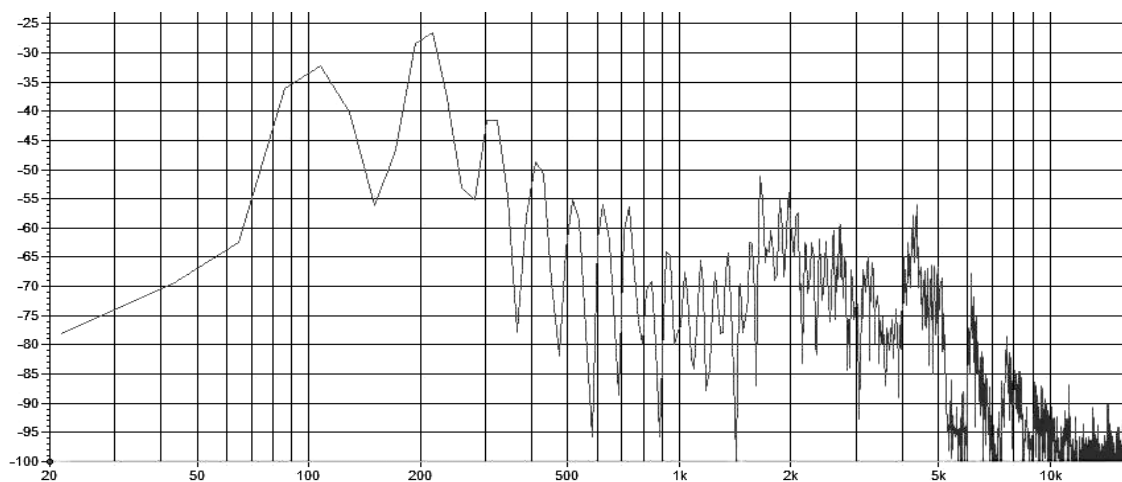


Abbildung 4.1: Spektrum eines männlichen Summers bei  $F_0=103\text{Hz}$ ; X-Achse: Frequenz (0-20kHz), Y-Achse: Pegel in dB

Die Daten des ermittelten Frequenzgangs wurden in einen Equalizer einprogrammiert. Ein Dirac-Impuls wurde hindurch geschickt und die Impulsantwort aufgezeichnet.

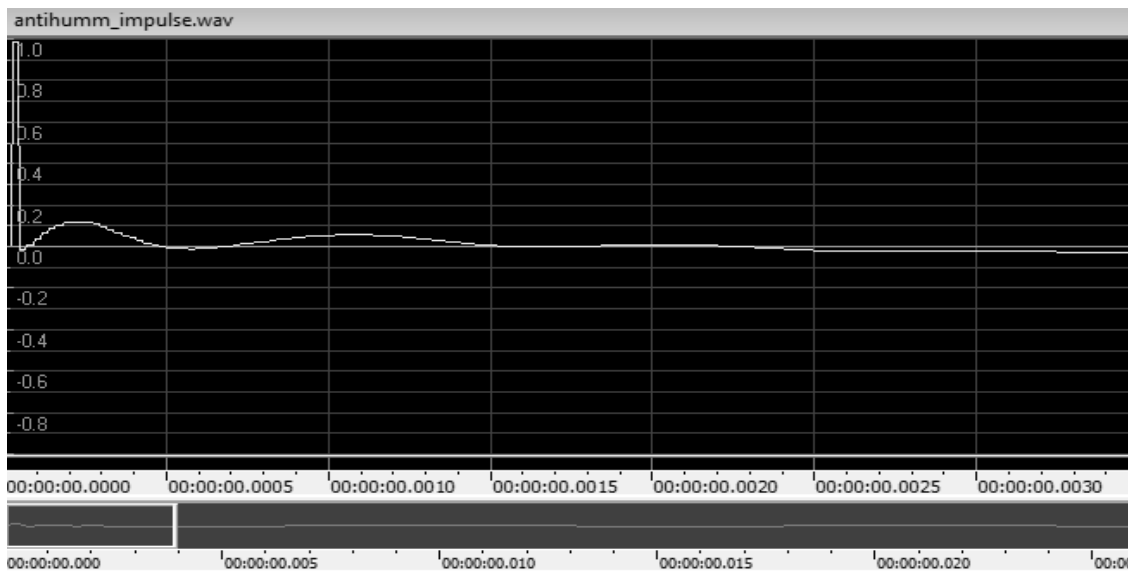


Abbildung 4.2: Impulsantwort des Antiformant-Filters; X-Achse: Zeit (0-0,003Sek), Y-Achse: Pegel

Die Impulsantwort konnte nun als FIR Filterkoeffizienten verwendet werden. In „Hummel“ wird ein FIR Filter der Ordnung 1303 verwendet. Aus dem Bild ist erkennbar, dass die Koeffizienten nicht symmetrisch sind und wir einen IIR Equalizier verwendet haben. Nicht symmetrische Koeffizienten führen zu frequenzabhängigen Phasenverschiebungen. Ein phasenneutrales Filter kann zu einer zusätzlichen Verbesserung der späteren Extraktionsqualität führen. Ein symmetrisches FIR Filter hoher Ordnung ließ sich mit Signal Processing Toolbox der verwendeten Matlab Version aufgrund von Programmfehlern nicht erzeugen.

Die Differenzgleichung für die Filterantwort lautet für ein System m-ter Ordnung im Zeitbereich:

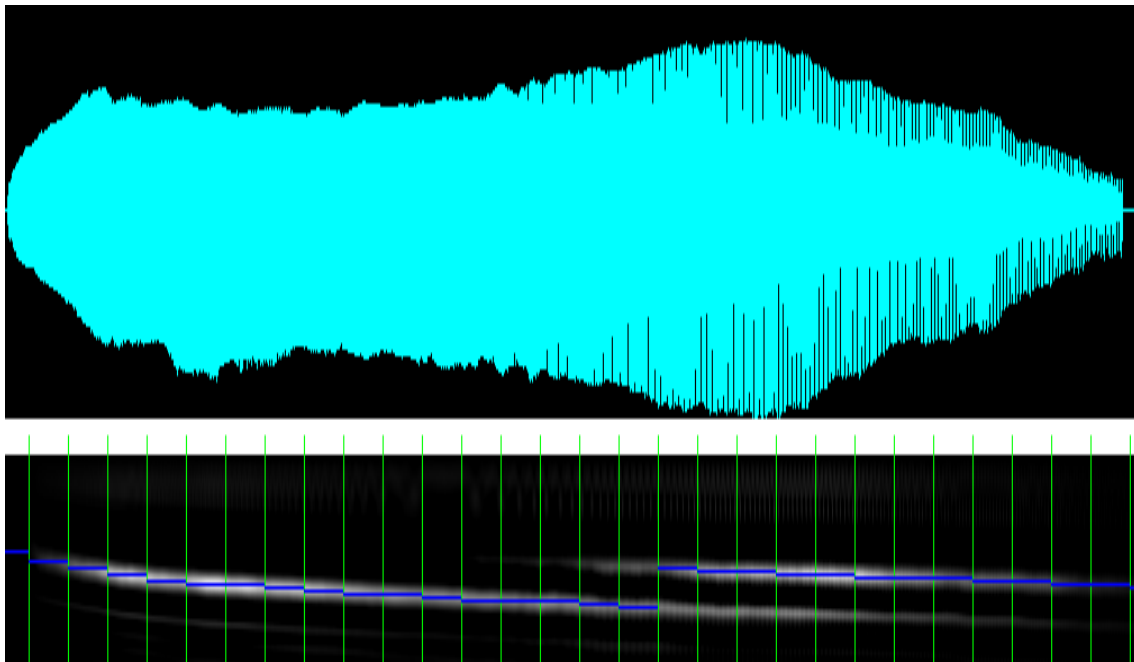
$$y[k] = \sum_{i=0}^m h(i) \cdot u_{m-i}$$

Die dabei auftretenden Faktoren  $h(i)$  stellen die Filterkoeffizienten dar.

Vereinfachte Implementierung des FIR Filters in  $O(n^2)$ :

```
for (int i=0;i<numSamples;i++)
{
    float sum=0;
    for (int j=0;j<FilterOrdnung;j++)
    {
        long ofs= i+j;
        sum += sample[ofs] * filterKoeffizient[j];
    }
    sampleNew[i]=sum;
}
```

Für die folgenden Abbildungen wurde von einer Versuchsperson ein 8 Sekunden langer „Sweep“ gesummt. Die Versuchsperson musste mit dem höchsten Ton, den sie singen konnte, beginnen und dann die Grundfrequenz kontinuierlich bis zum tiefsten Ton, den sie singen konnte, verringern.



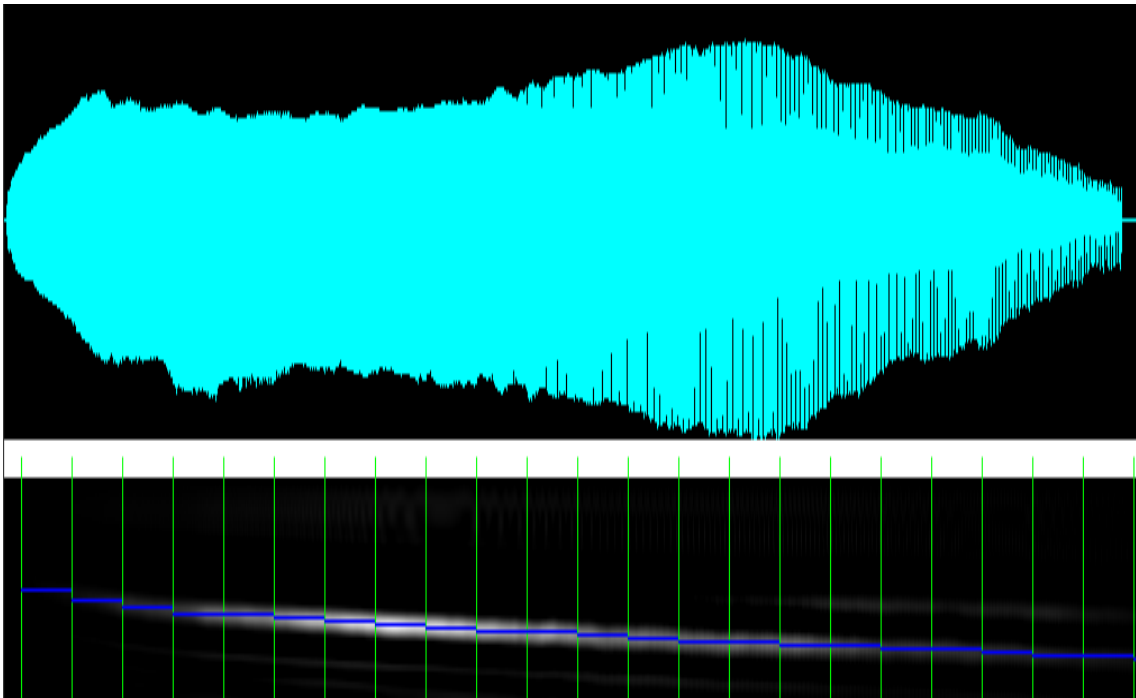
*Abbildung 4.3: Grundfrequenzsuche ohne Filter*

*Oben: Audiosignal eines gesumnten, männlichen „Sweeps“ (400-100 Hz); X-Achse: Zeit (0-8 Sek.); Y-Achse: Amplitude*

*Unten: Notenextraktion aus dem AKF+COMB Spektrogramm für  $k=1$ ; X-Achse: Zeit (0-8 Sek.); Y-Achse: Note*

In der zweiten Hälfte des oberen Bildes wird die Grundfrequenz aufgrund der Frequenzanhebung des ersten Formanten falsch erkannt (blaue Linie). Es wird die doppelte Frequenz erkannt.

*„Die Autokorrelationsfunktion wiederum ist bei der GFB gegen Rauschen sehr unempfindlich, gegenüber dominanten Formanten jedoch recht empfindlich.“  
[Hes 05-2]*



*Abbildung 4.4: Grundfrequenzsuche mit vorgeschaltetem „Antihumm-Filter“ zur spektralen Glättung*

*Oben: Audiosignal eines gesummierten, männlichen „Sweeps“ (400-100 Hz); X-Achse: Zeit (0-8 Sek.); Y-Achse: Amplitude*

*Unten: Notenextraktion aus dem AKF+COMB Spektrogramm für  $k=1$ ; X-Achse: Zeit (0-8 Sek.); Y-Achse: Note*

Durch die spektrale Glättung des „Antihumm-Filters“ konnten die Frequenzverdopplungsfehler behoben werden. Die blaue Linie folgt nun korrekt dem kontinuierlichen Grundfrequenzverlauf.

### 4.3.1 Evaluierung des Filters

Verschiedene Sänger haben eine Melodie in das Mikrofon gesummt. Aus den aufgenommenen Audiodaten wurden die Grundfrequenz mithilfe der AKF+COMB Methode bestimmt. Anschließend wurde eine Rhythmusextraktion mit der Autokorrelationsmethode durchgeführt. Die aus der Melodieextraktion gewonnenen Noten-Daten für  $k=1$  wurden in eine Midi Datei exportiert und in einen Midi-Sequencer geladen. Die Zahl der „korrekt erkannten Noten“ wurde dann manuell gezählt und ausgewertet.

Die hier ermittelten Werte lassen sich grob auf die meisten autokorrelationsbasierten Algorithmen zur Zyklizitätssuche übertragen. Sie lassen sich jedoch nicht auf Cepstrum-Verfahren übertragen, da diese gegenüber spektralen Einbeulungen weniger empfindlich sind.

	Melodielänge in Noten	Korrekt erkannte Noten ohne Filter	Korrekt erkannte Noten mit Filter	Verbesserung durch das Filter
Melodie 1, Sänger 4, weiblich	14	13	13	0%
Melodie 2, Sänger 5, weiblich	20	19	19	0%
Melodie 3, Sänger 4, weiblich	28	26	26	0%
Melodie 4, Sänger 5, weiblich	17	15	16	7%
Melodie 5, Sänger 5, weiblich	18	17	17	0%
Durchschnittliche Verbesserung				1%

*Tabelle 4.1: Notenerkennung mit und ohne Filter bei weiblichen Sängern*

	Melodielänge in Noten	Korrekt erkannte Noten ohne Filter	Korrekt erkannte Noten mit Filter	Verbesserung durch das Filter
Melodie 1, Sänger 1, männlich	14	10	14	40%
Melodie 2, Sänger 2, männlich	20	11	19	72%
Melodie 3, Sänger 3, männlich	28	16	27	68%
Melodie 4, Sänger 2, männlich	17	14	17	21%
Melodie 5, Sänger 1, männlich	18	12	14	17%
Melodie 6, Sänger 2, männlich	16	14	16	14%
Melodie 7, Sänger 3, männlich	28	21	28	33%
Melodie 8, Sänger 1, männlich	26	19	26	37%
Melodie 9, Sänger 2, männlich	16	13	16	23%
Melodie 10, Sänger 3, männlich	28	13	25	92%
Melodie 11, Sänger 1, männlich	19	13	17	31%
Melodie 12, Sänger 1, männlich	18	12	14	17%
Durchschnittliche Verbesserung				39%

*Tabelle 4.2: Notenerkennung mit und ohne Filter bei männlichen Sängern*

**Durch die spektrale Glättung mit dem „Anti-humm Filter“ konnte die Notenerkennungsrate für männliche Sänger um 39% gesteigert werden.**

**Das Filter hat fast keinen Einfluss auf die Erkennungsrate von weiblichen Sängern. Aufgrund der höheren mittleren Grundfrequenz treten Frequenzverdopplungsfehler bei weiblichen Sängern weniger häufig auf.**

#### **4.4 Vorverarbeitung von gesungenen Melodien**

Beim Singen verändert sich die Form des Rachenraumes und die Frequenzantwort kontinuierlich. Es werden stimmhafte Vokale, stimmlose Konsonanten, stimmhafte Konsonanten oder keine Laute erzeugt.

Die Grundfrequenzextraktion von Gesungenem ist technisch schwieriger, so dass hier keine Annahme über die Form des Vokaltraktes gemacht werden kann. Bei stimmlosen Lauten kann keine Grundfrequenz extrahiert werden. Die störenden Formanten lassen sich hier mit einer weißen Filterung entfernen. Dabei wird der Frequenzgang durch adaptives Equalizing so verändert, dass die Energie statistisch über das Spektrum über einen geringen Bereich gleich verteilt ist.

Eine weitere Verbesserungsmöglichkeit für die Datenqualität besteht darin, das Signal mit 60Hz Hochpass zu filtern, weil die menschliche Stimme in der Regel keine Grundfrequenz darunter enthält und somit potentielle Störgeräusche gedämpft werden.

## 5 Grundfrequenzextraktion

### 5.1 Abstrakt

Bei der Grundfrequenzextraktion wird für jeden Zeitpunkt einer linearen Zeitreihe die wahrscheinlichste Zyklizität gesucht.

Für die Grundfrequenzsuche wurde die AKF+COMB Transformation zur Erzeugung von „Grundfrequenz Spektrogrammen“ entwickelt. Im Gegensatz zum klassischen, Fourier-basierten Spektrogramm wird hier eine lineare Zeitreihe nicht nach Sinusschwingungen, sondern nach beliebig geformten Zyklizitäten durchsucht. Der Algorithmus zeichnet sich durch eine um 27% höhere Erkennungsrate für Noten als die Autokorrelation aus. Der AKF+COMB Algorithmus erkennt in 98% der Fälle die Noten korrekt.

Die AKF+COMB Transformation ist ein neues Verfahren und steht in Konkurrenz zu anderen Methoden wie die Wavelet Transformation [Cha 99] [Pop 02] [Wu 00], die diskrete Fourier Transformation (DFT) [Agr 93], Singulärwertzerlegung (SVD) [Kor 97] und zur Piecewise Aggregate Approximation (PAA) [Yi 00].

Das Grundfrequenzerkennungsproblem für Query-by-Humming Anwendungen wurde durch die AKF+COMB Transformation gelöst.

### 5.2 Problemstellung

*„Auf den ersten Blick sieht die Aufgabe einfach aus. Man muss lediglich die Grundfrequenz (oder -periodendauer) eines quasiperiodischen Signals bestimmen. Wenn es sich jedoch um Sprachsignale handelt, ist die Annahme der Quasiperiodizität oft weit von der Realität entfernt.“* [Hes 05]

Die automatisierte Grundfrequenzerkennung in unsaubereren Signalen ist ein bis heute noch aktuelles Forschungsthema, das immer noch nicht zufriedenstellend gelöst wurde.

*„Bis heute ist, trotz der Vielfalt existierender Verfahren, keines bekannt, welches gleichzeitig die verschiedensten Anforderungen wie Sprecherunabhängigkeit (bei der Spracherkennung und -codierung), Robustheit gegenüber Hintergrundstörern, geringen Bedarf an komplizierten Rechenoperationen bei Echtzeitanwendungen und Genauigkeit der Grundfrequenzanalyse (GFA) erfüllen kann.“* [Hes 83].

Mögliche Anwendungsgebiete für Grundfrequenz-Algorithmen:

- Phonetik, Computerlinguistik: Grundfrequenzextraktion für Sprachdaten
- Finanzmathematik: Analyse von Aktienkursen
- Medizin: Pulsmessung, EKG
- Geophysik: Auswertung von Seismogrammen
- Codierungstheorie: Analyse von Bitströmen
- Allgemein: Visualisierung und Suche von periodischen, auch nicht-Sinus förmigen Zyklizitäten im Spektrum

Bei der Grundfrequenzerkennung handelt es sich um die Suche nach Periodizitäten auf linearen Zeitreihen.

Es gibt Algorithmen im Frequenzbereich und Zeitbereich und hybride Ansätze. Die erfolgreichsten Ansätze sind die Cepstral Methode und Autokorrelation.

*„Fast jeder Analysealgorithmus zur Ton-Analyse basiert auf einer Grundfrequenzbestimmung. Diese muss unter Umständen sehr exakt sein, weil von ihr die weiteren Analyseergebnisse abhängen.“ [Med 91]*

Im folgenden werden zwei neue Methoden zur Grundfrequenzanalyse vorgestellt.

### 5.3 Grundfrequenzsuche mit Hilfe der Fourier Transformation

Die verschiedenen Grundfrequenzen von zyklischen Wellenformen werden vom Ohr als „Tonhöhe“ wahrgenommen. Niedrige Grundfrequenzen entsprechen dabei „tiefen Tönen“ und hohe Grundfrequenzen „hohen Tönen“. Ein „Ton“ besteht jedoch nicht nur aus dem Grundton, sondern auch aus einer Vielzahl von harmonischen Obertönen. Die Obertöne sind eine Reihe von ganzzahlig vielfachen Frequenzen der Grundfrequenz ( $F_0$ ).

Die Obertöne wirken bei der Suche nach Zyklizitäten störend, weil es auf unsauberen Daten häufig vorkommt, dass deren Amplitude größer ist als die der Grundfrequenz. Die naive Suche nach einem Maximum im Spektrum ist nicht zuverlässig genug.

Daher liegt es nahe, die Daten nicht nur nach einer einzelnen Frequenz, sondern nach einer Gruppe von Frequenzen zu durchsuchen.

Im folgenden soll ein neuer Algorithmus auf FFT Basis zur Suche nach Zyklizitäten auf linearen Zeitreihen vorgestellt werden.

Die Grundfrequenzanalyse auf FFT Basis lässt sich in drei Teilschritte zerlegen:

- Erstellung eines Spektrogramms mit Hilfe der Fourier Transformation
- Logarithmieren im Frequenzbereich
- Durchsuchen des Spektrums nach kammartigen Strukturen durch Kreuzkorrelation

#### 5.3.1 Erstellung eines Spektrogramms mit Hilfe der Fourier Transformation

Die lineare Zeitreihe wird per FFT in ein 2 dimensionales Bild (Spektrogramm) transformiert. Da bei einer Abtastrate von 44 kHz nach Frequenzen bis 40 Hz gesucht werden soll, wird eine Fenstergröße von 2048 Samples verwendet. Es wird ein Kaiserfenster mit beta 5 und 50% overlap verwendet. Die X-Achse des Spektrogramms entspricht dem Zeitverlauf, die Y-Achse der Frequenz. Der Helligkeitswert bei  $(X,Y)$  entspricht dem Pegel der Frequenz Y zum Zeitpunkt X. Die Fouriertransformation liefert für jede Frequenz einen Real- und einen Imaginärteil. Der Pegel für jede Frequenz berechnet sich aus der euklidischen Distanz aus Real- und Imaginärteil. Die Phasenlage für die jeweilige Frequenz berechnet sich aus dem Arcustangens des Quotienten von Real und Imaginärteil.

**Bereits an dieser Stelle geht bei allen FFT basierten Suchmethoden Information verloren, da die Phaseninformation nicht in die weiteren Berechnungsschritte mit einfließt.**

### 5.3.2 Logarithmieren des Spektrogramms

Die Fourier Transformation löst tiefe Frequenzen sehr ungenau und hohe Frequenzen sehr detailliert auf. Durch das Logarithmieren des Spektrogramms im Frequenzbereich soll eine Gleichverteilung der Energie pro Oktave erreicht werden. Eine Oktave entspricht genau einer Verdoppelung der Frequenz. Darüber hinaus ist ein Logarithmieren für den später erfolgenden Arbeitsschritt nötig. Da in der europäischen Musik 12 Halbtöne pro Oktave verwendet werden, wird die Frequenz ausgehend von 65 Hz (C4) exponentiell mit dem Faktor 1.0594 erhöht ( $1.0594^{12}=2$ ). Die einzelnen Punkte des Quellbildes mit exponentiell wachsendem Y-Wert und linearer Interpolation abgetastet und so ein logarithmiertes Zielbild erzeugt. Um im hohen Frequenzbereich Aliasing zu vermeiden, wird 20x oversampling verwendet.

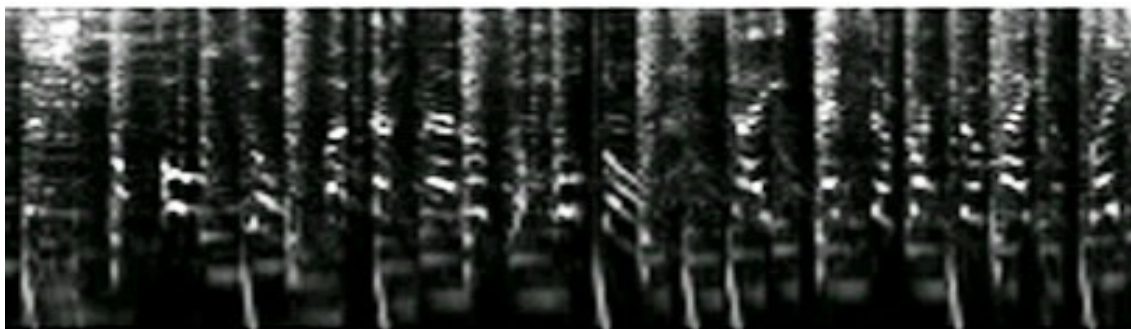


Abbildung 5.1: Logarithmiertes Spektrogramm („The Power of Love – Huey Lewis & The News“); X-Achse=Zeit (0-10 Sek), Y-Achse=Frequenz (40-5 kHz)

### 5.3.3 Durchsuchen des Spektrums nach kammartigen Strukturen

Nach dem Fourier Theorem lässt sich jede beliebige zyklische Wellenform als Summe von Sinuswellen zusammensetzen. Das Spektrum des Sägezahns lässt sich ausdrücken durch die Fourierreihe:

$$x(t) = c \cdot \sum_{k=1}^{\infty} \frac{\sin(2\pi k f t)}{k}$$

Dieses Wissen über die Struktur der Reihe kann helfen, die Suche zu verbessern. Die Amplituden der einzelnen Sinusschwingungen können in einen Suchalgorithmus mit einprogrammiert werden. Auf diese Weise ist es möglich, ein Spektrum gezielt nach bestimmten zyklischen Wellenformen wie Sägezahn oder Rechteck zu durchsuchen. Die Suche nach derartigen Strukturen kann in der Praxis durch eine zusätzliche Kreuzkorrelation des Spektrums der zu durchsuchenden Daten mit dem Spektrum der Wellenform realisiert werden.

„Die direkte Bestimmung der Grundfrequenz  $F_0$  aus dem ersten Maximum des Leistungsspektrums erweist sich als unzuverlässig. Vorzugsweise wird deswegen die harmonische Struktur des Signals bzw. des Spektrums untersucht. Dies erfolgt beispielsweise mit Hilfe der spektralen Kompression; hierbei wird die Grundfrequenz als der größte gemeinsame Teiler der Frequenzen aller Harmonischen berechnet. Das Leistungsspektrum wird entlang der Frequenzachse im Verhältnis 1:2, 1:3 usw. affin gepresst und anschließend auf das ursprüngliche Spektrum aufaddiert. Durch den kohärenten Beitrag aller höheren Harmonischen ergibt sich hierbei ein kräftiges Maximum bei der Frequenz  $F_0$ .“ [Sch 68]

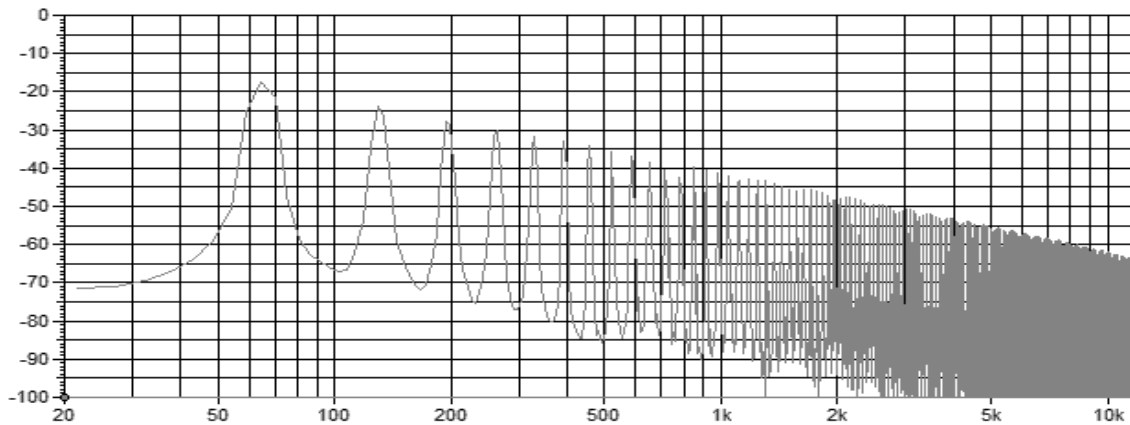


Abbildung 5.2: Sägezahnspektrum bei  $F_0=65\text{Hz}$ ; X-Achse=Frequenz (0-10kHz), Y-Achse=Pegel in dB

### 5.3.4 Die Wahl der Kreuzkorrelationsfunktion

„Im allgemeinen Fall erlaubt die Kreuzkorrelationsfunktion KKF Aussagen über die Ähnlichkeit eines Signals  $g(n)$  zu einem um die Zeit  $t$  verschobenen Signal  $v(n+t)$ . Sie ist definiert als:

$$AKF(k) = \sum_{n=-\infty}^{+\infty} g(n) \cdot g(n+k),$$

wobei die Zahlenfolgen  $g(n)$  und  $v(n)$  diskrete (digitalisierte) Signale im Zeitbereich sind.“

[Res 99]

Abbildung 5.3 zeigt eine mögliche Kreuzkorrelationsfunktion, um ein Spektrum gezielt nach Sägezahnswingungen zu durchsuchen:

$$f(x) = 1 / (x * x / 200 + 1) * \cos( (1,05946^x - 1) * 2 * \text{Pi} )$$

Das Maximum bei  $x=0$  entspricht der Grundfrequenz. Das zweite Maximum entspricht der ersten Oberschwingung. Würde man das Spektrum nach einem Rechteck absuchen, müsste das zweite Maximum erst bei der zweiten Oberschwingung definiert werden, da das Rechteck nur aus ganzzahlig ungeradzahlig Sinuswellen besteht.

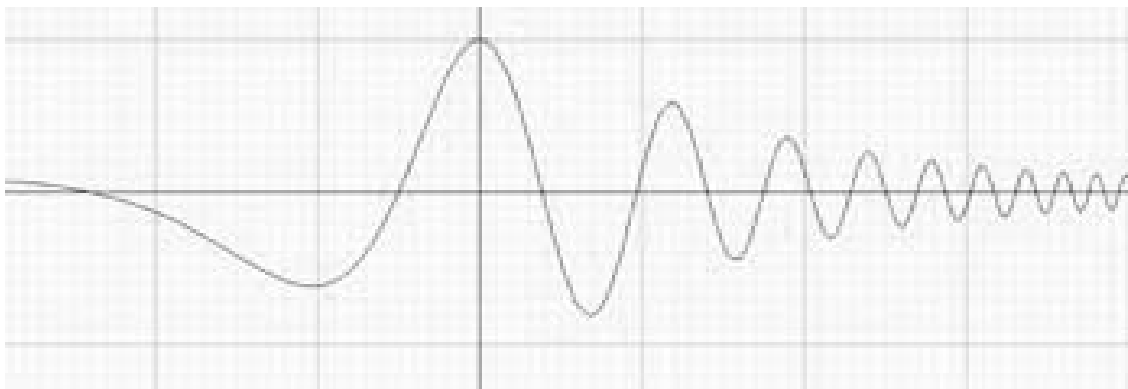


Abbildung 5.3: Kreuzkorrelationsfunktion für sägezahnartige Wellenformen; X-Achse: Frequenz; Y-Achse: Amplitude;

### 5.3.5 Faltung

Das logarithmierte Spektrogramm von Abbildung 5.1 wird in Y-Richtung mit der Kreuzkorrelationsfunktion gefaltet.

Der Algorithmus zur Faltung des Spektrogramms mit der Kreuzkorrelationsfunktion lautet (vereinfacht):

```
for (int x=0;x<bildBreite;x++)
{
    for (int y=0;y<bildHoehe;y++)
    {
        float sum=0;
        for (int i=0;i<bildHoehe;i++)
        {
            sum += Spektrogramm[x][i] * Kreuzkorrelationsfunktion[y-i+c];
        }
        GefaltetesBild[x][y] = sum;
    }
}
```

Die lokalen Maxima im gefalteten Spektrogramm sollten nun die Grundfrequenzen der im Bild vorkommenden Sägezahnspektren zeigen...

### 5.3.6 Komplexität

Die Komplexitätsklasse zum Erzeugen des Spektrogramms auf Basis der FFT ist  $o(x \cdot n \cdot \log n)$ , wobei  $x$  der Zahl der durchzuführenden Transformationen ist und  $n$  der Fenstergröße entspricht.

Die Komplexitätsklasse für das Logarithmieren des Spektrogramms ist  $o(x \cdot y \cdot v)$ , wobei  $x$  der Bildbreite,  $y$  der Bildhöhe und  $v$  dem Oversamplingfaktor entspricht.

Die Komplexitätsklasse für den Korrelationsalgorithmus ist entsprechend den drei verschachtelten for-Schleifen  $o(x \cdot y \cdot i)$ , wobei  $x$  der Bildbreite,  $y$  der Bildhöhe und  $i$  der Länge der Kreuzkorrelationsfunktion entspricht.

### 5.3.7 Schwächen der Methodik

Das folgende Bild zeigt ein mit der Kreuzkorrelationsfunktion gefaltetes logarithmiertes Spektrogramm. Die Faltung mit der Kreuzkorrelationsfunktion hat nicht, wie erwartet, zu einer Vereinfachung der Entropie des Bildes geführt. Es sind gegenüber dem Spektrogramm zusätzliche horizontale Linien erkennbar.

Als Quellmaterial für das Bild wurde bewusst ein spektral komplexes Musikstück gewählt. Der gezeigte Ausschnitt enthält zeitgleich durch Formanten geprägte Vokale, stimmlose Konsonanten, Rhythmusinstrumente, Naturinstrumente und synthetische Klänge.

Als Kreuzkorrelationsfunktion wurde das Spektrum einer Sägezahnwelle gewählt. In der Praxis enthalten jedoch lediglich die Vokale und die synthetischen Klänge im Musikstück zum Sägezahn ähnliche Spektren.

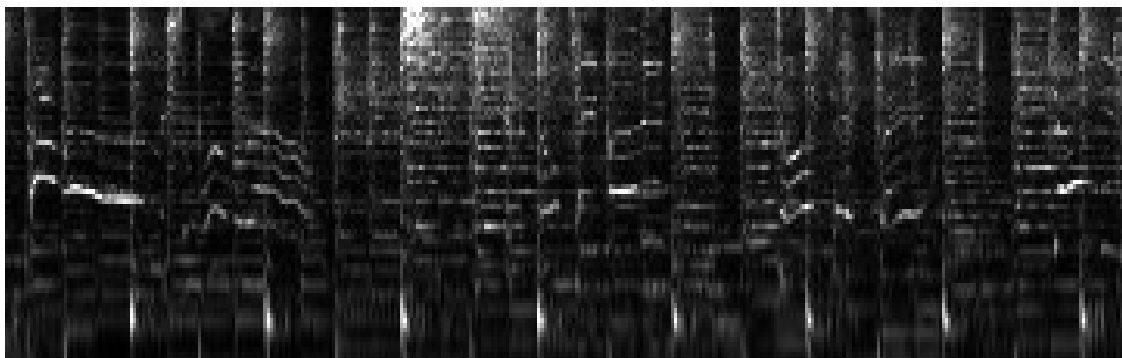


Abbildung 5.4: Mit Kreuzkorrelationsfunktion gefaltetes Spektrogramm („The Power of Love – Huey Lewis & The News“); X-Achse: Zeit (45-55 Sek), Y-Achse: Frequenz (40-5 kHz)

Die Anwendung der Kreuzkorrelation auf dem logarithmierten FFT Spektrum hat nicht zu einer Verringerung der Entropie im Bild geführt.

**Die fourierbasierte Grundfrequenzsuche mit Kreuzkorrelationsfunktion liefert nur dann sinnvolle Ergebnisse, wenn der spektrale Inhalt der zu suchenden Wellenform exakt bekannt ist.**

**Eine für den spektralen Inhalt ungeeignet gewählte Kreuzkorrelationsfunktion verschlechtert das Suchergebnis.**

Bei Sprache ist durch die individuelle Sprecherphysiognomie und den Formveränderungen im Vokaltrakt die Frequenzantwort nicht statisch. Eine genaue Annahme über die spektrale Zusammensetzung des Signals kann somit nicht gemacht werden.

*„Die Unterschiede in der Periodizität rühren von verschiedenen Faktoren der Sprachbildung. Zum einen ist die Wellenform die durch die Stimmbänder erzeugt wird, keine reine Impulsfolge. Da neben der Periodizität auch die genaue Struktur des Signals variiert, ist die Erkennung relevanter Amplituden-Maxima mit Schwierigkeiten verbunden. Zum anderen wird durch den Vokaltrakt das ursprüngliche Stimmbandsignal derart verformt, dass eine eindeutige Darstellung der Grundfrequenz problematisch wird.“ (Jürgen Bock, Algorithmen zur Tonhöhenenerkennung und Vergleich verschiedener Implementierungen, 2004)*

Für Query-by-Humming Anwendungen ist daher die in diesem Kapitel beschriebene Methodik nicht gut geeignet. Es wird ein Algorithmus benötigt, der eine lineare Zeitreihe nach Zyklizitäten durchsucht, dabei jedoch keine Annahme über deren spektrale Zusammensetzung machen muss.

## 5.4 Grundfrequenzsuche mit dem AKF+COMB Algorithmus

Im folgenden wird die AKF+COMB Methode vorgestellt („verbesserte Autokorrelation mit Kamm“).

AKF+COMB ist ein leistungsfähiger Algorithmus, der in Grundzügen auf der Autokorrelation basiert und Vorteile gegenüber den existierenden Verfahren aufweist.

Im Gegensatz zum klassischen, Fourier-basierten Spektrogramm wird beim AKF+COMB Algorithmus eine lineare Zeitreihe nicht nach Sinusschwingungen, sondern nach beliebigen zyklischen Wellenformen durchsucht.

Der AKF+COMB Algorithmus lässt sich in zwei Teilschritte zerlegen:

- 1) Einer verbesserten Autokorrelation, die die Daten in Abhängigkeit von der Verschiebung auf Selbstähnlichkeit untersucht (AKF+)
- 2) Einer Kreuzkorrelation mit einer Kammfunktion, die n-fach Maximas eliminiert (COMB)

Im Folgenden wird die Funktionsweise der AKF+COMB schrittweise veranschaulicht.

### 5.4.1 Spektrale Eigenschaften des zu suchenden Signals

Der Sägezahn enthält eine harmonische Struktur, die der menschlichen Stimme ähnlich ist.

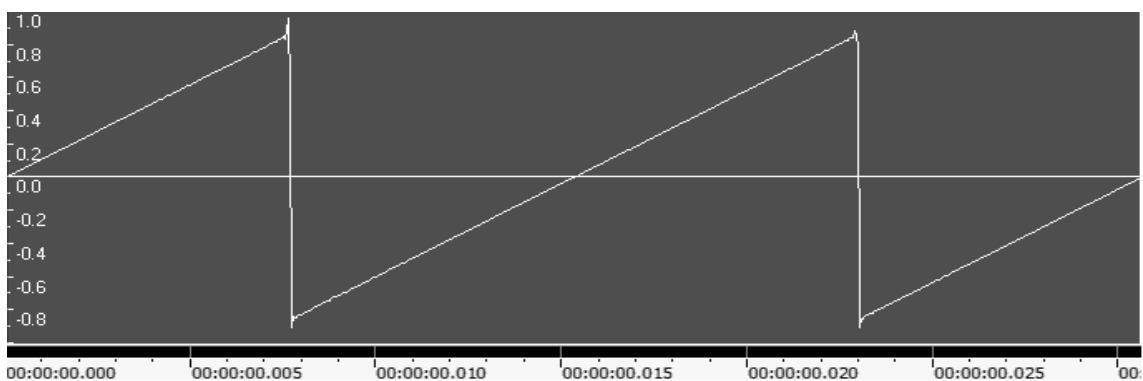


Abbildung 5.5: Sägezahnschwingung bei 65Hz, Abtastrate 44kHz; 2 Zyklen

Das vorher gezeigte Bild zeigt eine Sägezahn­schwin­gung im Zeitbereich. Abbildung 5.6 visualisiert das zugehörige Spektrum im Frequenzbereich. Abbildung 5.7 zeigt das Spektrum der selben Funktion bei der doppelten Grundfrequenz.

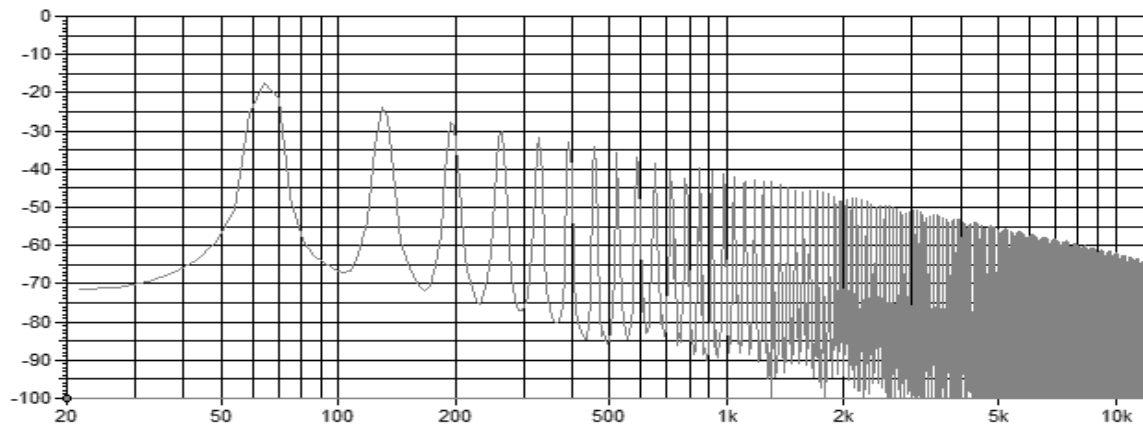


Abbildung 5.6: Sägezahnspektrum bei  $F_0$ : 65Hz; X-Achse: Frequenz (0-10kHz), Y-Achse: Pegel in dB

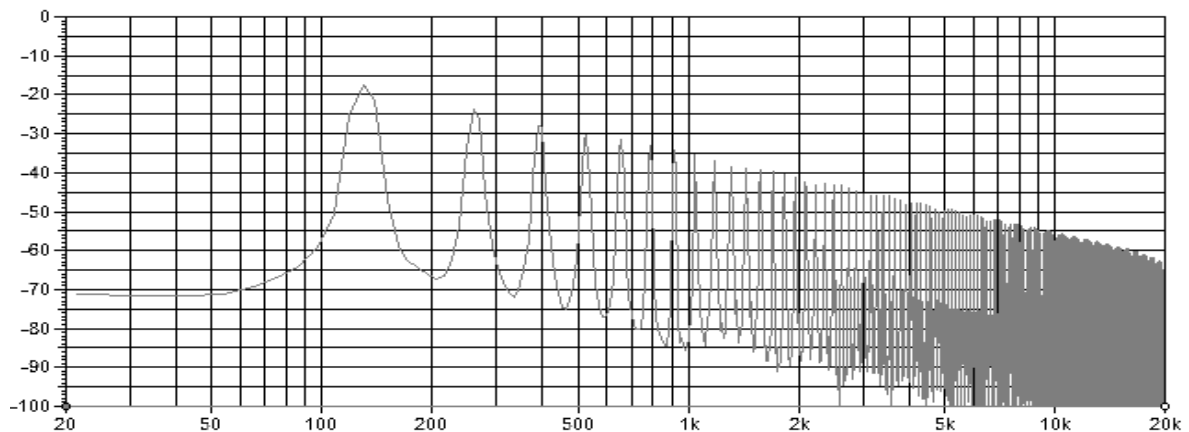


Abbildung 5.7: Sägezahnspektrum bei  $F_0$ : 130Hz; X-Achse: Frequenz (0-20kHz), Y-Achse: Pegel in dB

Es fällt auf, dass die beiden Spektren einander sehr ähnlich sind. Sie haben gemeinsame Maxima bei allen geradzahli­gen, ganzzahlig  $n$ -fachen von 65 Hz ( $65 \cdot 2$  Hz,  $65 \cdot 4$  Hz,...). Die ungeradzahli­gen, ganzzahlig  $n$ -fachen ( $65$  Hz,  $65 \cdot 3$  Hz,  $65 \cdot 5$  Hz,...) fehlen im zweiten Bild.

Bei der Suche nach der Grundfrequenz kann die hohe spektrale Ähnlichkeit zu Frequenzverdopplungs- und Frequenzhalbierungsfehlern führen.

„Der Grundfrequenzbereich der menschlichen Stimme liegt zwischen 50 - 300 Hz und ist vom Sprecher abhängig. Für das zugehörige Spektrum wird festgestellt, daß es zu den hohen Frequenzen hin um ca. 6 dB pro Oktave abnimmt... Da die Grundfrequenz der menschlichen Stimme im niederfrequenten Bereich des Spektrums liegt, sind hohe Frequenzen bei der Analyse des Sprachsignals nicht erwünscht. In den hohen Frequenzen sind jedoch harmonische Anteile der Grundfrequenz enthalten. Diese harmonischen Anteile der Grundfrequenz finden sich in den Maxima wieder..." [Boc 04]

Im Gegensatz zum exakt definierten und sauberen Sägezahnspektrum hat man es bei der menschlichen Stimme jedoch mit einem sehr unsauberen Signal zu tun. Der Frequenzgang weist deutliche Ausdellungen bei den Formantfrequenzen auf. Abbildung 5.8 zeigt das Spektrum eines „Problemsignals“. Dort ist die Frequenz des ersten Formanten (210 Hz) der Grundfrequenz (103 Hz) sehr nahe. Ein naiv implementierter Suchalgorithmus, der nur nach dem spektralen Energiemaximum sucht, würde hier 206 Hz erkennen.

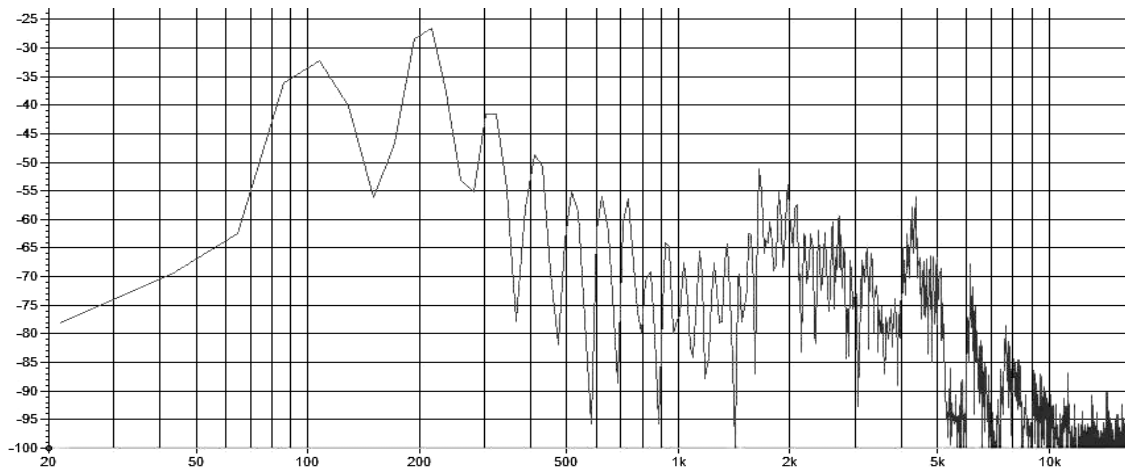


Abbildung 5.8: Spektrum eines männlichen Summers bei  $F_0=103\text{Hz}$ ; X-Achse: Frequenz (0-20kHz), Y-Achse: Pegel in dB

Durch das Antiformant-Filter, das bereits im Vorverarbeitungsschritt vorgestellt wurde, können die Ausdellungen im Frequenzgang grob ausgeglichen werden. Frequenzverdopplungs- und Frequenzhalbierungsfehler können aber aufgrund von Stimmindividualitäten jedoch immer noch auftreten. Um die Frequenzsuche robuster zu machen, ist daher ein Nachverarbeitungsschritt nötig (COMB).

## 5.4.2 Die Wahl eines Testsignals

Als Testsignal für die Bilder im folgenden Beispiel wurden von einem Signalgenerator nacheinander für eine Dauer von jeweils 0.2 Sekunden die Frequenzen 65 Hz, 73 Hz, 82 Hz, 87 Hz, ..., 260 Hz erzeugt. Die Frequenzfolge entspricht der Tonleiter C4, D4, E4, F4, G4, A4, B4, C5, ... C6. Als Trägerwellenform diente ein Sägezahn.

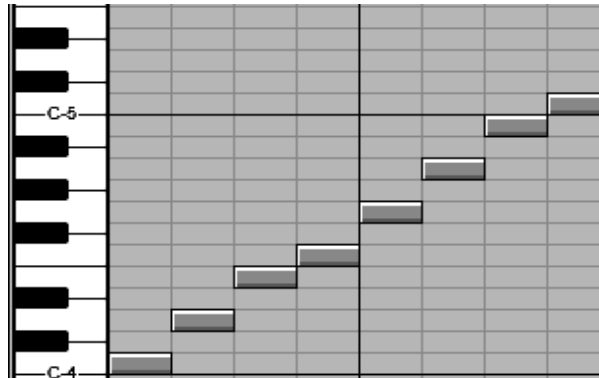


Abbildung 5.9: Tonleiter in der Sequencersoftware; X-Achse: Zeit (0-2,6 Sek), Y-Achse: Tonhöhe

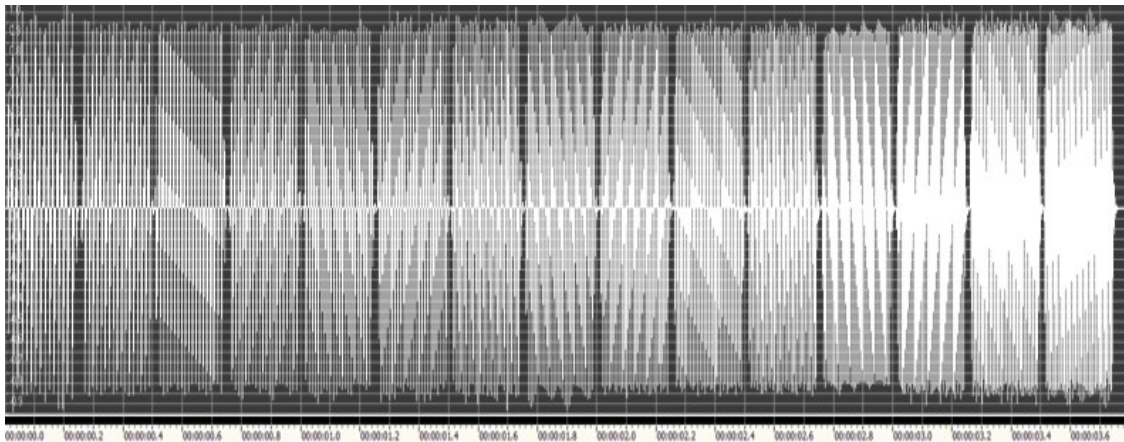


Abbildung 5.10: Tonleiter als Zeitreihe; X-Achse: Zeit (0-2,6 Sek), Y-Achse: Pegel

Zur Analyse und Weiterverarbeitung können die später extrahierten Grundfrequenzen in „Hummel“ in eine WAV Datei resynthetisiert werden.

### 5.4.3 Der AKF+ Algorithmus - die verbesserte Autokorrelation

Def.: Autokorrelation

*„Die Autokorrelation ist ein Begriff aus der Statistik und der Signalverarbeitung. Im statistischen Modell geht man von einer geordneten Folge von Zufallsvariablen aus. Vergleicht man die Folge mit sich selbst, so spricht man von Autokorrelation. Da jede unverschobene Folge mit sich selbst am ähnlichsten ist, hat die Autokorrelation für die unverschobenen Folgen den höchsten Wert. Wenn zwischen den Gliedern der Folge eine Beziehung besteht, die mehr als zufällig ist, hat auch die Korrelation der ursprünglichen Folge mit der verschobenen Folge in der Regel einen Wert, der signifikant von Null abweicht. ... In der Signalverarbeitung spricht man von Autokorrelation, wenn die kontinuierliche oder zeitdiskrete Funktion (z. B. ein- oder mehrdimensionale Funktion über die Zeit oder den Ort) mit sich selbst korreliert wird. Beispielsweise  $x(t)$  mit  $x(t+\text{Verschiebung})$ . ...“ [WIK 08-2]*

### 5.4.4 Vorverarbeitung

Das Ergebnis der Autokorrelation kann durch eine Hochpassfilterung der Zeitreihe verbessert werden. In der Theorie sollte dabei die cutoff Frequenz eines unendlich steilflankigen Filters dabei genau der Wellenlänge der längsten vorkommenden Korrelationsverschiebung entsprechen.

In der Praxis eignen sich dazu gut phasenneutrale FIR Filter. Da sich ein unendlich steilflankiges Filter nicht realisieren lässt, wird die cutoff etwas tiefer angesetzt.

**Durch die Hochpassfilterung werden irrelevante und die Messung verschlechternde Informationen wie niederfrequente Störsignale oder eine Verschiebung der Funktion entlang der Y-Achse ausgeblendet.**

*„Die Autokorrelationsfunktion wiederum ist bei der GFB gegen Rauschen sehr unempfindlich, gegenüber Formanten jedoch recht empfindlich.“ [Hes 05]*

Durch das im vorherigen Kapitel vorgestellte Anti-Formant Filter für gesummte Signale wird dieser Störeinfluss stark abgeschwächt.

## 5.4.5 Autokorrelation

Bei Abbildung 5.11 wird eine 'klassische' Autokorrelation mit der Sägezahnntonleiter als Testsignal durchgeführt. Die Verschiebung wird logarithmisch skaliert, so dass genau 12 Zeilen einer Frequenzverdoppelung oder Wellenlängenhalbierung entsprechen. Die logarithmische Skala ist hier nicht nur der Performance dienlich, sondern auch für einen später erfolgenden Rechenschritt nötig. Die unterste Zeile ist die längste gemessene Verschiebung mit 1500 Abtastwerten, die oberste Zeile entspricht der kürzesten mit 30 Abtastwerten. Anhand der Zeilenhöhe kann die Grundfrequenz des entsprechenden Halbtons abgelesen werden. Die 'Helligkeiten' der Korrelationswerte werden im folgenden immer mit der Fenstergröße normiert.

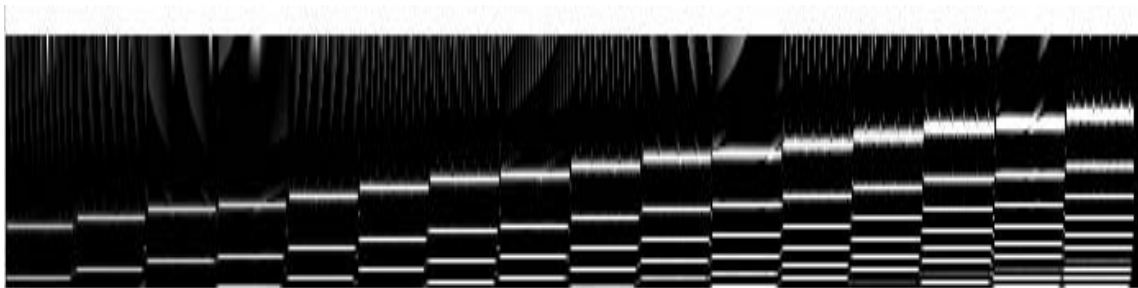


Abbildung 5.11: Autokorrelierte Tonleiter; X-Achse: Zeit (0-2,6 Sek); Y-Achse: Verschiebung (1500-30 Samples bei 44kHz); 1 Zyklus Fenstergröße; Rechteck Fenster

Das oben gezeigte Bild ist einem Spektrogramm ähnlich. Die Helligkeiten der Pixel im Bild entsprechen den Wahrscheinlichkeiten für das Auftreten der logarithmisch skalierten Grundfrequenz  $Y$  in Abhängigkeit von der Zeit  $X$ . Die oberste der horizontalen Linien entspricht der realen zu suchenden Grundfrequenz. Dieses Bild der 'klassischen' Autokorrelation weist jedoch drei Arten von Artefakten auf:

- Bei sehr kurzen Verschiebungen und langen 'realen' Wellenlängen ist die Zeitreihe sich selbst häufig ähnlich. Dies lässt sich auf dem Bild an den Zackenmustern am oberen Rand erkennen.
- Bei Verschiebungsdistanzen, die dem ganzzahlig vielfachen der Grundfrequenz entsprechen, treten auch Korrelationsmaxima auf. Diese sind als zusätzliche horizontale Linien unterhalb der Grundfrequenz zu sehen.
- In Abhängigkeit von der Wellenlänge erscheint das Bild verschoben. Die Startzeitpunkte der horizontal aufeinander 'gestapelten' Linien sind nicht identisch.

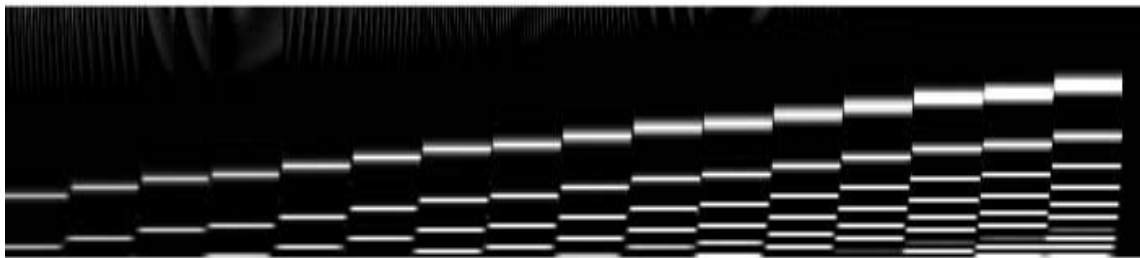
## 5.4.6 Autokorrelation mit Fensterung

Um das Bild unten zu erzeugen, werden die Abtastwerte bei der Faltung mit einem Kaiserfenster ( $\beta=5$ ) gewichtet. Werte im Mittelpunkt werden dadurch stärker gewichtet als Werte nahe am Randbereich. Eine Fenstergröße von nur einem Zyklus ist bei Hinzunahme einer Fensterfunktion nicht mehr sinnvoll, da zu starke Seitenbandeffekte ähnlich einer Amplitudenmodulation auftreten. Es werden für die folgenden Bilder immer  $n$ -fach ganzzahlig Vielfache der zu untersuchenden Wellenlänge als Fenstergröße gewählt.

*„Zur Periodenbestimmung sind mindestens zwei aufeinanderfolgende Perioden des Signals nötig“. [Med 91]*

**Wie bei der Verallgemeinerung der Heisenbergschen Unschärferelation gilt hier:**

**Je größer die Zahl der Zyklen gewählt wird, desto genauer kann die Frequenz aufgelöst werden. Dabei sinkt jedoch die Schärfe der Zeitauflösung. Für die Grundfrequenzanalyse bei Sprachdaten liefern 6 Zyklen mit Kaiser  $\beta=3$  den besten Kompromiss aus Zeitauflösung, Performance und Frequenzauflösung.**



*Abbildung 5.12: Autokorrelierte Tonleiter; X-Achse: Zeit (0-2,6 Sek); Y-Achse: Verschiebung (1500-30 Samples bei 44kHz); 4 Zyklen mit Kaiser-Fenster; ohne Offsetkorrektur*

## 5.4.7 Autokorrelation mit Fensterung und Offsetkorrektur (AKF+)

Im obigen Bild sind die Zackenmuster am oberen Rand durch die mit dem Kaiserfenster gewichtete Korrelation über vier Grundfrequenzzyklen verschwunden. Das Selbstähnlichkeitsproblem bei sehr kurzen Verschiebungen wurde dadurch gelöst.

Es treten aber immer noch Mehrfachmaximas und 'windschief' gestapelte Linien auf. Um die Linien wieder rechtwinklig anzuordnen, müssen die Startzeitpunkte in Abhängigkeit von der Suchfrequenz korrigiert werden. Die Offsets werden genau um die halbe Fenstergröße nach vorne verschoben. Die beiden Fenster, die korreliert werden, liegen dadurch zeitlich symmetrisch vor und hinter dem Messpunkt.

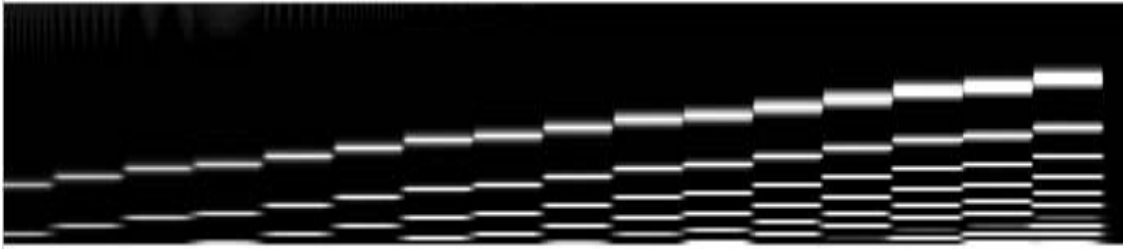


Abbildung 5.13: Autokorrelierte Tonleiter; X-Achse: Zeit (0-2,6 Sek); Y-Achse: Verschiebung (1500-30 Samples bei 44kHz); 8 Zyklen mit Kaiser-Fenster; mit Offsetkorrektur

Der Algorithmus für die Erzeugung eines AKF+ Spektrogramms lautet (vereinfacht):

```

fr = startfrequenz;
expfact = 1.05946;
//Bild zeilenweise aufbauen
for (int y=0;y<bildHöhe;y++)
{
    //Korrelation über 6 Zyklen
    int numZykl = 6;
    //Exponentiell wachsende Analysefrequenz
    fr *= expfact;
    //Korrelationsverzögerung in Samples
    long del = (long)(samplerate / fr);
    //Gesamtzahl der zu korrelierende Samples
    long nums = (long)(samplerate / fr)*numZykl;
    //Offsetausgleich für Symmetrie
    ofs = (long)((samplerate / frStart)*numZykl/2-nums/2);
    //Erzeuge Kaiserfenster der Länge nums mit beta=3
    kaiserWin(winfunc,nums,3);
    for (long x=0;x<bildBreite;x++)
    {
        float sum = 0;
        //Mit Fensterfunktion gewichtete Autokorrelation über mehrere Zyklen
        for (int i=0;i<nums;i++)
        {
            float a = sample[ofs+del+i];
            float b = sample[ofs+i];
            sum += (a*b)*winfunc[i];
        }
        //Normieren
        sum /= (bildHöhe);
        sum /= nums;
        //Erzeuge Bild
        im[x][y] = sum;
        //50% overlap
        ofs += winsiz/2;
    }
}

```

## 5.4.8 Komplexität und Optimierung der AKF+

Die Komplexitätsklasse der AKF+ ist  $o(n^3)$ . Der Rechenaufwand für lange Zyklichkeiten ist größer als der für kurze, da immer über eine feste Anzahl von Zyklen korreliert wird. Die Zahl der zu korrelierenden Samples verdoppelt sich alle  $n$  Frequenzanalyseschritte, weil eine logarithmische Skalierung im Frequenzbereich gewählt wurde. Daher bietet sich die Anwendung einer Multiskalenstrategie an. Es wird alle  $n$  Schritte die Abtastrate halbiert, so dass kleine Verschiebungen mit hoher und große Verschiebungen mit niedriger Abtastrate berechnet werden. Zur Vermeidung von Aliasing, muss vorher ein resampling mit FIR Halbbandfiltern auf dem Analysesignal durchgeführt werden. Die Komplexitätsklasse lässt sich so auf  $o(n^2 \cdot \log(n))$  verringern.

## 5.4.9 Autokorrelation mit Fensterung, Offsetkorrektur und Kammfilterung (AKF+COMB)

Wie in Abbildung 5.13 zu sehen ist, hat die negative Offsetkorrektur das Verschiebungsproblem der Autokorrelationsfunktion gelöst.

Es sind immer noch mehrfach-Maxima erkennbar. Sie treten bei spektral statischen Messdaten immer bei der Verschiebung um das genau ganzzahlig  $n$ -fache ( $2x$ ,  $3x$ ,  $4x$ , ...,  $Nx$ ) der Wellenlänge der zu suchenden Grundfrequenz auf. In Abbildung 5.13 sind dies alle Linien, die sich unterhalb der obersten befinden. Da für die Autokorrelation eine logarithmische Frequenzaufteilung gewählt wurde, sind die Abstände der Linien zueinander und deren charakteristische Muster von der Form her immer gleich.

*„Es ist daher sinnvoll, auch Zeitbedingungen abzufragen und beispielsweise eine Stützstelle  $q$  auch nach ihrem Verhalten gegenüber der Nachbarschaft zu überprüfen.“ [Hes 05]*

Das unten gezeigte Bild veranschaulicht das  $n$ -fach Maxima Problem im Zeitbereich. Die gelbe Wellenform ist nicht nur der blauen mit der Distanz  $w$  ähnlich, sondern auch der grünen mit der Distanz  $2 \cdot w$  und der weißen mit  $3 \cdot w$ .

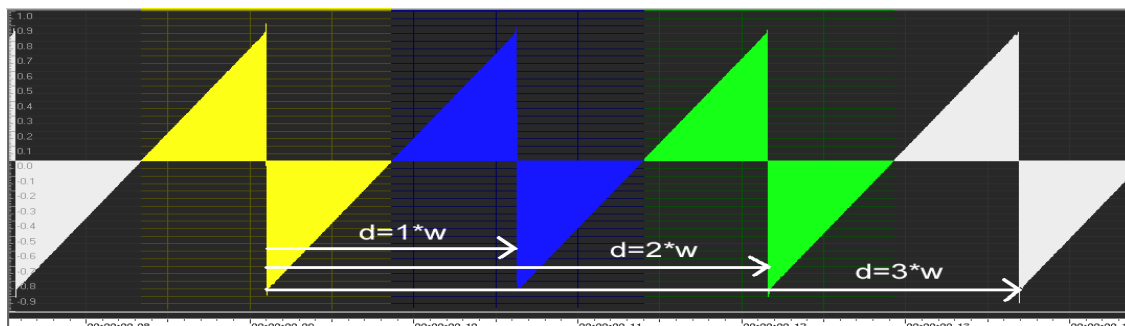


Abbildung 5.14:  $n$ -fach Maxima Problem der Autokorrelation im Zeitbereich; X-Achse: Zeit; Y-Achse: Amplitude

Die oberste der horizontalen Linien in Abbildung 5.15 entspricht der realen zu suchenden Grundfrequenz. Da die zu untersuchenden Zeitreihen jedoch in der Praxis eine Vielzahl an Störsignalen enthalten, reicht es hier nicht aus, das oberste Maximum auszuwählen. Der Algorithmus soll robust sein. Daher liegt es nahe, das Bild nach einer kammartigen Struktur zu durchsuchen, die auch die Information über die Mehrfachmaximas berücksichtigt. Es wird ein charakteristisches Muster in einem Bild gesucht.

Der rote Streifen im folgenden Bild ist das n-fach Maxima Muster, nach dem gesucht wird. Da eine logarithmische Frequenzaufteilung für die Autokorrelation gewählt wurde, bleibt die charakteristische Form unabhängig von der Grundfrequenz nahezu erhalten. Das Muster wird im folgenden näherungsweise als verschiebungsinvariant zur Y-Achse angenommen.



Abbildung 5.15: Verschiebungsinvariantes n-fach Maxima Muster

Die Helligkeiten der Pixel im Bild oben entsprechen den Wahrscheinlichkeiten für das Auftreten der logarithmisch skalierten Grundfrequenz Y in Abhängigkeit von der Zeit X. Stellt man die „Helligkeitsverteilung“ der roten Linie als Funktionswert einer Funktion dar, enthält man folgendes Bild:

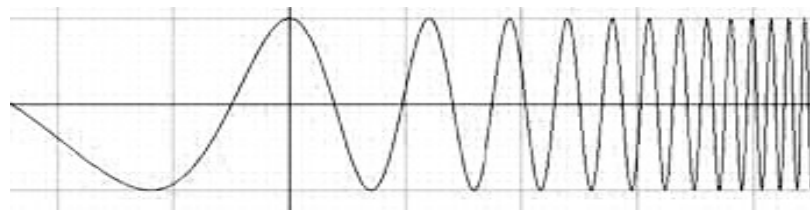
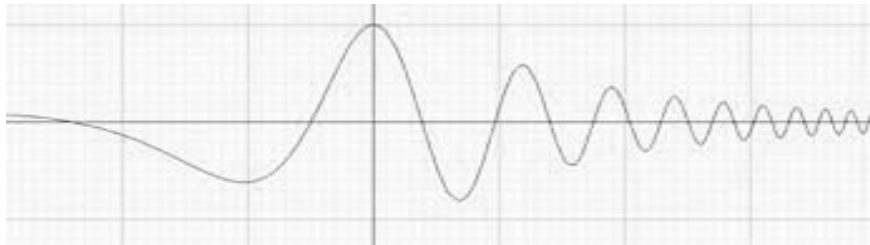


Abbildung 5.16: n-fach Maxima Muster für  $f(x) = \cos((1,05946^x - 1) * 2 * \pi)$ ; X-Achse: Verschiebung; Y-Achse: Amplitude

Das Maximum bei  $x=0$  entspricht der Grundfrequenz und das zweite Maximum bei  $x=12$  der genau doppelten Wellenlänge (Oktave). Der Faktor 1,05946 in der Formel entspricht der zwölften Wurzel aus zwei. Die Zahl zwölf ergibt sich für unsere Anwendung aus der logarithmischen Einteilung nach Halbtönen in der Musik. Je nach Bedarf lassen sich hier auch beliebig andere Skalierungen wählen.

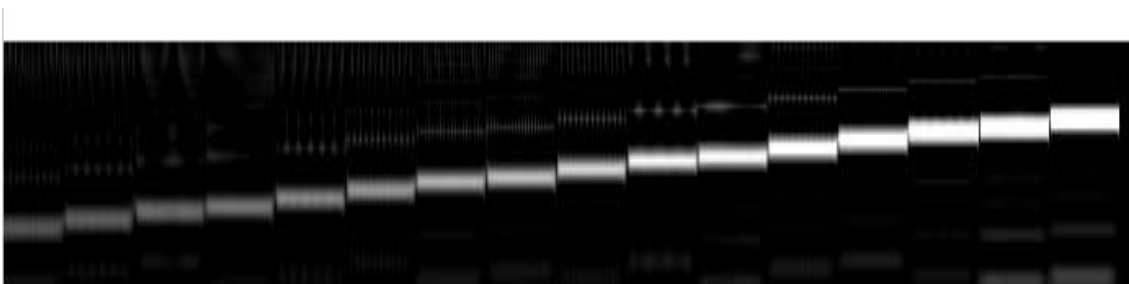
Bei der Funktion fällt auf, dass sie auch negative Werte annimmt. Dadurch werden auch negative Werte der Autokorrelationsfunktion als zusätzliche Informationsquelle mit einbezogen.

Da nur ein beschränkt großer Messbereich vorliegt, sollen Informationen im Randbereich der  $n$ -fach Maxima Funktion weniger stark gewichtet werden, als Informationen im Zentrum. Dazu wird die Funktion mit einer Fensterfunktion multipliziert.



*Abbildung 5.17: Gefensterter  $n$ -fach Maxima Muster; X-Achse: Verschiebung; Y-Achse: Amplitude*

Abbildung 5.15 soll nun nach Strukturen, die der roten Linie entsprechen, durchsucht werden. Dazu wird die Kreuzkorrelation verwendet. Das Bild wird pixelweise in Y-Richtung mit der Impulsantwort der Funktion des  $n$ -fach Maxima Musters gefaltet.



*Abbildung 5.16: AKF+COMB; X-Achse: Zeit (0-2,6 Sek); Y-Achse: Verschiebung (1500-30 Samples bei 44kHz); 4 Zyklen mit Kaiser-Fenster;*

Der Algorithmus für die COMB Kammfilterung der AKF+ Spektrogramms lautet (vereinfacht):

```
for (int x=0;x<bildBreite;x++)
{
    for (long y=0;y<bildHoehe;y++)
    {
        float sum=0;
        for (int i=0;i<bildHoehe;i++)
            sum += ungefaltetesBild[x][i]*crosscorrelfunc[y-i+c];
        gefaltetesBild[x][y] = sum;
    }
}
```

#### 5.4.10 Vergleich zwischen AKF und AKF+COMB Spektrogrammen

*„Die Kurzzeittransformation soll alle Informationen zur Periodizität eines Signals in einem Messintervall auf einen einzigen Spitzenwert abbilden.“ [Hes 02]*

In den unten gezeigten Bildern zum Vergleich von AKF und AKF+COMB ist deutlich eine Verringerung der Entropie zu erkennen. Im hohen Frequenzbereich tendiert die AKF zu vertikalen Linienmustern. Im tiefen Frequenzbereich sind horizontale Kammstrukturen erkennbar. Beide Artefakte treten bei der AKF+COMB nicht mehr auf.



Abbildung 5.17: AKF; 10 Sekunden Ausschnitt aus „Bomfunk MC's: i know“

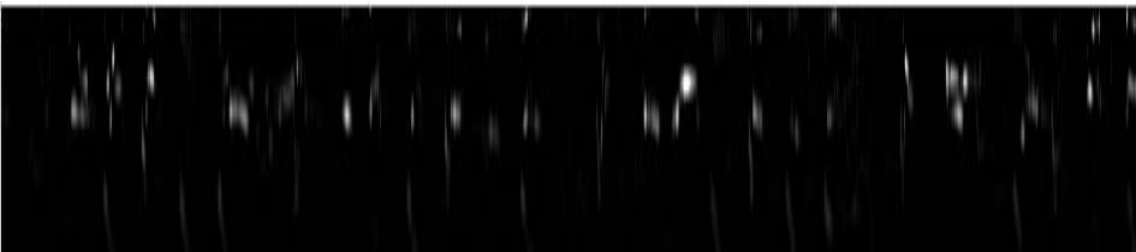


Abbildung 5.18: AKF+COMB; 10 Sekunden Ausschnitt aus „Bomfunk MC's: i know“



Abbildung 5.19: AKF; 10 Sekunden Ausschnitt aus „Huey Lweis & The News: The power of love“



Abbildung 5.20: AKF+COMB; 10 Sekunden Ausschnitt aus „Huey Lweis & The News: The power of love“

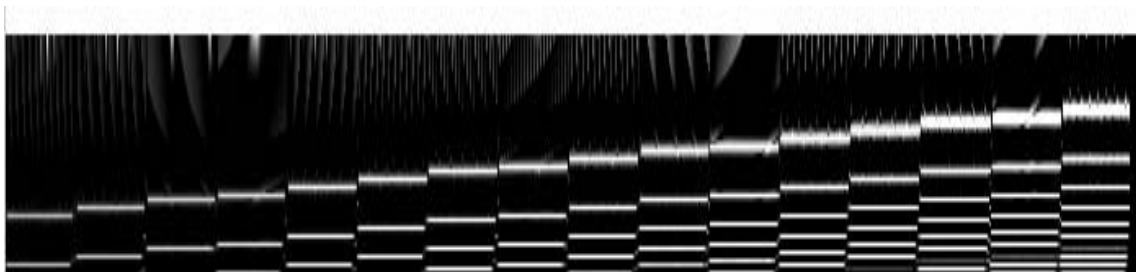


Abbildung 5.21: AKF; Sägezahn mit 65 Hz, 73 Hz, 82 Hz, 87 Hz, ..., 260 Hz

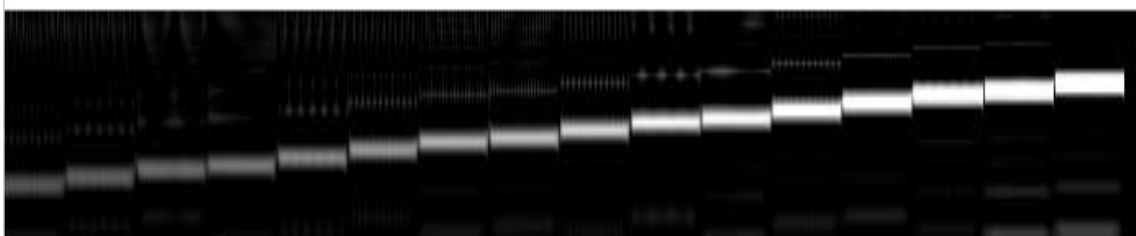


Abbildung 5.22: AKF+COMB; Sägezahn mit 65 Hz, 73 Hz, 82 Hz, 87 Hz, ..., 260 Hz

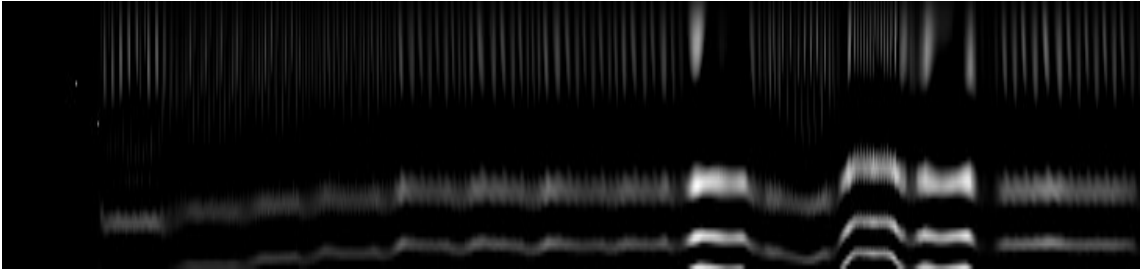


Abbildung 5.23: AKF; „Fuchs du hast die Gans gestohlen“, gesummt mit Antiformantfilter



Abbildung 5.24: AKF+COMB; „Fuchs du hast die Gans gestohlen“, gesummt mit Antiformantfilter

Der AKF+COMB Algorithmus kann zum Erzeugen von „Grundfrequenz Spektrogrammen“ verwendet werden. Im Gegensatz zum klassischen, Fourier-basierten Spektrogramm wird beim AKF+COMB Algorithmus eine lineare Zeitreihe nicht nach Sinusschwingungen, sondern nach beliebig geformten Zyklizitäten durchsucht.

Die Helligkeit  $h=f(x,y)$  im „Grundfrequenz Spektrogramm“ entspricht der Wahrscheinlichkeit  $p$  für das Auftreten einer Zyklizität mit der Grundfrequenz  $y$  zum Zeitpunkt  $x$ .

### 5.4.11 Komplexität und Echtzeitfähigkeit

Die Komplexität lässt sich durch die Anwendung der FFT bei der Faltung mit der Impulsantwort von  $O(n^3)$  auf  $O(n^2 \cdot \log(n))$  verringern.

Der AKF+COMB Algorithmus ist echtzeitfähig, da sich das Spektrogramm auch spaltenweise aufbauen lässt. Die Verarbeitung kann auch auf einem Teilausschnitt der linearen Zeitreihe erfolgen.

Die Verarbeitungslatenz  $L$  in Samples entspricht genau:

$$L = \text{Abtastrate} / F_t * N_Z$$

$F_t$  entspricht dabei der tiefsten zu Analysierenden Grundfrequenz.  $N_Z$  entspricht der Zahl der Zyklen, über die korreliert wird.

In der Praxis liegt die Latenz bei der Analyse von Audiodaten bei 60 ms. Die Analyse von Audiodaten lässt sich auf einem modernen Prozessor mit 3 GFlops in Echtzeit durchführen.

In "Hummel" dauert die Analyse eines 10 Sekunden langen Ausschnittes auf einem 2,2 Ghz Intel Prozessor mit einem Kern etwa 5 Sekunden.

### 5.4.12 Evaluierung

Verschiedene Sänger haben eine Melodie in das Mikrofon gesummt. Das Spektrum wurde mit dem in Kapitel 4 vorgestellten Filter geglättet. Aus den aufgenommenen Audiodaten wurden die Grundfrequenz mithilfe der AKF+COMB oder der Autokorrelationsmethode bestimmt. Anschließend wurden die Rhythmusextraktion und die Melodieextraktion durchgeführt. Die gewonnenen Noten-Daten für  $k=1$  wurden dann manuell gezählt und ausgewertet.

**Die klassische Autokorrelation hat in 71% der Fälle die gesummteten Noten korrekt erkannt. Es wurde häufig die halbe Grundfrequenz oder die minimal zulässige verschobene Folge detektiert.**

**Der AKF+COMB Algorithmus hat in 98% der Fälle die Noten korrekt erkannt.**

**Das Grundfrequenzerkennungsproblem für Query-by-Humming Anwendungen wurde gelöst.**

	Melodielänge in Noten	Korrekt erkannte Noten mit Autokorrelation	Korrekt erkannte Noten mit AKF+COMB
Melodie 1, Sänger 1, männlich	19	16	19
Melodie 2, Sänger 5, weiblich	12	10	12
Melodie 3, Sänger 4, weiblich	21	17	21
Melodie 4, Sänger 5, weiblich	11	11	11
Melodie 5, Sänger 1, männlich	11	11	11
Melodie 6, Sänger 1, männlich	23	15	21
Melodie 7, Sänger 1, männlich	16	6	16
Melodie 8, Sänger 5, weiblich	23	16	23
Melodie 9, Sänger 4, weiblich	13	10	13
Melodie 10, Sänger 2, männlich	24	17	24
Melodie 11, Sänger 2, männlich	16	5	16
Melodie 12, Sänger 1, männlich, erkältet	26	16	26
Melodie 13, Sänger 1, männlich	15	11	15
Melodie 14, Sänger 3, männlich	17	13	16
Melodie 15, Sänger 3, männlich	19	13	18
Melodie 16, Sänger 5, weiblich	18	14	18
Melodie 17, Sänger 3, männlich	21	16	21
Noten insgesamt	305	217	301
<b>Erkennungsrate</b>		<b>71%</b>	<b>98%</b>

*Tabelle 5.1: Vergleich - Notenerkennung mit AKF+COMB oder Autokorrelation*

## 6 Automatisierte Rhythmuserkennung

### 6.1 Abstrakt

Die automatisierte Rhythmuserkennung findet die Taktgeschwindigkeit und den Startzeitpunkt einer gesumnten Melodie.

Es wurde ein Modul entwickelt, das die Taktgeschwindigkeit und den Startzeitpunkt eines Taktes findet. Die Methodik funktioniert bei synthetischen Melodien zuverlässig, versagt jedoch bei gesungenem Material.

Es wird ein Algorithmus vorgestellt, der auf einer linearen Zeitreihe die Phasenlage und Amplitude einer beliebigen Frequenz in linearer Laufzeit berechnet.

*Durch die automatische Takterkennung kann auf das Dynamic Time Warping im Matching Modul verzichtet werden und der Ähnlichkeitsvergleich effizienter gelöst werden. [Maz 01], [Jan 01], [Zhu 01]*

*Die automatisierte Segmentierung von gesumnten Melodien ist eine sehr anspruchsvolle Aufgabe. [Tol 00]*

### 6.2 Problemstellung

Existierende Query-by-Humming Systeme geben dem Sänger mit einem Metronom einen Rhythmus vor. Der Summer muss dann passend zum Takt die Melodie singen. Die Grundfrequenz des Gesangs wird anschließend im Takt zu den Noten quantisiert. Dadurch muss für das QbH System keine Rhythmuserkennung implementiert werden.

Diese Methodik hat jedoch Nachteile:

- Das Metronom ist ein zusätzliches Audiosignal, das die Gesangsaufnahme stören kann.
- Die Geschwindigkeit von Liedern ist unterschiedlich. Der Sänger muss die BPM Rate am Metronom vor dem Singen einstellen. BPM Raten richtig einzuschätzen gelingt aber nur erfahrenen Musikern.
- Für die Einstellung des Metronoms ist neben dem Microphon zusätzliche Eingabehardware nötig.

Es werden in diesem Abschnitt zwei Algorithmen vorgestellt, die den Rhythmus von Gesang erkennen.

Was das menschliche Ohr als Gesangs- oder Sprechrhythmus wahrnimmt, sind zyklisch wiederkehrende Energiemaxima im Spektrum. Die automatisierte Rhythmuserkennung ist die Suche nach Zyklizität in der Energieverteilung einer linearen Zeitreihe.

### 6.3 Rhythmuserkennung auf FFT Basis

Im folgenden wird ein Algorithmus zur Rhythmuserkennung auf FFT Basis vorgestellt. Er wurde ursprünglich zur Extraktion von beat-Features für MPEG7 Datenbanken entwickelt und wird in der aktuellen Version des QbH Systems aus Gründen der Performance und Präzision nicht mehr verwendet. Daher wird hier die Funktionsweise nur kurz erläutert.

Zum Erzeugen des Spektrogramms wird eine FFT (Abtastrate 44,1 kHz, Fenstergröße 1024 Samples, Kaiserfenster mit beta 5, 50% overlap) auf dem Audiosignal durchgeführt. Da die FFT Frequenzen unterschiedlich genau auflöst, wird das Spektrum im Frequenzbereich mit 20-fach oversampling logarithmiert. 6 Zeilen im logarithmierten Bild entsprechen einer Frequenzverdoppelung.

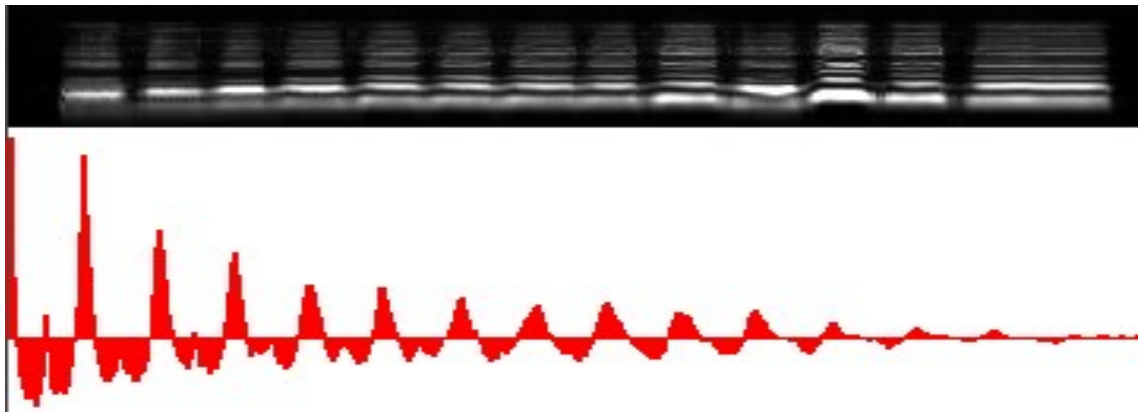


Abbildung 6.1:

Oben: Logarithmiertes FFT Spektrogramm zu „Fuchs, du hast die Gans gestohlen“; X-Achse: Zeit (0-9 Sekunden), Y-Achse: Frequenz (50-14000 Hz)  
Unten: Zugehöriges Histogramm über die Autokorrelation des FFT Spektrogramms; X-Achse: Verschiebung in Pixeln; Y-Achse: 2. Ableitung der Häufigkeitsverteilung

Anschließend wird eine Autokorrelation des logarithmierten FFT Spektrogramms in Richtung der X-Achse durchgeführt. Aus den Daten der Autokorrelation lässt sich ein Histogramm über die Häufigkeitsverteilung der Energiemaxima in Abhängigkeit von deren zeitlichen Verschiebung erstellen. Um die Maxima im Histogramm besser erkennbar zu machen, wird die zweite Ableitung gebildet. Im obigen Bild vom abgeleiteten Histogramm entspricht das erste Maximum der Verschiebung um 0 und das zweite Maximum der gesuchten BPM Rate.

### 6.3.1 Schwächen der Methodik

- Die Komplexität für die Autokorrelation des Spektrogramms ist  $O(n^3)$ .
- Bei der Autokorrelation geht die Phaseninformation verloren. Der Algorithmus kann somit zwar die Taktgeschwindigkeit bestimmen, er kann aber nicht den Zeitpunkt erkennen, wann der Takt beginnt.
- Bei wenigen Analysedaten wird die Histogramm-Methode zunehmend ungenau.

## 6.4 Takterkennung auf Basis der Autokorrelation

Der Algorithmus zur automatisierten Takterkennung soll folgende Eigenschaften erfüllen:

- Zuverlässige Erkennung der BPM Rate (Frequenz)
- Erkennen des Startzeitpunktes eines Taktes (Phaseninformation)
- Robustheit
- Gute Performance

In der vorhergehenden Suche nach der Grundfrequenz wurde bereits ein AKF+COMB Spektrogramm erstellt. Es liegt nahe, dieses existierende Spektrogramm für die Takterkennung wieder zu verwenden.

### 6.4.1 Bestimmung der Taktgeschwindigkeit

Im ersten Schritt werden die Energien der im AKF+COMB Spektrogramm enthaltenen Grundfrequenzen in Y-Richtung aufsummiert. Dadurch wird aus dem zweidimensionalen Bild eine eindimensionale Zeitreihe (im Bild unten rot eingefärbt).

Eine andere geeignete Methodik zum Erstellen einer Zeitreihe über die Gesamtenergieverteilung ist die spektrale Leistungsdichte (PSD = power spectral density).

Eine FFT eignet sich zur Suche nach Zyklizität auf der Funktion über die Gesamtenergie des Spektrums weniger, da die Länge der Zeitreihe von der Dauer der gesungenen Melodie abhängt und nur ein kleiner Frequenzbereich untersucht werden soll.

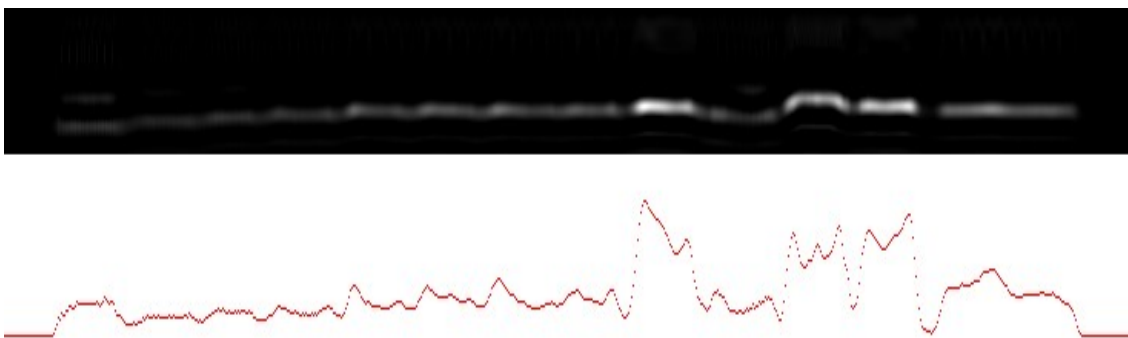


Abbildung 6.2:

Oben: AKF+COMB Spektrogramm; „Fuchs, du hast die Gans gestohlen“; X-Achse: Zeit (0-8 Sek.); Y-Achse: Grundfrequenz (40 – 2500 Hz);

Unten: „Energie des Spektrums“; X-Achse: Zeit (0-8 Sek.); Y-Achse: Energie

Der Beginn und das Ende von Noten sind signifikante Rhythmusmerkmale in einem Lied. Daher wird im zweiten Schritt die Zeitreihe über die Energie des Spektrums differenziert. Durch das Differenzieren wird erreicht, dass die Startzeitpunkte von Noten mit positiven Werten (Energie steigt) und die Endzeitpunkte von Noten mit negativen Werten (Energie fällt) bewertet werden. Allgemein erhalten Änderungen in der Energieverteilung durch das Differenzieren eine höhere Gewichtung.

Die differenzierte Energieverteilung wird nun nach Zyklizität durchsucht. Da die Melodien von Liedern nicht beliebig schnell oder langsam gesungen werden, kann der Suchbereich auf 90-180 BPM (1,5 - 3 Hz) beschränkt werden. Die starke Beschränkung des Suchbereichs ermöglicht eine sehr effiziente Implementierung.

Es wird eine normierte Autokorrelation auf dem beschränkten Suchbereich durchgeführt. Das Ergebnis der Autokorrelation ist eine Funktion, die Aussagen über die Häufigkeitsverteilung der im Lied vorkommenden Taktgeschwindigkeiten ermöglicht.

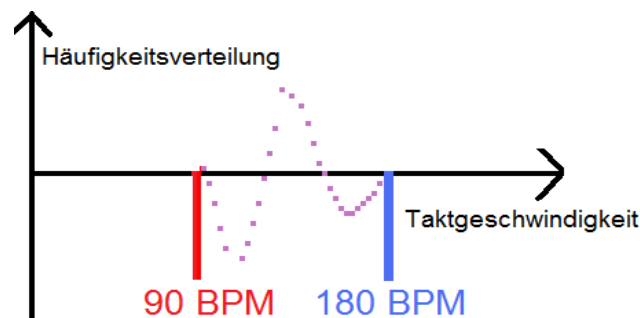


Abbildung 6.3: Autokorrelation der differenzierten Gesamtenergie; Analysedaten: „Fuchs, du hast die Gans gestohlen“

**Die gesuchte Taktgeschwindigkeit entspricht dem Maximum der Autokorrelationsfunktion der differenzierten Gesamtenergie.**

Bei den oben analysierten Daten wird dies bei etwa 125 BPM erreicht.

Der Algorithmus der Autokorrelation zur Bestimmung der Taktgeschwindigkeit lautet (vereinfacht):

```
//Für alle Verschiebungen
for (int xdelt=delaymin;xdelt<=delaymax;xdelt++)
{
float sum=0;
for (int x=0;x<xMax;x++)
sum+=ener[x]*ener[x+xdelt];
//Normieren
sum /= xMax;
beatHist[xdelt]= sum;
}
```

## 6.4.2 Bestimmung des Taktzeitpunkts

Durch die AKF konnte die Taktgeschwindigkeit (Frequenz) bestimmt werden. Die Phaseninformation ging aber durch die Anwendung der AKF verloren. Um die Gesangsmelodie zu einem späteren Zeitpunkt vernünftig quantisieren zu können, muss aber auch der Zeitpunkt bekannt sein, zu dem der erste Takt beginnt.

Die Frequenz ist also bereits bekannt, die Phase wird noch gesucht. Durch das Wissen über die Frequenz kann die Komplexität des folgenden Rechenschritts stark vereinfacht werden. Anstatt eine komplette DFT durchzuführen, kann man sich hier auf die Analyse eines einzigen spektralen Bandes mit der BPM Rate als Frequenz beschränken.

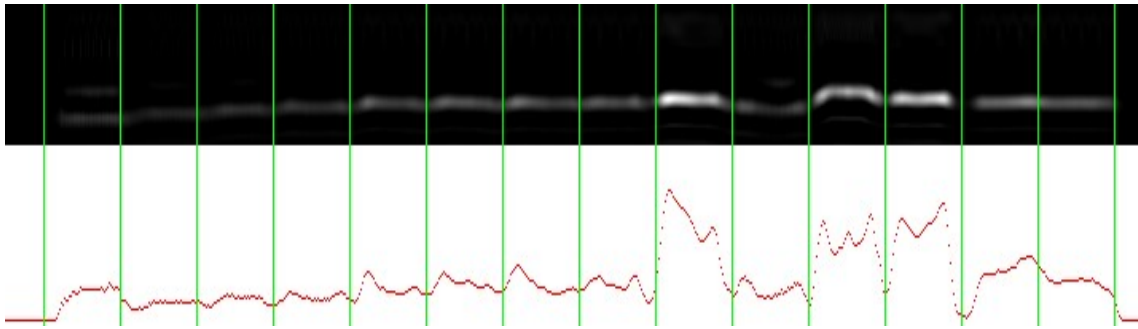
Wie bei der Fourier Transformation wird hier das Signal mit dem Sinus und dem Cosinus gefaltet. Mithilfe der Arcustangensfunktion kann aus dem Quotienten der komplexen Summen die Phase errechnet werden. Da durch die vorhergehende Differenzierung des Signals eine Phasenverschiebung um  $90^\circ$  erfolgt ist, muss für den Anwendungsfall der Takterkennung  $\pi/2$  von der Phase subtrahiert werden.

Der Pegel (Radius) errechnet sich aus der euklidischen Distanz der komplexen Summen.

**Der Algorithmus berechnet die Phasenlage und Amplitude einer einzelnen, beliebigen Frequenz auf einer beliebig langen linearen Zeitreihe in  $O(n)$ .**

Er lautet (vereinfacht und verallgemeinert):

```
float sinsum=0;
float cossum=0;
for (int t=0;t<laenge;t++)
{
    float ph = 2 * Pi * (float)t * freq;
    sinsum += sin(ph) * sample[t];
    cossum += cos(ph) * sample[t];
}
float phase = atan2(sinsum,cossum);
float ampl = sqrt(sinsum*sinsum + cossum*cossum);
```



**Abbildung 6.4:** Automatisierte Takterkennung in „Hummel“; Grün: Startzeitpunkte des Taktes  
 Oben: AKF+COMB Spektrogramm; „Fuchs, du hast die Gans gestohlen“; X-Achse: Zeit (0-8 Sek.); Y-Achse: Grundfrequenz (40 – 2500 Hz);  
 Unten, rot: „Energie des Spektrums“; X-Achse: Zeit (0-8 Sek.); Y-Achse: Energie

Im oberen Bild wird nochmals die komplette, automatisierte Takterkennung gezeigt:

- Aus dem schwarzen AKF+COMB Spektrogramm wird die rote Energieverteilungsfunktion erzeugt.
- Die Energieverteilungsfunktion wird durch Autokorrelation auf Zyklizität durchsucht und die Taktgeschwindigkeit bestimmt (Distanz der grünen Linien zueinander).
- Mithilfe der Frequenzinformation werden die Startzeitpunkte der Takte (grüne Linien) errechnet.

### 6.4.3 Komplexität

Die Komplexität für die Bestimmung der Taktgeschwindigkeit auf einer PSD Zeitreihe ist  $O(n \cdot m)$ , wobei  $n$  der Länge der Zeitreihe und  $m$  der Zahl der zu untersuchenden Verschiebungen entspricht. In der Regel gilt  $n \gg m$ .

Die Komplexität für die Bestimmung der Taktzeitpunkte auf Basis der vorangehenden Berechnung ist  $O(n)$ , wobei  $n$  der Länge der Zeitreihe entspricht.

### 6.4.4 Anwendung in der Praxis

Der Algorithmus auf Basis der Autokorrelation hat die Problemstellung einer automatisierten Takterkennung für percussive Rhythmen und synthetische Klänge gelöst. Er hat sich in der praktischen Anwendung als performant, zuverlässig und robust erwiesen.

Bei gesungenem Material treten Probleme auf. Popgeräusche bei der Aufnahme können aufgrund der hohen spektralen Energie die Takterkennung aus dem Takt werfen. Daher ist die Verwendung von Headset- oder „Plopschutzmicrophonen“ empfehlenswert.

Je länger eine gesungene Passage ist, umso wahrscheinlicher wird es, dass ein Sänger aus dem Takt gerät und seine Geschwindigkeit erhöht oder verlangsamt. Da für QbH Systeme nur kurze Passagen mit einer Länge von wenigen Sekunden gesungen werden, ist dies vernachlässigbar.

### Versuchsaufbau:

Ein Metronomsignal mit verschiedenen Taktgeschwindigkeiten wurde aufgezeichnet. Über einen Kopfhörer wurde den Versuchspersonen das Taktsignal vorgespielt. Die Versuchspersonen mussten passend zur Taktgeschwindigkeit mehrere zufällige Melodien in ein Mikrofon summen. Das Signal wurde aufgezeichnet und mit der automatischen Takterkennung auf Basis des AKF+COMB Spektrogramms ausgewertet.

Analysedaten	Reelle rate	BPM	Erkannte BPM	Abweichung
Gesummte Melodie 1	100		93	8%
Gesummte Melodie 2	100		96	4%
Gesummte Melodie 3	100		99	1%
Gesummte Melodie 4	120		99	21%
Gesummte Melodie 5	120		105	14%
Gesummte Melodie 6	120		117	3%
Gesummte Melodie 7	120		101	19%
Gesummte Melodie 8	120		87	38%
Gesummte Melodie 9	120		99	21%
Gesummte Melodie 10	120		101	19%
Gesummte Melodie 11	140		81	73%
Gesummte Melodie 12	140		81	73%
Gesummte Melodie 13	140		82	73%
Gesummte Melodie 14	140		81	73%
Gesummte Melodie 15	160		117	37%
Gesummte Melodie 16	160		83	93%
Gesummte Melodie 17	160		82	95%

*Tabelle 6.1: Automatische Takterkennung bei gesummen Melodien*

Das Ergebnis war nicht zufriedenstellend. In nur vier Fällen wurde die Geschwindigkeit mit einer Toleranz von 10% „richtig“ erkannt (grün). In zwei Fällen wurde genau die halbe Geschwindigkeit gefunden (blau).

Die Methode die Energien der im AKF+COMB Spektrogramm enthaltenen Grundfrequenzen in Y-Richtung aufzusummieren, ist für die Rhythmuserkennung von gesummtem Material ungeeignet. Die Verwendung der Kurzzeitenergie könnte hier bessere Ergebnisse liefern.

### Versuchsaufbau:

Von verschiedenen, im Handel erhältlichen, „Sample CDs“ wurden kurze Gesangspassagen mit fest definierten Taktgeschwindigkeiten ausgewählt und mit der automatischen Takterkennung auf Basis des AKF+COMB Spektrogramms ausgewertet.

Analysedaten	Reelle rate	BPM	Erkannte BPM	Abweichung
Gesang 1	120		92	30%
Gesang 2	120		123	2%
Gesang 3	120		83	44%
Gesang 4	125		103	21%
Sprechgesang 1	125		131	5%
Gesang 5	130		136	5%
Sprechgesang 2	140		92	52%
Sprechgesang 3	140		161	15%
Gesang 6	140		112	25%
Gesang 7	140		113	24%

*Tabelle 6.2: Automatische Takterkennung bei gesungenen Melodien*

Auch bei Gesang versagt die Methode. Frikative stören hier die Takterkennung zusätzlich.

### Versuchsaufbau:

Von verschiedenen, im Handel erhältlichen, „Sample CDs“ wurden kurze synthetische Rhythmuspassagen mit fest definierten Taktgeschwindigkeiten ausgewählt und mit der automatischen Takterkennung auf Basis des AKF+COMB Spektrogramms ausgewertet.

Analysedaten	Reelle rate	BPM	Erkannte BPM	Abweichung
Percussiver Rhythmus	120		117	3%
Percussiver Rhythmus	120		120	0%
Percussiver Rhythmus	120		80	50%
Percussiver Rhythmus	120		120	0%
Percussiver Rhythmus	120		117	3%
Percussiver Rhythmus	126		126	0%
Percussiver Rhythmus	126		126	0%
Percussiver Rhythmus	130		87	49%
Percussiver Rhythmus	130		87	49%
Percussiver Rhythmus	130		84	55%
Percussiver Rhythmus	126		126	0%
Percussiver Rhythmus	140		140	0%
Percussiver Rhythmus	140		140	0%
Percussiver Rhythmus	140		132	6%
Percussiver Rhythmus	160		161	1%
Percussiver Rhythmus	160		161	1%
Percussiver Rhythmus	160		161	1%
Percussiver Rhythmus	160		152	7%

*Tabelle 6.3: Automatische Takterkennung bei synthetischen Melodien*

Bei synthetischen Klängen funktioniert die Methode zuverlässig. In einigen Fällen (blau) wurden genau zwei Drittel der gesuchten Geschwindigkeit detektiert. Die Untersuchung der Daten zeigte, dass in diesen Testsignalen besonders viele punktierte Noten existierten. Das „Punktierungsproblem“ kann durch den Einsatz der in Kapitel 5 vorgestellten Kamm-Methode minimiert werden. Um die Robustheit gegenüber punktierten Noten zu steigern, müssen in die Bewertung der Autokorrelationsfunktion auch sekundär-Maxima bei der 1,5-fachen Verschiebung mit einfließen.

Aus Gründen des Umfangs dieser Diplomarbeit kann hier nicht mehr weiter auf die Takterkennung auf Basis der Kurzzeitenergie und die detaillierte Lösung des „Punktierungsproblems“ eingegangen werden.

## 7 Melodieextraktion

### 7.1 Abstrakt

In der Melodieextraktion wird für den Zeitpunkt jeder Viertelnote eine Liste mit  $k$  gewichteten Noten extrahiert.

Für die automatisierte Melodieerkennung wird ein Algorithmus vorgestellt, der zuverlässig kontinuierliche Grundfrequenzverläufe in diskrete Notenwerte quantisiert.

**„Eine gute Notenerkennung ist ausschlaggebend für die Erkennungsqualität von QbH Systemen.“ [Tol 00]**

### 7.2 Die Auswirkung der Unschärferelation

Die Helligkeit im AKF+COMB Spektrogramm entspricht der Wahrscheinlichkeit für das Auftreten einer Grundfrequenz  $y$  zum Zeitpunkt  $x$ .

Wenn man das Bild unten genau betrachtet, ist eine Unschärfe sowohl im Zeitbereich, als auch im Frequenzbereich erkennbar. Die Unschärfe im Zeitbereich ist vernachlässigbar, da die Melodie später auf ein im Verhältnis dazu sehr grobes Raster quantisiert werden wird. Die Unschärfe im Frequenzbereich ist für die späteren Berechnungen nicht vernachlässigbar.

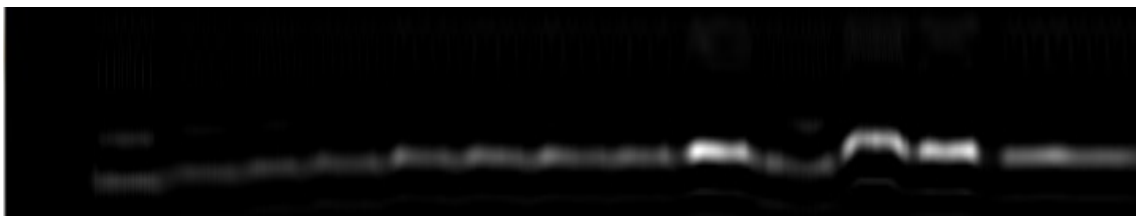


Abbildung 7.1: AKF+COMB Spektrogramm; Grundfrequenzverlauf von „Fuchs du hast die Gans gestohlen“; X-Achse: Zeit (0-8 Sek.); Y-Achse: Grundfrequenz (40 – 2500 Hz);

Die Unschärfe im Frequenzbereich entsteht durch:

- Grundfrequenzschwankungen in der menschlichen Stimme
- Falsch gesungene Töne
- Die AKF+COMB Transformation

Bei der AKF+COMB kann eine größere Schärfe im Frequenzbereich erzielt werden, indem die Autokorrelation über eine größere Anzahl von Zyklen durchgeführt wird.

Wie bei der Verallgemeinerung der Heisenbergschen Unschärferelation gilt hier: Je größer die Zahl der Zyklen gewählt wird, desto genauer kann die Frequenz aufgelöst werden, jedoch sinkt die Schärfe der Zeitauflösung. In der Praxis hat aber eine Faltung über mehr als 6 Zyklen bei der Grundfrequenzanalyse von Sprachdaten zu keiner Verbesserung der Extraktionsqualität geführt, sondern lediglich die Performance verschlechtert.

Anzahl der Zyklen	Erkennungsqualität Melodie 1 in %	Erkennungsqualität Melodie 2 in %	Erkennungsqualität Melodie 3 in %	Durchschnitt in %
1	10	8	9	9
2	16	7	10	11
3	14	8	11	11
4	14	10	15	13
6	17	11	17	15
8	14	11	15	13,33
10	13	11	15	13
12	14	10	15	13
16	15	11	15	13,67
32	15	10	15	13,33
48	16	9	14	13
64	14	9	13	12

*Tabelle 7.1: Extraktionsqualität für gesungene Melodien in Abhängigkeit von der Anzahl der Autokorrelationszyklen (Kaiser-Fenster mit  $\beta=5$ )*

### 7.3 Unsichere Noten

Der Grundfrequenzverlauf der gesungenen Melodie muss von einer nahezu kontinuierlichen Funktion (Bild 43) in eine diskrete quantisiert werden. Pro Takteinheit werden die  $k$  wahrscheinlichsten Noten mit einer Gewichtung extrahiert. Durch das Betrachten von mehr als einer Note zu jedem Zeitpunkt wird die Extraktionsqualität verbessert.

Durch diese Methodik wird erreicht, dass Schwankungen in der Grundfrequenz („leicht falsch gesungene Töne“) auch noch brauchbare Informationen für die Suche beitragen.

Bei sehr tiefen Stimmen mit hohem Energieanteil in der ersten Harmonischen ist sich der Algorithmus zur Grundfrequenzsuche nicht immer sicher. Im unteren Bild ist beim ersten Ton auch Energie bei der doppelten Frequenz. Dies ist an den zwei übereinander liegenden, horizontalen Linien erkennbar.

Da die Muskeln unserer Sprachorgane träge sind und sich die bewegten Massen nicht beliebig schnell bewegen können, kann die Grundfrequenz der menschlichen Stimme nicht sprunghaft von Note zu Note verändert werden. Stattdessen ergibt sich ein kontinuierlicher, von Note zu Note gleitender Verlauf. Besonders deutlich wird das im rechten Drittel des unteren Bildes. Dieser An- und Abglitt kann die Suchergebnisse verschlechtern, weil die Mittelwerte der Grundfrequenzen im quantisierten Raster nach oben erhöht oder unten erniedrigt werden. Das kann zu einer „leicht falsch gesungenen Note“ führen. Durch die „k-wahrscheinlichsten Noten Methode“ wird auch hier noch versucht, mehr brauchbare Information zu extrahieren.

## 7.4 Rasterung

Im Modul der Takterkennung auf Basis der Autokorrelation ist der Startzeitpunkt (Phase) und die Taktgeschwindigkeit (Frequenz) bereits bestimmt worden. Anhand dieser Werte wird das Quantisierungsraster angelegt. Jede Rastereinheit entspricht später genau einer Viertelnote. Innerhalb jeder Rastereinheit werden die k wahrscheinlichsten Noten anhand der gewichteten Grundfrequenzen ermittelt. Dazu werden die Wahrscheinlichkeiten jeder Grundfrequenz mit einem Hammingfenster multipliziert und aufsummiert. Durch die Fensterung wird erreicht, dass den Grundfrequenzen nahe am Mittelpunkt eine größere Bedeutung zukommt, als denen im Randbereich. Wenn viele Unsauberkeiten im Randbereich vorkommen, wird eine Verbesserung der Extraktionsqualität erreicht.

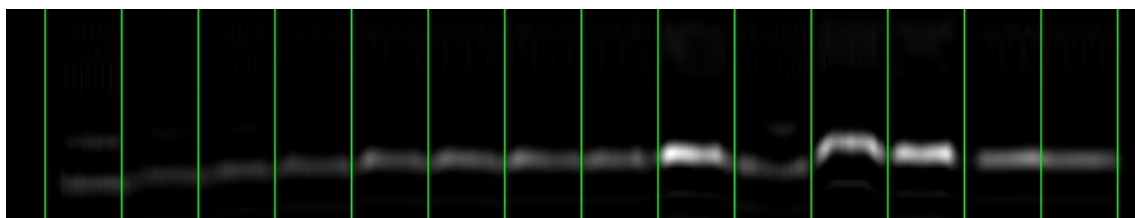


Abbildung 7.2: Gerastertes AKF+COMB Spektrogramm; Grundfrequenzverlauf von „Fuchs, du hast die Gans gestohlen“; X-Achse: Zeit (0-8 Sek.); Y-Achse: Grundfrequenz (40 – 2500 Hz);

Für jeden Vierteltakt sind nun alle Wahrscheinlichkeiten für die Existenzen der Noten  $Y$  gegeben, jedoch werden aus Optimierungsgründen nur die k wahrscheinlichsten Noten für die Weiterverarbeitung verwendet. Um die k wahrscheinlichsten Noten zu bestimmen, werden alle Noten in jedem Vierteltakt nach ihrer Wahrscheinlichkeit sortiert und die k besten selektiert.

Der Algorithmus zur Quantisierung und Berechnung der Wahrscheinlichkeiten von Noten lautet (vereinfacht):

```
//Für jede Viertelnote
for (long bar=0;bar<numbars;bar++)
{
    //summiere im Takt
    for (long y=0;y<bildHoehe;y++)
    {
        sum[y]=0;
        for (long x=0;x<Taktlaenge;x++)
        {
            float p = bild[x + bar * Taktlaenge + Taktstart][y];
            //Hamming Fenster +90°
            sum[y] += ( 0.04 + 0.46 * sin( x / Taktlaenge * Pi ) ) * p;
        }
        ...
    }
}
```

## 7.5 Logarithmierung der Wahrscheinlichkeiten

Eine logarithmische Gewichtung der Wahrscheinlichkeiten führte zu einer Verbesserung der Erkennungsrate.

„Ein Nebeneffekt der Logarithmierung ist die Anpassung an die menschliche Wahrnehmung der Frequenzamplituden“. [Wes 04]

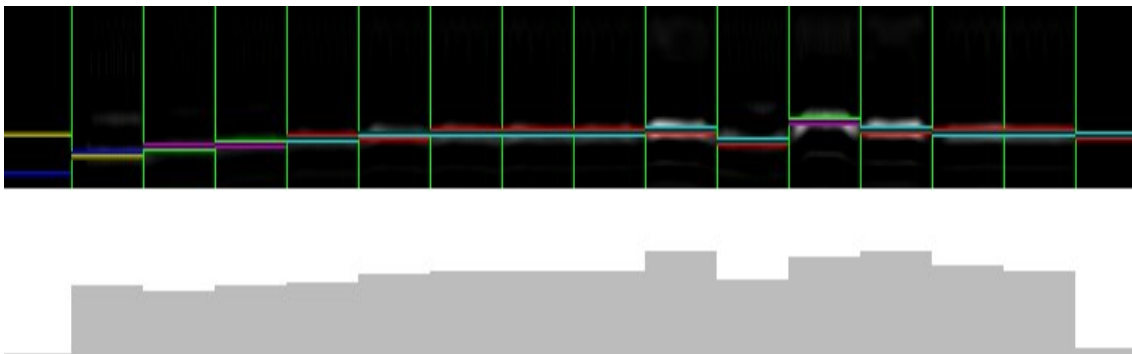


Abbildung 7.3: Notenextraktion aus dem AKF+COMB Spektrogramm; „Fuchs, du hast die Gans gestohlen“; X-Achse: Zeit (0-8 Sek.); Y-Achse: Grundfrequenz (40 – 2500 Hz);

Oben: Vertikale Linie blau: Note k=1; Vertikale Line Grün: Note k=2; Vertikale Line Rot: Note k=3

Unten, grau: X-Achse: Zeit (0-8 Sek.); Y-Achse: Logarithmierte Gewichtung der Note k=1

## 7.6 Die richtige Wahl des k-Wertes

Die Selektierung und Weiterverarbeitung der 5 wahrscheinlichsten Noten hat in der Praxis zu den besten Erkennungsraten geführt. Aus diesem Wert lässt sich erkennen, dass ein durchschnittlich begabter Sänger häufig um bis zu  $\pm 2$  Halbtöne um die Referenztonhöhe schwankt. Größere Werte haben zu einer Verschlechterung geführt. Die Verschlechterung ist damit zu erklären, dass bei einer zu großzügig gewählten Betrachtung der Tonhöhe zu viele Lieder in der Datenbank der gesungenen Melodie ähnlich sind.

Anzahl der extrahierten Noten pro Takt (k)	Erkennungsrate Melodie 1, Sänger 1 in %	Erkennungsrate Melodie 2, Sänger 2 in %	Erkennungsrate Melodie 3, Sänger 3 in %	Durchschnitt in %
1	18	20	16	18
2	20	22	18	20
3	19	22	11	20,67
4	23	26	28	25,67
5	33	34	30	32,33
6	12	13	15	13,33
7	10	12	11	11
8	7	9	5	7

Tablelle 7.2: Erkennungsrate für gesungene Melodien in Abhängigkeit von der Anzahl der extrahierten Noten bei logarithmischer Gewichtung

## 7.7 Export

In „Hummel“ kann die extrahierte Melodie dann zur Weiterverarbeitung und Analyse in der Software von Drittherstellern ins Midi Format gespeichert werden. Die Gewichtungen der einzelnen Noten werden auf einen Wertbereich von 0-127 normiert und als „velocity“ (=Lautstärke) gespeichert.

## 7.8 Weitere Verbesserungsmöglichkeiten

Die europäische Musik hat zwar 12 Noten, jedoch werden davon nur 7 genutzt (Tonarten). Dadurch ließe sich das Pitchtracking gröber rastern und der Suchraum verkleinern. Die aktuellen QbH Systeme nutzen diese Eigenschaft nicht aus.

Verallgemeinert lassen sich durch das zusätzliche Weltwissen über Tonarten detaillierte Annahmen über die Wahrscheinlichkeiten einzelner Noten in den Melodien machen. Für Dateien in der Midi Datenbank lässt sich die Tonart relativ leicht bestimmen, bei schlecht gesungenen Melodien wird diese Aufgabe anspruchsvoller.

## 8 Datenbank

### 8.1 Abstrakt

„Midi“ ist ein Datenbankformat, das sich gut für QbH Systeme eignet. Um gute Suchergebnisse zu erzielen, sollten alle Lieder in der Datenbank per Hand vorselektiert und vorbearbeitet werden. Der Implementierungsaufwand einer zuverlässigen Importfunktion für Midi Dateien ist hoch.

### 8.2 Midi

Def.: MIDI

*„(engl.: musical instrument digital interface ... = „Digitale Schnittstelle für Musikinstrumente“) ist ein Datenübertragungs-Protokoll zum Zwecke der Übermittlung, Aufzeichnung und Wiedergabe von musikalischen Steuerinformationen zwischen Instrumenten oder Computern. ..“ [WIK 08]*

In „Hummel“ wurde als Dateiformat für die Liederdatenbank das Midi Format gewählt.

Vorteile:

- Seit 1983 definierter Standard
- Geringer Speicherplatzbedarf
- Flexibilität
- Verfügbarkeit einer Vielzahl von Liedern im Netz
- Schnelle, maschinenfreundliche Verarbeitung
- Hervorragende Soft-und Hardwareunterstützung

Nachteile:

- Herstellerspezifische Eigenheiten
- Komplexer Import
- Format ist für den Menschen nicht lesbar
- Melodien werden von den Autoren oft unnötig mit zusätzlichen Noten „ausgeschmückt“ und müssen nachbearbeitet werden

Für das Testsystem wurde eine Datenbank von 100 bekannten Kinderliedern erstellt. Kinderlieder sind häufig Volksgut und frei von Urheberrechten. Im Midi-Quellformat sind in „Hummel“ pro Lied durchschnittlich 3,2 kB Speicherplatz nötig.

Um gute Suchergebnisse zu erzielen, sollten alle Lieder in der Datenbank per Hand vorselektiert und vorbearbeitet werden. Nur die Hauptmelodie sollte gespeichert werden. Akkorde, Rhythmusinstrumente und ausschmückende

Nebenmelodien sind für die Suche irrelevant. Sie vergrößern den Suchraum und verschlechtern somit das Suchergebnis und die Performance. Bei professionellen Systemen sollte daher der personelle Aufwand für die Aufbereitung der Datenbank per Hand mit berücksichtigt werden.

Durch die manuelle Vorverarbeitung benötigt eine einminütige, einstimmige Hauptmelodie mit einer Auflösung von halben Noten im Rohformat nur noch etwa 200 Bytes an Speicherplatz.

Im Rahmen einer Diplomarbeit ist eine komplette, manuelle Vorverarbeitung einer großen Datenbank nicht machbar. Daher wurden für „Hummel“ die verwendeten Lieder nur vorgehört und vorselektiert. Der größere Suchraum musste für das experimentelle System in Kauf genommen werden.

### **8.3 Import von Midi Daten**

Das Midi Format ist sehr flexibel und komplex. Besonders herstellereigene Eigenschaften machen die Implementierung der Importfunktion aufwendig. Auf technische Details der Implementierung soll daher hier nicht eingegangen werden.

#### **8.3.1 Realisierung**

Die Importfunktion lädt eine beliebige Midi Datei ein. Aus dem Header wird die Geschwindigkeit (BPM Rate) extrahiert. Entsprechend der BPM Rate wird das Lied in Viertelnoten rasterisiert. Note-On/Off Events definieren den Beginn und das Ende der einzelnen Noten. Die Hauptmelodie ist häufig im ersten oder zweiten Track zu finden, daher wird nach der Nummer des niedrigsten Tracks gesucht. Nur die Noten der Hauptmelodie werden für die weitere Suche verwendet. Velocity Daten und sonstige Events werden beim Import von Midi Liedern ignoriert, da sie in der Praxis keine für die Suche relevante Information beinhalten.

Nach dem Import steht zur Weiterverarbeitung die Information über die Geschwindigkeit, die Information über die Länge des Liedes und eine Matrix aus Noten zur Verfügung. Die Notenmatrix ist in Viertelnoten gerastert und definiert für jeden Zeitpunkt X eine Menge von Noten mit der Tonhöhe Y. Ist die Mächtigkeit der Menge zum Zeitpunkt X gleich Null, existiert zum gegebenen Zeitpunkt gerade keine Note. Es herrscht also gerade Ruhe.

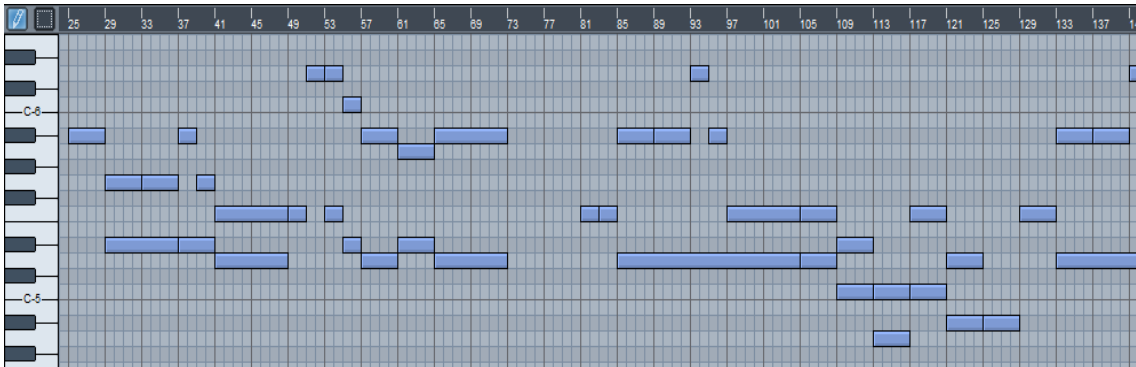


Abbildung 8.1: Polyphone Notenmatrix zu „Die Gedanken sind frei“ in der Sequencersoftware Orion; X-Achse: Zeit (25-140 Sek.); Y-Achse: Tonhöhe (G4-E6)

Um die Weiterverarbeitung zu beschleunigen, kann es bei sehr großen Datenbanken sinnvoll sein, die Notenmatrizen in einem Rohformat zu speichern. Somit entfällt ein Teil des komplexen Midi Imports. Darüber hinaus kann die Robustheit des Programms verbessert werden, da potentielle Buffer overflows, für die das Midi Format aufgrund des komplexen Formats prädestiniert ist, vermieden werden können.

Die Speicherung in Form von Listen, die alle zum Zeitpunkt X gespielten Noten definieren, verkleinert den Speicherplatzbedarf gegenüber von Matrizen etwa um den Faktor 50 (128 Midi Noten, durchschnittlich 2 Noten pro Zeiteinheit + Listenlänge).

## **9 Melodievergleich**

### **9.1 Abstrakt**

Im Vergleichsmodul wird die gesummte Melodie mit allen Melodien in der Datenbank verglichen.

Es konnte kein Algorithmus gefunden werden, der den Melodievergleich effizient löst und zugleich eine hohe Erkennungsrate hat.

Für den Melodievergleich konnte eine Distanztabelle gefunden werden, welche die Erkennungsqualität steigert.

### **9.2 Problemstellung**

Bei Gesungenem handelt es sich um sehr unsaubere Daten. Nur sehr selten entsprechen die Noten in der gesummten Melodie genau den Noten in der Midi-Datenbank.

Im Vergleichsmodul wird die gesummte Melodie mit allen Melodien in der Datenbank verglichen. Anhand der Vergleichswerte sollen die ähnlichsten Lieder zum Gesungenen gefunden werden.

Jeder paarweise Vergleich zwischen der gesummten Melodie und einer Melodie aus der Datenbank liefert einen Score. Je höher der Score ist, desto ähnlicher sind sich die beiden.

Der Score ist ein relatives Maß. Er ist abhängig von der gesummten Melodie, allen Melodien in der Datenbank und den Parametern von fast allen Algorithmen in QbH System. Er liefert nur die Aussage darüber, dass das Gesummte G der Midi Datei A ähnlicher ist, als der Midi Datei B. Anhand des Scores können später die Lieder der Ähnlichkeit nach sortiert werden. Lieder mit einem hohen Score entsprechen mit höherer Wahrscheinlichkeit dem gesuchten Lied als Lieder mit niedrigem Score.

Werden alle Scores einer größeren Datenbank zu einer gesummten Melodie berechnet, so lässt sich aus dem Quotienten von Summe und Gesamtzahl der Elemente der Datenbank ein Mittelwert errechnen. Ein reiner Zufallstreffer in der Suche erreicht im Durchschnitt genau diesen Mittelwert.

Der durchschnittliche Score ist folglich auch ein relatives Maß. Er ist abhängig von der gesummten Melodie, allen Melodien in der Datenbank und den Parametern von allen Modulen in QbH System.

Wie beim „Signal to Noise Ratio“ gilt: Je weiter sich der Score einer Anfrage vom Durchschnitts-Score distanziert, desto signifikanter ist das Suchergebnis zu werten.

### 9.3 Timestretching

In einem Vorverarbeitungsschritt wird die gesummte Melodie mit den Faktoren 1, 2, 3 und 4 gestreckt. Durch das Strecken mit Skalierungsfaktoren 1 und 4 wird erreicht, dass auch halb und doppelt so schnell gesungene Melodien erfasst werden. Der Faktor 3 fängt potentielle Quantisierungsfehler bei der Rhythmuserkennung von stark punktierten Rhythmen ab. Da die Streckung um Faktor 3 und 4 weniger wahrscheinlich ist, werden dort Scores um den Faktor 0.5 verringert.

Jedes Lied in der Datenbank wird mit der gesummten Melodie viermal in unterschiedlicher Skalierung und Gewichtung verglichen.

### 9.4 Ähnlichkeitsberechnung

Das Modul zur Ähnlichkeitsberechnung vergleicht eine gestreckte Gesangsmelodie mit einer Melodie aus der Midi-Datenbank. Für eine Datenbank von  $n$  Liedern, wird aufgrund des Timestretchings  $N \cdot 4$  mal ein Score berechnet.

Ein Algorithmus vergleicht eine zweidimensionale Noten-Matrix, die aus den Midi Daten erzeugt wurde mit den einzeln gewichteten,  $k$  wahrscheinlichsten Noten aus der Melodieextraktion. Die Funktion liefert einen Score, einen Startzeitpunkt und einen Transponierungsfaktor zurück.

Bei der Ähnlichkeitsberechnung ist folgendes zu berücksichtigen:

- Die gesummte Melodie ist an einer beliebigen Stelle in der Melodie der Midi Datei zu finden (Verschiebung in der Zeit; X-Achse)
- Die gesummte Melodie kann in einer beliebigen Tonart sein (Verschiebung in der Y-Achse)
- Die gesummte Melodie ist meist nur ein kurzer Ausschnitt der Midi Melodie in der Datenbank
- Die gesummte Melodie enthält fast immer Fehler und kann unvollständig sein
- Die Melodie in der Datenbank kann Fehler enthalten
- In der gesummten Melodie werden pro Takt  $k$  gewichtete, verschiedene Noten gegeben
- Die Melodie in der Midi-Datenbank kann polyphon sein. Es können mehrere Noten gleichzeitig gespielt werden.
- Die gesuchte Melodie ist häufig am Anfang von Liedern in der Datenbank zu finden
- Technisch bedingte Frequenzverdopplungs- und Frequenzhalbierungsfehler sollen beim Matching weniger bestraft werden
- „Leicht falsch gesungene Noten“ sollen toleriert werden
- Zu einem Zeitpunkt können keine oder beliebig viele Noten in beliebiger Lautstärke spielen

Der relative Score kann mit verschiedenen Matching Methoden errechnet werden. Unter Matching versteht man die Korrelation zusammengehöriger Passpunkte.

**Beim Vergleich von zwei polyphonen Melodien, wird der Punkt gesucht, an dem zwei zweidimensionale Notenmatrizen so gegeneinander verschoben sind, dass sie einander am ähnlichsten sind. Die Notenmatrizen können sowohl in X-Richtung (Zeitpunkt), als auch in Y-Richtung (Tonhöhe) verschoben werden.**

Die Verschiebung in der Tönhöhe wird in der Musik als Transponieren von Tonarten bezeichnet. Die Notenwerte ändern sich dabei, die Halbtonschritte der Intervalle zueinander bleiben dabei erhalten. Zwei Melodien, die eine identische Abfolge von Intervallen und identische Rhythmen haben, sind unabhängig von ihrer Tonart für das QbH System als „identisch“ zu sehen. Diese Anforderung ergibt sich daraus, dass ein Großteil der Menschen Tonhöhen nur relativ unterscheiden kann. Sie können vorgespielte Töne nach der Tonhöhe ordnen und Intervallfolgen rhythmisiert wiedergeben. Die Wahl der Tonhöhe beim Singen ergibt sich meist willkürlich und ist stark abhängig von der Physiognomie des Vocaltrakts eines Sängers.

**Ein QbH System muss Melodien unabhängig von ihrer Tonhöhe vergleichen können.**

Bei einer Suchanfrage kann nicht davon ausgegangen werden, dass die ersten Noten im Gesungenen auch den ersten Noten in den Melodien in der Datenbank entsprechen. In der Praxis singen die Benutzer meist die Melodie eines Refrains. Der Refrain kann sowohl am Anfang als auch in der Mitte eines Liedes zu finden sein. Der Zeitpunkt des Refrains ist somit undefiniert.

**Ein QbH System muss Melodien unabhängig vom Zeitpunkt vergleichen können.**

## **9.5 Matchingalgorithmen**

Ein Matchingalgorithmus wird verwendet, um den Zeitpunkt und die Tonhöhenverschiebung zwischen zwei Notenmatrizen zu finden, bei denen sie einander maximal ähnlich sind. Darüber hinaus berechnet er einen Score, der als Ähnlichkeitsmaß für spätere Berechnungen dient.

Für „Hummel“ wurden 3 Matchingalgorithmen implementiert. Sie unterscheiden sich in Komplexität, Implementierungsaufwand und Erkennungsrate:

- Binäres Matching
- Matching mit Ableitungsfunktion
- Matching mit gewichteter Distanzfunktion

## 9.6 Binäres Matching

### 9.6.1 Abstrakt

Das binäre Matching ist eine einfache Variante, um zwei polyphone Melodien miteinander zu vergleichen.

### 9.6.2 Funktionsweise

Zunächst wird die Polyphonie zu jedem Zeitpunkt in den Melodien bestimmt. Es wird gezählt, wie viele Noten zu jedem Zeitpunkt gleichzeitig spielen. Spielen zu einem Zeitpunkt sehr viele Noten gleichzeitig, so wird später mit geringerer Gewichtung bestraft, als bei nur einer Note. Die Polyphonie-Werte werden in einem Array für die spätere Weiterverarbeitung und Optimierung gespeichert.

Die Werte X und Y sind unbekannt. Nun werden die beiden zweidimensionalen Matrizen punktweise in X und Y Richtung so gegeneinander verschoben, dass der Score maximal ist. Der Vergleich von zwei Liedern liefert also einen Score, einen X-Wert und einen Y-Wert zurück. Der maximal-Score zwischen den beiden Liedern ist als Maß dafür zu sehen, wie ähnlich die gesummte Melodie an der zu ihr ähnlichsten Stelle in der Melodie aus der Datenbank ist. Der X-Wert beschreibt den Zeitpunkt in der Melodie in der Datenbank und der Y-Wert beschreibt die ideale Transponierung.

Zur Erstellung eines Scores zur Verschiebung X/Y wird der Score zunächst auf 0 gesetzt. Dann wird die erste Matrix mit der Verschiebung um X/Y punktweise mit der zweiten verglichen. Wenn zwei Noten übereinander liegen, wird der Score „belohnt“. Existiert an der Stelle nur eine von zwei Noten, wird der Score mit einem geringeren Wert „bestraft“. Existieren an der zu untersuchenden Stelle in beiden Melodien keine Noten, bleibt der Score gleich. Je größer der Score ist, desto mehr Noten überlappen sich.

An einer Stelle bei der Verschiebung um X/Y ist der Score maximal. Dieser maximal- Score, dessen Zeitpunkt und dessen ideale Transponierung werden von der Funktion zurückgeliefert und für die Weiterverarbeitung verwendet.

### 9.6.3 Komplexität

Da die naive Implementierung mit zwei sich gegeneinander verschiebenden zweidimensionalen Matrizen eine quartische Komplexität hat und zu einem Zeitpunkt nur eine geringe Anzahl von Noten gleichzeitig spielen, sollte mindestens eine der beiden Melodien als Liste implementiert sein. Die Liste speichert zu jedem Zeitpunkt alle gespielten Noten. Dadurch kann die Komplexitätsklasse auf  $O(l \cdot n^3)$  verringert werden, wobei l der durchschnittlichen Anzahl von gleichzeitig gespielten Noten in der Listenmelodie entspricht.

Durch minimum bounding box Verfahren lässt sich der Suchraum noch zusätzlich verkleinern.

## 9.6.4 Schwächen der Methodik

Der Algorithmus belohnt oder bestraft, unabhängig von der Distanz und der musikalischen Ähnlichkeit von Noten. So unterscheidet er beispielsweise nicht, ob sich zwei Noten nur um einen Halbton oder um eine große Distanz unterscheiden.

Aufgrund dessen und der hohen Komplexität wird der Algorithmus in der aktuellen Version von „Hummel“ nicht mehr verwendet.

## 9.7 Matching mit Ableitungsfunktion

### 9.7.1 Abstrakt

Durch das Verwenden der Ableitungsfunktion kann die Komplexitätsklasse des Matchings auf  $o(n^2)$  verringert werden.

### 9.7.2 Funktionsweise

Die Problematik der hohen Komplexität für das im vorangehenden vorgestellte Binäre Matching beruht darin, dass zwei, um beliebige Werte in  $X$  und  $Y$  Richtung verschobene Matrizen verglichen werden.

Wenn man voraussetzt, dass jede der beiden Melodiematrizen zu jedem Zeitpunkt genau eine Note enthält, kann man die Matrize als lineare Zeitreihe darstellen. Der Wert  $Y$  zum Zeitpunkt  $X$  entspricht dann der Tonhöhe der Note. So müssen nur noch zwei, um die Zeit  $X$  und die Transpositionskonstante  $Y$  verschobene lineare Zeitreihen verglichen werden.

Sei  $g$  eine beliebige Funktion und  $Y$  konstant, dann gilt:

$$f'(C) = 0$$
$$f'(g + Y) = f'(g)$$

Veranschaulicht bedeutet das, dass zwei identische, nur in der  $Y$ -Achse verschobene Funktionen die selbe Ableitung haben.

Diese Eigenschaft der Ableitungsfunktion kann verwendet werden, um die unbekannte Transpositionskonstante  $Y$  los zu werden. Anstatt den Ähnlichkeitsvergleich auf den beiden, um die Zeit  $X$  und die Konstante  $Y$  verschobenen Zeitreihen durchzuführen, wird der Ähnlichkeitsvergleich zwischen den Ableitungen der um die unbekannte  $X$  verschobenen Zeitreihen berechnet. Durch das Verwenden des Ableitungstricks kann so die Komplexitätsklasse des Matchings auf  $o(n^2)$  verringert werden.

Im ersten Schritt werden die Lücken in den Melodien, an denen keine Note spielt, mit dem Wert der letzten gespielten Note aufgefüllt. Es kann immer nur genau eine Note pro Zeitpunkt spielen. Anschließend werden die Funktionen der beiden Melodien differenziert.

Zur Berechnung eines Scores mit der Verschiebung  $X$ , wird der Score zunächst auf 0 gesetzt. Dann wird die erste abgeleitete Zeitreihe mit der Verschiebung um  $X$  punktweise mit der zweiten verglichen. Wenn die Werte nahe beieinander liegen, wird der Score um einen bestimmten Wert in Abhängigkeit von der Distanz erhöht. Je näher die Distanz, desto höher ist der Wert.

An einer Stelle bei der Verschiebung um  $X$ , ist der Score maximal. Dieser maximal-Score und der Zeitpunkt  $X$  werden von der Funktion zurückgeliefert und für die Weiterverarbeitung verwendet.

### 9.7.3 Komplexität

Die Komplexitätsklasse für das Vergleichen von zwei um die Unbekannte  $X$  verschobenen Zeitreihen ist  $O(n^2)$ .

### 9.7.4 Schwächen der Methodik

In der Praxis hat sich gezeigt, dass sich das Matching mit Ableitungsfunktionen für den Vergleich von Melodien nicht gut eignet.

- Ein einzelner Fehler in der Ableitungsfunktion hat, musikalisch betrachtet, gravierende Auswirkungen auf die Melodik.
- Das Ableiten verzerrt die Semantik vom Ähnlichkeitsmaß Score stark. Nach dem Ableiten wird beispielsweise ein konstant gehaltener Ton und eine Tonleiter als ähnlich gewertet, da deren Ableitungen sich nur um  $\pm 1$  unterscheiden. Tieffrequente Differenzen zwischen den zu vergleichenden Sequenzen gehen in der Bewertungsfunktion verloren.
- Die Qualität der zu vergleichenden Daten ist zu unsauber für die Methodik.
- In der Praxis enthalten die Notensequenzen viele „Löcher“, an denen keine Note spielt. Die Werte in der Ableitungsfunktion sind hier undefiniert.
- Die Methode kann kein polyphones Material vergleichen. Sowohl die  $k$  möglichen gewichteten Grundfrequenzen als auch die Midi Sequenzen sind oft polyphon.

Da das Matching mit der Ableitungsfunktion, verglichen mit den beiden anderen Verfahren, die schlechtesten Ergebnisse liefert, wird der Algorithmus in „Hummel“ nicht mehr verwendet.

Möglicherweise führt die Verwendung eines steileren Hochpassfilters bessere Ergebnisse als die Verwendung des Differenzgliedes (Ableitung), das über das gesamte Spektrum eine Flankensteilheit von 6dB pro Oktave aufweist. So lässt sich zumindest die Verzerrung der Semantik des Ähnlichkeitsmaßes auf einen bestimmten Frequenzbereich beschränken.

## **9.8 Matching mit gewichteter Distanzfunktion**

### **9.8.1 Abstrakt**

Das Matching mit Distanzfunktion liefert von den drei Algorithmen die besten Ergebnisse. Die Wahl der Distanztabelle hängt von den Fähigkeiten eines Sängers und dem verwendeten Grundfrequenzalgorithmus ab.

### **9.8.2 Funktionsweise**

Zuerst wird die Polyphonie zu jedem Zeitpunkt in den Melodien bestimmt. Es wird gezählt, wie viele Noten zu jedem Zeitpunkt gleichzeitig spielen. Spielen zu einem Zeitpunkt sehr viele Noten gleichzeitig, so wird später mit geringerer Gewichtung bestraft, als bei nur einer Note. Die Polyphonie-Werte werden in einem Array gespeichert.

Es werden die beiden zweidimensionalen Matrizen punktweise in X und Y Richtung so gegeneinander verschoben, dass der Score maximal ist. Der Vergleich von zwei Liedern liefert einen Score, einen X-Wert und einen Y-Wert. Der X-Wert beschreibt den Zeitpunkt in der Melodie in der Datenbank und der Y-Wert beschreibt die ideale Transponierung.

Der Score wird zunächst auf 0 gesetzt. Dann werden die Tonhöhen in Abhängigkeit von der Verschiebung um X/Y verglichen. Eine Distanztabelle ordnet jeder Notendifferenz einen Wert zu. Existiert zum gegebenen Zeitpunkt eine Midi-Note, so wird der Score um den Wert der Distanztabelle, gewichtet mit dem Konfidenzwert aus der Grundfrequenzextraktion, erhöht. Existiert in der Midi-Melodie keine Note, so wird der Score mit einem geringeren konstanten Wert, gewichtet mit dem Konfidenzwert aus der Grundfrequenzextraktion, erniedrigt.

Bei der Suche nach Kinderliedern erhalten Melodien, die in den ersten 20 Sekunden erkannt werden, einen kleinen Bonus. In der Praxis konnte nämlich bei diesem Material häufig beobachtet werden, dass die Probanden nicht den Refrain, sondern den Beginn des Liedes gesungen haben.

An einer Stelle, bei der Verschiebung um X/Y, ist der Score maximal. Dieser maximal- Score, dessen Zeitpunkt und dessen ideale Transponierung werden von der Funktion zurückgeliefert und für die Weiterverarbeitung verwendet.

### 9.8.3 Distanztabelle

Die Distanztabelle ermöglicht es, die Fehler in Abhängigkeit von der Notendifferenz zu werten. In der Musik ist der Aufbau dieser Tabelle nicht linear, sondern ergibt sich aus den Ähnlichkeiten von ganzzahligen Frequenzverhältnissen zwischen Intervallen. So sind sich beispielsweise zwei Töne mit einer Distanz von 12 Noten (Oktave) und dem Frequenzverhältnis 1:2 ähnlicher als zwei Töne mit einer Distanz von 6 Noten (Tritonus) und dem Frequenzverhältnis 729:512.

Die Praxis hat gezeigt, dass Menschen die Grundfrequenz häufig nur mit einer Genauigkeit von 15% (+-2 Halbtöne) treffen. Der Algorithmus muss also tolerant gegenüber „leicht falsch“ gesungenen Noten sein. Der Zielfrequenz nahe gelegene Noten werden von der Distanztabelle mit leicht geringerer Gewichtung toleriert.

**Gegenüber dem binären Matching, das exakt gesungene Noten erfordert, konnte die Erkennungsqualität durch das Anwenden einer Distanztabelle um 20% gesteigert werden.**

Distanzfunktion Gewichtung in Abhängigkeit von der Halbtondifferenz	Kommentar	EQ Lied 1 Sänger 1 männl.	EQ Lied 2 Sänger 2 weibl.	EQ Lied 3 Sänger 3 männl.	EQ Lied 4 Sänger 4 weibl.
16,4,1,0,0,0,0,0,0,0,0,0,0,4,0,0,0,0,0,0,0,0,0,0	Toleriert Oktavfehler – für gute Sänger geeignet	19	21	18	19
16,0	Binäres Matching – „hit or miss“	19	20	17	17
16,12,4,0,0,0,0,0,0,0,0,0,0,8,6,0,0,0,0,0,0,0,0,0		21	23	19	19
16,14,12,6,0,0,0,0,0,0,0,0,0,8,7,6,3,0,0,0,0,0,0,0,0	Tolerant mit Oktavfehlern - für schlechte Sänger geeignet	23	23	20	20
16,14,12,10,8,6,0,0,0,0,0,0,5,8,5,0,0,0,0,0,0,0,0,0	Zu tolerant - zu viele ähnliche Lieder werden gefunden	7	9	6	7
16,14,12,6,0,0,0,0,0,0,0,2,4,8,4,2,0,0,0,0,0,0,0,0,0	Toleriert Oktavfehler - für schlechte Sänger geeignet	22	24	19	20
16,8,4,0,0,0,0,0,0,0,2,4,8,4,2,0,0,0,0,0,0,0,0,0	Toleriert Oktavfehler - für gute Sänger geeignet	15	17	12	13
16,16,16,0,0,0,0,0,0,0,0,8,8,0,0,0,0,0,0,0,0,0,0,0		22	24	19	19
16,16,16,8,0,0,0,0,0,0,12,12,12,0,0,0,0,0,0,0,0,0,0	Zu tolerant - zu viele ähnliche Lieder werden gefunden	10	12	7	8
8,0,0,0,0,0,0,0,0,0,0,7,0,0,0,0,0,0,0,0,0,0,0,0	Exaktes Matching mit Oktavfehlern	7	9	4	5
16,14,12,6,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0		17	19	14	14
16,14,12,6,0,0,0,0,0,0,0,4,8,4,0,0,0,0,0,0,0,0,0,0	Toleriert Oktavfehler - für schlechte Sänger geeignet	21	22	18	18
16,14,12,6,0,0,0,0,0,0,0,7,8,7,0,0,0,0,0,0,0,0,0,0	Toleriert Oktavfehler - für schlechte Sänger geeignet	23	25	20	22
16,14,12,6,0,0,0,0,0,0,0,3,4,3,0,0,0,0,0,0,0,0,0,0	Toleriert Oktavfehler - für schlechte Sänger geeignet	19	21	16	16
16,14,12,10,6,0,0,0,0,0,0,7,8,7,0,0,0,0,0,0,0,0,0,0	Zu tolerant - zu viele ähnliche Lieder werden gefunden	6	8	4	6
16,12,8,4,0,0,0,0,0,0,0,4,8,4,0,0,0,0,0,0,0,0,0,0	Toleriert Oktavfehler - für mittlere Sänger geeignet	21	22	18	20

Tabelle 9.1: Vergleich von verschiedenen Distanzfunktionen

Die „Erkennungsqualität“ EQ ist hier definiert als Quotient aus dem erreichten Score und dem durchschnittlichen Score aus allen Liedern in der Datenbank.

Wie beim „Signal to Noise Ratio“ gilt: Je weiter sich der Wert von 1 distanziert, desto besser ist das Suchergebnis. Für die Messung wurde eine Datenbank von 100 Kinderliedern verwendet. Zur Grundfrequenzextraktion diente der AKF+COMB Algorithmus mit  $k=5$ .

Bei fast allen Distanztabelle wurde die gesungene Melodie an erster Stelle gefunden. Eine Evaluierung nach Ranking war daher nicht sinnvoll und würde eine größere Datenbank und eine größere Zahl an Versuchen erfordern. Dies war aufgrund der beschränkten Rechenkapazität und des hohen Zeitaufwands im Rahmen der Diplomarbeit nicht möglich.

### **Die Wahl der Distanztabelle hängt von den Fähigkeiten eines Sängers und dem verwendeten Grundfrequenzalgorithmus ab.**

Ein guter Sänger trifft die Grundfrequenz einer Note mit einer höheren Genauigkeit als ein schlechter. Die Distanztabelle für schlechte Sänger enthalten daher auch noch bei den Distanzen  $-1$ ,  $-2$  und  $-3$  eine positive Gewichtung.

Einen Extremfall stellt das bereits vorgestellte binäre Matching dar. Hier werden nur exakt richtige Noten toleriert. Da der verwendete AKF+COMB Algorithmus um  $F_0$  herum für  $k>1$  relativ breit streut, wurden auch hier noch gute Ergebnisse erzielt.

Die verschiedenen Grundfrequenzextraktionsalgorithmen sind unterschiedlich anfällig gegenüber Frequenzvervielfachungs- und Frequenzteilungsfehlern und erfordern eine individuelle Anpassung der Distanzfunktion.

Beim AKF+COMB treten in der Praxis manchmal Frequenzverdopplungsfehler auf. Deshalb berücksichtigen manche der oben gezeigten Distanzfunktionen Ähnlichkeiten um  $d=12$  (Oktave).

Die klassische Autokorrelation ist anfällig gegenüber Frequenzteilungsfehlern, da oft Zyklizitäten mit der 2-fachen, 3-fachen und X-fachen Wellenlänge erkannt werden.

Cepstrumbasierte Grundfrequenzextraktionsalgorithmen sind anfällig gegenüber Frequenzvervielfachungsfehlern. Hier werden aufgrund der Pegelanhebung des ersten Formanten um 200Hz manchmal Zyklizitäten mit der 2-fachen, 3-fachen und X-fachen Frequenz gefunden.

### **9.8.4 Komplexität**

Da die naive Implementierung mit zwei sich gegeneinander verschiebenden zweidimensionalen Matrizen eine quartische Komplexität hat und zu einem Zeitpunkt nur eine geringe Anzahl von Noten gleichzeitig spielen, sollte mindestens eine der beiden Melodien als Liste implementiert sein. Die Liste speichert zu jedem Zeitpunkt alle gespielten Noten. Dadurch kann die Komplexitätsklasse auf  $O(l \cdot n^3)$  verringert werden, wobei  $l$  der durchschnittlichen Anzahl von gleichzeitig gespielten Noten in der Listenmelodie entspricht.

Durch minimum bounding box Verfahren lässt sich der Suchraum möglicherweise noch zusätzlich verkleinern.

## 9.8.5 Schwächen der Methodik

Die hohe Komplexität macht die Ähnlichkeitsberechnung sehr teuer. Der Algorithmus ist nur bedingt robust gegen fehlende oder überschüssige Elemente in den Melodien.

## 9.9 Vergleich der Matchingalgorithmen

Jeder der untersuchten Algorithmen bietet sowohl Vor- als auch Nachteile. Eine effiziente Lösung mit guten Suchergebnissen konnte für den Vergleich von Melodien nicht gefunden werden. Deshalb bleibt das Matching der Flaschenhals bei der Ähnlichkeitssuche.

Das Matching mit gewichteter Distanzfunktion liefert in „Hummel“ die besten Ergebnisse.

### 9.9.1 Binäres Matching

Vorteile:

- Einfach zu implementieren
- Kann polyphones Material vergleichen

Nachteile:

- Performance
- Intolerant gegenüber geringen Grundfrequenzfehlern

Ausblick und Verbesserungsmöglichkeiten

- *Durch die Verwendung von Dynamic Time Warping könnte der Algorithmus robuster gegenüber von Rhythmusfehlern werden. [Ber 94]*

## 9.9.2 Matching mit Ableitungsfunktion

Vorteile:

- Performance

Nachteile:

- Kann nur monophones Material vergleichen
- Das Ableiten verzerrt die Semantik vom Ähnlichkeitsmaß
- Schlechte Suchergebnisse
- Wenig geeignet für den Vergleich von Melodien

Ausblick und Verbesserungsmöglichkeiten

- Die Verwendung eines steileren Hochpassfilters anstatt der Ableitungsfunktion könnte zu einer Verbesserung der Suchergebnisse führen. Die Verzerrung der Semantik des Ähnlichkeitsmaßes würde so auf einen kleineren Frequenzbereich beschränkt werden.
- Ein Algorithmus, der automatisiert den Grundton aus Akkorden extrahiert, kann den Algorithmus auch für die Analyse von polyphonem Material brauchbar machen.
- Durch die Verwendung von Dynamic Time Warping könnte der Algorithmus robuster gegenüber von Rhythmusfehlern werden.

## 9.9.3 Matching mit gewichteter Distanzfunktion

Vorteile:

- Kann polyphones Material vergleichen
- Gute Suchergebnisse
- Distanzfunktion kann Schwächen im Grundfrequenzextraktionsalgorithmus mit berücksichtigen

Nachteile:

- Performance

Ausblick und Verbesserungsmöglichkeiten

- *Durch die Verwendung von Dynamic Time Warping könnte der Algorithmus robuster gegenüber von Rhythmusfehlern werden. [Yun 02]*

## 10 Sortierung und Ranking

### 10.1 Abstrakt

Im Ranking Modul werden die Lieder nach Priorität gefiltert und sortiert. Es wird eine Liste, der zur gesummtten Melodie ähnlichsten Lieder, ausgegeben.

### 10.2 Funktionsweise

Im Matching Modul wurde die gesummtte Melodie mit allen Liedern aus der Datenbank verglichen. Jeder Vergleich lieferte einen Score zurück. Je höher der Score ist, desto ähnlicher sind sich die beiden Melodien.

Der Score ist ein relatives Maß und liefert nur die Aussage darüber, dass das Gesummtte der Midi Datei A ähnlicher ist, als der Midi Datei B.

Aus dem Quotienten der Gesamtsumme aller Scores und der Anzahl der Lieder in der Datenbank kann man einen durchschnittlichen Score errechnen. Dieser Mittelwert variiert von Anfrage zu Anfrage und hängt stark von den Audiodaten der gesummtten Melodie ab. Der Score einer Anfrage lässt sich in Relation zu dem Mittelwert aller Anfragen setzen. Je weiter dieser vom Mittelwert abweicht, desto mehr unterscheidet sich die Anfrage von einem Zufallstreffer. Diese Abweichung vom Mittelwert kann wie das „Signal to Noise Ratio“ (SNR) verstanden werden. Bei einer hohen positiven Abweichung vom Mittelwert ist sich das System mit seiner Aussage sehr sicher. Ein Wert von 1 entspricht einem reinem Zufallstreffer. Lieder, deren SNR unterhalb von 1 liegt, schließt das System als potentielle Kandidaten aus.

Falls keine feste Anzahl von besten Liedern zurückgegeben werden soll, kann das SNR als Schwellwert benutzt werden. So kann es sinnvoll sein, alle Lieder mit einem SNR größer als 2 zurück zu liefern.

Die Lieder werden dann anhand ihres Scores sortiert. Die k besten Treffer von der Gesamtmenge aus n Kandidaten werden zurückgeliefert.

Ist k konstant und  $k \ll n$ :

Sind nur die k besten Treffer aus n Liedern interessant, kann die Menge von k+1 bis n unsortiert bleiben und der Algorithmus dementsprechend optimiert werden.

Ist k variabel und wird ein SNR als Schwellwert benutzt:

Die Liste aus allen Scores kann vorgefiltert werden. Nur die Kandidaten, deren SNR über dem Schwellwert liegt, müssen sortiert werden.

Nach der Sortierung bekommt der Benutzer eine Liste von möglichen Titeln ausgegeben. Das Lied, das das QbH System für am ähnlichsten zur gesummtten Melodie hält, führt diese Liste an.

## 11 „Hummel“ - eine praktische Umsetzung eines QbH Systems

Eine wichtige Messgröße für ein Query-by-Humming System ist die Erkennungsrate. Ein gutes System findet mit großer Wahrscheinlichkeit das „richtige“ Lied zur gesumzten Melodie.

Ursprünglich sollten im Umfang dieser Diplomarbeit lediglich einzelne Module zur Unterstützung von QbH Systemen entwickelt werden. Einzelne Module lassen sich zwar separat implementieren, aber um deren Leistungsfähigkeit hinsichtlich der Erkennungsrate sinnvoll testen zu können, ist es nötig, das komplette System zu betrachten.

Daher wurde mit hohem Aufwand ein bis auf das Sampling Modul komplettes, experimentelles QbH System implementiert. Aufgrund der hohen Ansprüche an Performance und Flexibilität wurde C++ als Programmiersprache gewählt. Als Entwicklungsumgebung diente „Visual Studio 2008“. „Hummel“ ist eine Win32 Anwendung und umfasst die Gesangsmelodieanalyse, die Rhythmusanalyse, eine Musikdatenbank im Midi Format, den Import von verschiedenen Dateiformaten, den Export in verschiedene Dateiformate, die Visualisierung der Analysedaten, die Ähnlichkeitssuche auf der Datenbank und die Sortierung nach Ranking.

Die Quelltexte zum System umfassen insgesamt über 5000 Zeilen an Code.

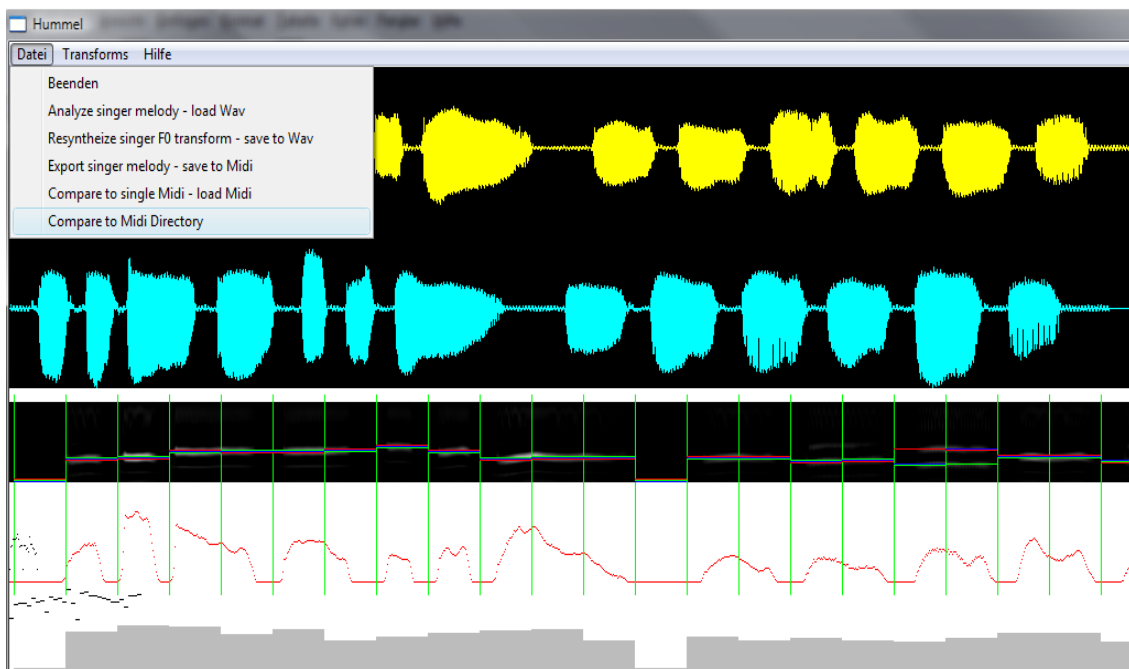


Abbildung 11.1: Screenshot des experimentellen QbH Systems „Hummel“

## **11.1 Probleme bei der Implementierung von QbH Systemen in der Praxis**

Das Verändern nur eines einzigen Parameters in den Algorithmen von QbH Systemen kann das Gesamtverhalten des Systems gravierend verändern. Zusammen mit der Gegebenheit, dass alle Ähnlichkeitsmaße relativ zueinander stehen, wird die Evaluierung und das Feintuning von Parametern sehr schwierig.

Es kann oft nur grob untersucht werden, ob die Veränderung eines Parameters eine Verbesserung oder Verschlechterung mit sich bringt. Es aber auch möglich das Verhalten zu beobachten, wenn einzelne Module weggelassen oder gegen andere ersetzt werden.

Erschwerend für die Evaluierung kommt hinzu, dass eine Suchanfrage auf einer Datenbank mit 100 Liedern in „Hummel“ trotz hardwarenaher Implementierung mehrere Minuten dauert. Eine effiziente Lösung des Matchingproblems mit hoher Erkennungsrate konnte im Rahmen dieser Diplomarbeit nicht gefunden werden.

Bei den Recodings zeigte sich, dass die Versuchspersonen die Melodien eines Liedes häufig korrekt singen konnten. Sie hatten jedoch oft Probleme, die gleiche Melodie zu summen. Es waren oft mehrere Aufnahmen nötig, bis das Klangmaterial den Qualitätskriterien entsprach.

## 12 Zusammenfassung

Im Umfang dieser Diplomarbeit wurden viele neue Ideen implementiert. Einige Ansätze konnten signifikante Vorteile gegenüber existierenden Methoden erzielen.

Für die Vorverarbeitung wurde ein Filter zur spektralen Einebnung entwickelt, das die Notenerkennungsrate für gesumstes Audiomaterial bei männlichen Sängern um 39% steigert.

Für die Grundfrequenzsuche wurde die AKF+COMB Transformation zur Erzeugung von „Grundfrequenz Spektrogrammen“ entwickelt. Der Algorithmus erreicht in Kombination mit dem Filter aus der Vorverarbeitung eine Notenerkennungsrate von 98%. Das Grundfrequenzerkennungsproblem von Query-by-Humming Anwendungen konnte gelöst werden. Der Anwendungsbereich der Transformation ist nicht nur auf QbH Anwendungen beschränkt. Er könnte auch wertvolle Erkenntnisse bei der Analyse von Aktienkursen, bei der Auswertung von Seismogrammen und bei der Analyse von Spektren in der digitalen Funktechnologie liefern. Die Transformation ist eine allgemeine Methode zur Visualisierung und Suche von periodischen, auch nicht sinusförmigen Zyklizitäten im Spektrum.

Es konnte ein Algorithmus auf Basis der Autokorrelation gefunden werden, der die automatisierte Takterkennung bei synthetischen Audiomaterial effizient löst.

Für die automatisierte Melodieerkennung wurde ein Algorithmus vorgestellt, der kontinuierliche Grundfrequenzverläufe in diskrete Notenwerte quantisiert.

Für den Melodievergleich konnte eine Distanztabelle gefunden werden, welche die Erkennungsrate steigert. Die Distanztabelle hilft, dass auch falsch gesungene Noten für die Suche verwendet werden können.

Es wurde ein experimentelles QbH System entwickelt, das den Titel von Melodien automatisiert erkennen kann.

Das System umfasst:

- Den Im- und Export von WAV Dateien
- Eine große Anzahl von Testsignalen
- Die Vorverarbeitungsroutinen
- Das Erkennen von Grundfrequenzverläufen
- Die Rhythmusextraktion
- Die Melodieerkennung
- Den Im- und Export von Midi Dateien
- Die Resynthese von Spektrogrammen
- Eine Midi Datenbank
- Mehrere Matchingalgorithmen zur Ähnlichkeitssuche

## 13 Ausblick

Um das QbH System zu einem marktreifen Produkt zu entwickeln, müssen in einigen Modulen noch weitere Verbesserungen vorgenommen werden.

Die Robustheit der automatisierten Takterkennung gegenüber von gesungenem Material sollte verbessert werden. Dazu ist eine alternative Extraktionsmethode der Kurzzeitenergie nötig. Das AKF+COMB Spektrogramm hat nur bei synthetischen Klängen gute Ergebnisse erzielt.

Es wird ein Matching Algorithmus benötigt, der den Melodievergleich effizient löst und zugleich eine hohe Erkennungsrate hat. Dadurch wäre die Suche auf großen Datenbanken und eine genauere Evaluierung der Leistungsfähigkeit des QbH Systems möglich.

Für den automatisierten Import von polyphonen Midi Dateien aus der Datenbank wäre eine Akkordauflösung auf den Grundton hilfreich. Dadurch könnte die zweidimensionale Midi-Matrix in eine eindimensionale Zeitreihe aus Noten gewandelt werden. Die Komplexität des Matchings lässt sich dadurch möglicherweise verringern.

Bis QbH Systeme erfolgreich kommerziell eingesetzt werden können, sind noch einige Probleme zu lösen:

*„...it is a very challenging task.“* [Lie 01], (Microsoft research)

## 14 Glossar

<b>Begriff</b>	<b>Bedeutung</b>
Abtastrate	Samplingrate, Samplerate Frequenz, mit der ein Signal pro Zeitintervall abgetastet wird
Akkord	Das gleichzeitige Erklingen mehrerer unterschiedlicher Töne in der Musik
AKF	Autokorrelationsfunktion
Array	k-dimensionale Datenstruktur aus der Informatik
Bandpass	Filter, das in einem Signalweg nur die Signale eines bestimmten Frequenzbandes durchlässt und die restlichen Frequenzbereiche sperrt bzw. deutlich abschwächt
BPM	Beats per minute, Taktgeschwindigkeit
dB	Dezibel, logarithmische Pegel­einheit
DFT	Diskrete Fouriertransformation
Distanzfunktion	Funktion, die je zwei Elementen eines Raums einen Wert zuordnet, der als Abstand der beiden Elemente voneinander aufgefasst werden kann
DSP	Digitale Signalverarbeitung Speicherung, Übermittlung und Transformation von Information im Sinne der Informationstheorie in Form von digitalen, zeitdiskreten Signalen
Faltung	Mathematischer Operator, der für zwei Funktionen f und g eine dritte Funktion liefert
Fensterfunktion	Die Fensterfunktion legt fest, mit welcher Gewichtung die bei der Abtastung eines Signals gewonnenen Abtastwerte innerhalb eines Ausschnittes (Fenster) in nachfolgende Berechnungen eingehen
FFT	Fast Fourier Transform
Filter	Elektrische Schaltung, die bestimmte Frequenzen aus einem Signalspektrum abschwächt

<b>Begriff</b>	<b>Bedeutung</b>
FIR	Filtertyp, Finite Impulse Response, endliche Impulsantwort
Frame	Signalausschnitt bei der Blockverarbeitung
Frequenzbereich	Spektrum eines über die Zeit wechselnden Vorganges
Grundfrequenz	Grundschiwingung, Grundton, F0, fundamental frequency
GFB	Grundfrequenzbestimmung
Halbton	Intervall mit dem Frequenzverhältnis 1:1,059
Hochpass	Filter, die nur Frequenzen oberhalb ihrer Grenzfrequenz ungeschwächt passieren lassen und tiefere Frequenzen dämpfen.
Intervall	Höhenunterschied zwischen zwei Tönen
IIR	Infinite Impulse Response, unendlich lange Impulsantwort
KKF	Kreuzkorrelationsfunktion
MIDI	Musical Instruments Digital Interface
Oktave	Note mit der doppelten Grundfrequenz
QbH	query by humming
Oberton	Frequenzen, die ganzzahligen Vielfachen der Grundfrequenz entsprechen
Quantisierung	Größe in einem System, in dem sie nur diskrete Werte annehmen kann
Rauschen	Überlagerung mehrerer Schwingungen mit unterschiedlicher Amplitude und Frequenz
Resynthese	Ein ursprüngliches Signal wird aus den Einzelschwingungen wieder zusammengesetzt
Signalverarbeitung	Bearbeitungsschritte, die das Ziel haben, Informationen aus einem Signal zu extrahieren oder Informationen für die Übertragung von einer Informationsquelle zu einem Informationsverbraucher vorzubereiten.

<b>Begriff</b>	<b>Bedeutung</b>
Spektrum	Frequenzspektrum, Spektralverteilung Gesamtheit der Frequenzen, die von einem schwingenden System erzeugt werden bzw. in einem Signal enthalten sind
Spektrogramm	Darstellung des zeitlichen Verlaufes des Spektrums eines Signals
Tiefpass	Filter, die Signalanteile mit Frequenzen unterhalb ihrer Grenzfrequenz annähernd ungeschwächt passieren lassen, Anteile mit höheren Frequenzen dagegen abschwächen.
SNR	Signal-to-noise ratio
Sequencer	Software zum Editieren von Midi Dateien
WAV	Dateiformat für Audiosignale

## 15 Inhaltsverzeichnis der beigelegten CD

Verzeichnis	Inhalt
Texte	Elektronische Version der Diplomarbeit als OpenOffice Writer Dokument und PDF Datei
Vortrag	Elektronische Version des Vortrags zur Diplomarbeit als OpenOffice Writer Dokument und PDF Datei
Quelltexte	Sourcecodes zum experimentellen QbH System „Hummel“ als Visual Studio C++ 2008 Express Projekt; Beim Compilieren ist auf die korrekte Verzeichnisstruktur zu achten.
Bilder	Bilddateien zur Diplomarbeit
Projektarbeit	Elektronische Version der vorangehenden Projektarbeit mit Peter Kunath als OpenOffice Writer Dokument und PDF Datei
Midi Datenbank	Musikdatenbank mit 107 Kinderliedern im Midi Format
Rhythmuserkennung	44 Testsignale zur Rhythmuserkennung im WAV Format
Grundfrequenzerkennung	49 Testsignale zur Grundfrequenzerkennung im WAV Format

## 16 Literaturverzeichnis

- [Agr 93] R. Agrawal, C. Faloutsos, A.N. Swami: „Efficient Similarity Search In Sequence Databases“, Proceedings of the 4<sup>th</sup> Conference of Foundations of Data Organisation and Algorithms (FODO), Springer Verlag, Chicago Illinois, Seiten 69-84, 1993
- [Ber 94] D. Berndt, J. Clifford: „Using dynamic time warping to find patterns in time series“, Advances in Knowledge Discovery and Data Mining, Seiten 229-248, AAAI/MIT, 1994
- [Boc 04] Jürgen Bock: „Algorithmen zur Tonhöhenenerkennung und Vergleich verschiedener Implementierungen“, 2004.
- [Cha 99] K.-P. Chan, A. W. -C. Fu: „Efficient time series matching by wavelets“, Proceedings of the 15<sup>th</sup> International Conference on Data Engineering, Sydney, Australia, Seiten 126-133, 1999
- [Hes 83] Wolfgang Hess: „Pitch Determination of Speech Signals“, Springer-Verlag, 1983.
- [Hes 05] Wolfgang Hess: „Sprachsignalverarbeitung“, Kap 4.1, 2005.
- [Jan 01] J.-S. R. Jang, H.-R. Lee: „Hierarchical filtering method for content-based music retrieval via acoustic input“, Proceedings of the ninth ACM international conference on Multimedia, Seiten 401-410, ACM Press, 2001
- [Kor 97] F. Korn, H. V. Jagadish, C. Faloutsos: „Efficiently supporting ad hoc queries in large datasets of time sequences“, SIGMOD International Conference on Management of Data, Tucson Arizona USA, Seiten 289-300, 1997
- [Lie 01] Lie Lu: „A new approach to query by humming in music retrieval“, 2001.
- [Maz 01] D. Mazzone, R.B. Dannenberg: „Melody matching directly from audio“, 2<sup>nd</sup> Annual International Symposium on Music Information Retrieval, Bloomington Indiana USA, 2001

- [Med 91] Medan, Chazan, Yair: „Grundfrequenzbestimmung“, 1991.
- [Pop 02] I. Popivanov, R. J. Miller: „Similarity search over time series data using wavelets“, ICDE, 2002
- [Res 99] Eckhard Bernd Reschke: „Implementierung eines Algorithmus zur robusten Analyse der Sprachgrundfrequenz“, 1999.
- [Sch 68] Schroeder M.R.: „Period histogram and product spectrum: new methods for fundamental-frequency measurement, Journ. Acoust. Soc. Am. 43, 819-834
- [Tol 00] T. Tolonen, M. Karjalainen: „A computational efficient multi-pitch analysis model.“, IEEE Transactions on Speech and Audio Processing, 2000
- [Wes 04] Helge Wessels: „Audio Information Retrieval“, 2004.
- [WIK 08] Wikipedia:  
[http://de.wikipedia.org/wiki/Musical\\_Instrument\\_Digital\\_Interface](http://de.wikipedia.org/wiki/Musical_Instrument_Digital_Interface),  
11.11.2008
- [WIK 08-2] Wikipedia:  
<http://de.wikipedia.org/wiki/Autokorrelation>  
10.10.2008
- [WIL 03] Arne van Wilgen: „Tonhöhenbestimmung für Verfahren der Melodieerkennung im Standard MPEG-7“, 2003.
- [Wu 00] Y.-L. Wu, D. Agrawal, A.E. Abbadi: „A comparison of dft and dwt based similarity search in time-series databases“, Proceedings of the 9<sup>th</sup> International Conference on Information and Knowledge Management, 2000
- [Yi 00] B.-K. Yi, C. Faloutsos: „Fast time sequence indexing for arbitrary lp norms“, VLDB 2000, Proceedings of 26<sup>th</sup> International Conference on Very Large Data Bases, Cairo Egypt, 2000
- [Yun 02] Yunyue Zhu, Dennis Shasha: „Query by Humming: a Time Series Database Approach“, 2002, Seiten 1-7

- [Zhu 01] Y. Zhu, M.S. Kankenhalli, C. Xu: „Pitch tracking and melody slope matching for song retrieval“, Advances in Multimedia Information Processing – PCM 2001, Second IEEE Pacific Rim Conference on Multimedia, Beijing China, Seiten 24-26, 2001