SOMeJukeBox

ao. Univ. Prof. Mag. Dr. Dieter Merkl
Univ. Ass. DI Dr. Andreas Rauber
Institut für Softwaretechnik
Arbeitsgruppe: Information & Software Engineering
www.ifs.tuwien.ac.at
A-1040 Wien

Diplomarbeit

Automatische Analyse und Organisation von Musikarchiven

Markus Frühwirth

März 2001

Technische Universität Wien

All jenen Menschen, die mich in meiner Studienzeit mehr beeinflusst haben, als sie es sich je vorstellen werden können.

Inhaltsverzeichnis

1. Vorwort

Mit der wachsenden Anzahl verfügbarer Musikstücke werden Systeme, die ein intuitives Durchforsten großer Archive erlauben, immer wichtiger. Durch die große Datenflut von Multimediadateien und deren völlig anderen Aufbau versagen textbasierte Verfahren zur Gruppierung und Durchsuchung solcher Bestände. Die bisher oft verwendeten Ansätze zur Gruppierung von Musik bestehen in aufwändigen, nicht automatisierbaren, händischen Einteilungen in Musikgruppen (Pop, Jazz, Klassik, Rock, Dance, House, Funk, Metal,...). Nicht nur wegen fehlender exakter Klassenbeschreibungen, sondern auch aufgrund der verschiedenen Geschmacksrichtungen, sind solche Verfahren extrem subjektiv. Durch die Vielzahl der heutzutage verfügbaren Musikstücke sind solche Verfahren vor allem wegen ihres personalintensiven Einsatzes nicht mehr zeitgemäß.

Wünschenswert wäre eine Applikation, die Musikstücke nicht anhand von Metainformationen (Liedtitel, Länge,...) sondern basierend auf ihrem „Inhalt“ gruppiert. Diese Applikation sollte ohne manuelle Unterstützung Ergebnisse erzielen, die einfach zu deuten und für jedermann verständlich sind.

In dieser Arbeit wird ein Lösungsansatz präsentiert, der ohne Benutzung von Metainformationen Musikstücke mittels einer Self-Organizing Map (SOM) aufgrund ihrer Klangcharakteristik vollautomatisch gruppiert. Frequenzspektren werden verwendet um Merkmale zu extrahieren und einen klang-dynamischen Vektor zur Verarbeitung in der SOM zu generieren. In einem Zweischrittverfahren werden zuerst kurze Musiksegmente nach ihrer Ähnlichkeit gruppiert. Danach wird, abgeleitet von diesem Ergebnis der Musiksegmente, eine Gruppierung der Musikstücke vorgenommen. Durch die topologischen Eigenschaften der SOM werden ähnlich klingende Musikstücke in benachbarten Regionen auf der resultierenden Karte gefunden. Durch die einfache Präsentation des Ergebnisses – in Form einer zweidimensionalen Karte – ist eine allgemeinverständliche Interpretation und Benutzung gesichert.

Im weiteren ist die Arbeit folgendermaßen aufgebaut: Kapitel 2 gibt eine ausführlichere

Einleitung; Kapitel 3 stellt Grundbegriffe und bisherige Arbeiten auf diesem Gebiet vor; Kapitel 4 erklärt das gesamte Verfahren und Kapitel 5 stellt die Experimente vor. Kapitel 6 bietet eine Zusammenfassung und einen Ausblick auf Verbesserungsmöglichkeiten.

2. Motivation

Durch die stetig sinkenden Hardwarekosten sind Musik erzeugende und verarbeitende Geräte heute schon für eine breite Masse erschwinglich. Zur Einrichtung eines einfachen Tonstudios bedarf es weder allzu großer finanzieller Aufwendungen, noch detaillierter fachspezifischer Kenntnisse, da sachgerechte Hilfe durch elektronische Medien, wie das Internet, einfach verfügbar ist [Dör00]. Die einfache Verfügbarkeit der Musik erzeugenden Geräte, sowie die Möglichkeit durch das Internet ein großes Publikum zu erreichen, führte dazu, dass immer mehr Interpreten und Komponisten, die keinem der großen Musik verkaufenden Konzerne (BMG, EMI, Warner, Sony,...) angehören, ihre Werke selbst der Öffentlichkeit zur Verfügung stellen. Abgesehen von den in den Medien derzeit viel zitierten illegalen Kopien durch das Urheberrecht geschützten bekannter Titeln, gibt es vor allem im Internet eine Vielzahl von Musikstücken, die ohne Leistung von Gebühren abgerufen werden dürfen (z.B. AudioGalaxy.com). Im Gegensatz zu berühmten Gruppen sind diese Interpreten meist wenig bekannt und in großen Sammlungen eher unterrepräsentiert. Da solche Gruppen in großen Musikstores (Virgin, Amadeus,...) meist nicht zu finden sind, muss man um die Stilrichtung einer Gruppe herauszufinden, entweder selbst ein Stück hören, oder sich auf die Empfehlung eines Dritten verlassen. Alle diese Ansätze beruhen darauf, dass irgendwann einmal eine manuelle Einteilung der Musikstücke erfolgte. Diese Einteilung basiert meist darauf, dass eine mehr oder weniger qualifizierte Fachkraft sich jedes einzelne Stück anhört, um es in ein vorher festgelegtes Gruppenschema einzuordnen. Problematisch dabei ist, dass es kaum objektive Gruppenbeschreibungen gibt. Selbst wenn Beschreibungen dieser Gruppen existieren, sind die Musikstücke, die dieser Definition exakt entsprechen, eher in der Minderheit. Durch diesen Missstand kann es sein, dass ein und dasselbe Stück einmal in einer bestimmten Gruppe *A* und an einer anderen Stelle in einer anderen Gruppe *B* gefunden wird. Diese Gruppen können verschieden heißen, dieselben Stücke beinhalten, oder tatsächlich zwei verschiedene Gruppen sein. Ein weiteres Problem entsteht, wenn ein Stück gar nicht in das vorgegebene

Gruppenschema passt. Eine Möglichkeit wäre, es in die noch am ehesten passende Gruppe zu geben; eine andere besteht darin, eine Sammelgruppe für nicht ins Schema passende Stücke zu kreieren, und die Dritte führt zur Neuanlegung einer Gruppe. Keine dieser drei Ansätze ist ideal. Bei der ersten Möglichkeit passt das Stück nicht zu den anderen, die zweite Möglichkeit eröffnet eine Gruppe, die keinerlei Definition besitzt und deren Stücke keine Gemeinsamkeit besitzen, wobei hingegen bei der Dritten dann merkwürdige Mischformen entstehen könnten, wie etwa „Pop Rock“ und „Rock Pop“[1]. Manche Organisationen (Libro) haben diesen Missstand bemerkt und versuchen daraus zu flüchten, indem sie eine einzige große Gruppe kreieren (Pop), in die alle Stücke fallen und innerhalb der ein einfaches Sortierkriterium herrscht (alphabetisch sortiert nach Name des Interpreten). Ohne Name des Interpreten wird man allerdings auch hier nicht fündig. Ein weiterer Ansatz zur Gruppierung besteht darin Genreeinteilungen nach Vorlieben anderer Benutzer aufzubauen. Manche Online Shops (z.B. Amazon) verfolgen die Bestellungen der Benutzer und präsentieren Listen mit Stücken, die andere Benutzer auch bestellt haben[2].

Zusammenfassend kann gesagt werden, dass manuelle Genreerkennung durch den personalintensiven Einsatz und dessen subjektive Ergebnisse keine optimale Lösung zur Gruppierung von Musikstücken ist. Ohne Genrekennung allerdings bleiben Gruppen, deren Name noch völlig unbekannt ist, einem größeren Publikum verschlossen.

Es gibt schon seit einiger Zeit maschinell unterstützte Versuche Musik nach ihrem „Inhalt“ zu gruppieren, doch basieren fast jede dieser Ideen auf irgendwelchen Einschränkungen, die vor der Klassifizierung gemacht werden (Kapitel 3.1). Was fehlt ist ein System, das ohne manuellen Eingriff Musik gruppiert. Ähnliche Musikstücke sollen zu Gruppen zusammengefasst werden und ähnliche Gruppen sollen beieinander liegen.

In dieser Arbeit wird ein Verfahren vorgestellt, das nach obigen Bedingungen arbeitet. Mittels eines ausgereiften Neuronalen Netzes, der Self-Organizing Map (SOM), werden Musikstücke zu ähnlich klingenden Gruppen zusammengefasst. Die SOM benötigt außer den Eingabewerten, die direkt aus der Musik erstellt werden, keine zusätzlichen Informationen und bildet die Ergebnisse in einer einfach zu verstehenden zweidimensionalen Tabelle – der Karte – ab. Ähnlich klingende Musikstücke werden visuell in benachbarten Regionen auf der Karte dargestellt, völlig konträr klingende Musikstücke liegen weiter auseinander. Während beispielsweise Klassik im unteren linken Eck der Karte zu finden ist, wird Disco-Musik in einem gänzlich anderen Bereich, z.B. oben rechts, zu finden sein. Durch diese Einteilung

[1] AudioGalaxy.com

[2] „people who bought this CD also looked at the following titles“

können alle oben beschriebenen Probleme gelöst werden. Weder eine mühsame händische Einteilung, noch eine subjektive Geschmacksrichtung, noch unbekannte Interpreten sind hier zu berücksichtigen.

3. Formalisierung

Im folgenden Kapitel werden Grundbegriffe, die in der restlichen Arbeit verwendet werden, eingeführt. Es wird auf frühere Arbeiten auf diesem Gebiet und auf deren Forschungsschwerpunkte hingewiesen. Die Self-Organizing Map (SOM), ein unüberwacht lernendes Neuronales Netz, wird vorgestellt. Das Konzept, die Funktionsweise und der Lernalgorithmus der SOM wird erklärt. Weiters wird ein Einblick in den Aufbau von Musik und ihre Aufzeichnung am Computer gewährt. Pulse Code Modulation (PCM) und Musical Instruments Digital Interface (MIDI) wird als Datenformat, indem Musik abgespeichert sein kann, vorgestellt. Vor all dem folgt aber ein Überblick über bisherige Arbeiten zur Musikklassifikation.

3.1. Frühere Arbeiten

Abbildung 3.1 zeigt einen Überblick über das Forschungsfeld Audioverarbeitung mit Schwerpunkt auf Musik verarbeitenden Systemen. Die Grafik enthält Forschungsgebiete und deren Themenverwandtschaften (schwarz), sowie Forscher (grau), die an den jeweiligen Gebieten arbeiten. Verfahrensverwandtschaften sind durch graue Linien markiert. Die Markierung „diese Arbeit“ (grau, Mitte) sortiert das hier gebrachte Verfahren zwischen den derzeit bekannten Ansätzen ein.

Bis zum heutigen Tag existiert eine Vielzahl von Ansätzen zur inhaltsbasierten Musikklassifikation, doch kann man fast alle der Frequenzanalyse basierend auf Wold [Wol96] oder der Melodieanalyse rund um Ghias [Ghi95] zurechnen.

Allgemein kann man Audiosignale in vier Gruppen einteilen: Stille, Stimme, Geräusche und Musik. Die größten Fortschritte bei der Audiosignalverarbeitung wurden bisher auf dem Gebiet der Stimmenverarbeitung erzielt. Das Gebiet der Sprecher- und Spracherkennung (engl: *Automatic Speech Recognition (ASR)*) wird bereits von industriellen Applikationen abgedeckt (z.B. IBM ViaVoice). Für dieses Forschungsfeld ist es wichtig Stimme und andere

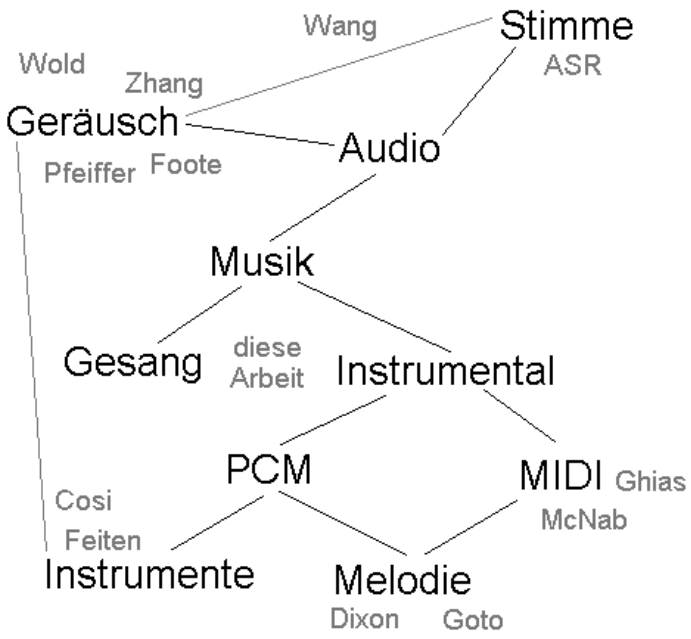

Abbildung 3.1.: Forschungsfeld Audio mit Schwerpunkt Musik

Audiosignale zu trennen [BW99, VE92]. Nach dieser Trennung kann man das übrig bleibende Geräusch (z.B. Applaus) durch Frequenzanalyse klassifizieren, um beispielsweise Pausen zwischen zwei Rednern zu finden. Jedes Audiosignal, das der Mensch wahrnimmt, wird durch das Ohr aufgefangen. Ein Hauptaugenmerk der Forschung gilt daher der Nachbildung des hörverarbeitenden Systems des Menschen. Ellis beschäftigt sich in seinen Arbeiten mit dem Entwurf solcher mathematischer Funktionen [ER95].

Wold et al. gehörten zu den ersten, die Audiosignale inhaltsbasiert analysierten [Wol96]. Ihre Arbeit konzentriert sich darauf, Merkmale wie Lautstärke, Tonhöhe, Bandbreite und Klangfarbe aus den Rohdaten zu extrahieren. Aus diesen Größen wird ein Vektor generiert, der zur Ähnlichkeitssuche benutzt wird. Die von den Autoren gegründete Firma Muscle Fish [Mus01] hat eine Applikation entwickelt, die ausgehend von dieser Arbeit in der Lage ist, Geräusche zu erkennen (z.B. Applaus, Gelächter, ...). Diese Einteilung wird erzielt, indem die zu messenden, vorher beschriebenen Vektoren, mit Vektoren in einer Referenzdatenbank verglichen werden.

Auch Foote [Fo97b] verwendet einen ähnlichen Ansatz. Hier werden nicht systemunabhängige Referenzvektoren (z.B. Applaus) verwendet, sondern spezielle Referenzvektoren vor dem eigentlichen Vergleich erstellt (z.B. männlicher Redner). Einen recht guten Überblick über dieses Forschungsfeld, ebenfalls von Foote, findet man in [Fo97a]. Im Rahmen des

MoCA Projekts [MoC98] konstruierten Pfeiffer et al. ein System [EPF96] zur automatischen Erkennung von Gewaltszenen im Fernsehen bzw. allgemeiner zur Extraktion von Informationen aus Videos. Auch dieses System vergleicht die Extrakte mit einer Datenbank und erinnert somit an den Ansatz von Wold [Wol96]. Zhang [ZK98] setzt auf demselben Ansatz auf. Nach Teilung der Audiodatei in Stimme, Stille, Musik und Geräusche erfolgt eine Feinsegmentierung, basierend auf Merkmalsextraktionen, die dann mit vorher festgelegten Vektoren (z.B. Applaus, Schrei, ...) verglichen werden. Die Vektoren werden hier durch so genannte Grundfrequenzen, die jede Quelle inne hat, gebildet. Die Grundfrequenz einer männlicher Stimme liegt etwa bei 120 Hz. All diese Arbeiten erzielen durchwegs sehr gute Ergebnisse.

Auch Ansätze zur Instrumentenerkennung [SW98] basieren auf Frequenzanalysen. Hier wird ein spezieller Filter angewandt, um Frequenzbereiche stärker zu gewichten. Cosi et al. [PCL94] benutzen in ihrer Arbeit ein Neuronales Netz um die Klangfarbe von Instrumenten zu klassifizieren. Um die Datenrate niedrig zu halten senkte man die Sampling Rate (SR) (siehe Kapitel 3.3) der Stücke um ein Vielfaches (teilweise nur 4000 Hz). Bei diesen Experimenten wurden aber nur Instrumente getrennt, und die Datensammlung enthielt nur getrennte Aufnahmen von Instrumenten. Feiten benutzte bereits eine SOM um Musik zu klassifizieren [FG94]. Er konzentrierte sich hauptsächlich auf die Klangfarbe und vernachlässigte alles andere, was ausgehend von den damals zur Verfügung stehenden Hardware Ressourcen der einzig gangbare Weg war. Genau wie bei dieser Arbeit steht am Anfang eine Auswahl von Samples (siehe Kapitel 3.3) und eine darauf folgende Transformation von dem Zeitbereich in die Frequenzebene. Danach benutzt Feiten eine mathematische Darstellung des Ohrs um das Signal anzupassen und generiert daraus den Eingabevektor für die SOM. Da Feiten aber nur einzelne Töne analysiert, fehlt die Melodiekomponente völlig.

An Ansätzen, die die Melodie berücksichtigen, arbeiten Dixon [Dix00] und Goto [MG97]. Dixon's Arbeit, die auf der Goto's beruht, konzentriert sich hauptsächlich auf die Konstruktion und Implementierung eines „Beat-Tracking" Systems. Durch lokale Maximasuche wird versucht, regelmäßige rhythmische Elemente zu identifizieren und dann aus den gewonnenen Daten auf das Tempo zu schließen. Als „Beat" bezeichnet Dixon das Tappen des Rhythmus, beispielsweise mit dem Fuß. Was für jedes Kind ein Leichtes ist, ist für den Computer eine wenig triviale Angelegenheit. Auf das Tempo wird geschlossen, indem man die Beats pro Minute zählt.

Andere melodiebasierte Ansätze setzen bei der Melodieanalyse auf ein spezielles Dateiformat (Musical Instruments Digital Interface (MIDI)) [MID96]. Dieses Dateiformat repräsentiert eine Notenpartitur und ist daher nicht ein binärer Datenstrom (siehe Kapitel 3.3). Die

Forschung hat hier bereits Dank des viel einfacher verarbeitbaren Formats große Fortschritte erzielt. Gjerdingen benutzt ein Neuronales Netz um Noten zu analysieren [Gje89]. Tseng konstruierte eine Applikation, die Themen in Musik findet [Tse99]. Bei diesem System kann man eine Suchanfrage in Noten stellen. Eine Stufe weiter geht McNab, der einen Prototyp beschreibt, der eine gesungene Suchanfrage versteht [McN96]. Eine umfassendere Beschreibung dieses Systems ist in [McN99] zu finden. Hauptprobleme der erwähnten Lösungen sind Tonhöhe und eine unscharfe, verfälschte Suchanfrage durch den Benutzer. Eine sehr verständliche Einführung in dieses Forschungsfeld ist neben den bisher erwähnten Quellen [UZ98]. Alle MIDI basierten Ideen setzen meist die Theorie von Ghias et al. ein [Ghi95]. Ghias formt eine vorgesummte Anfrage in drei Strings U, D, S um, wobei diese drei Zeichen eine höhere, niedrigere oder gleich hohe Note als die vorhergehende repräsentiert. Das Problem der Melodiesuche wird dadurch in eine einfache Stringsuche umgewandelt und kann mit genau diesen Algorithmen bestens gelöst werden [Dow99]. In [CHL98] findet man eine Zusammenfassung von Liu, Chen und Hsu's Verbesserungen dieses Weges, und in [BDE99] noch eine weitere Applikation.

All die gebrachten Ansätze enden jedoch entweder auf der einen (Frequenz) oder anderen (Zeit) vorgegebenen Achse und versuchen nicht diese beiden Ebenen zu verknüpfen. Diese Arbeit stellt eine Möglichkeit dar, sowohl Zeit- als auch Frequenzachsen zu berücksichtigen. Die genaue Verfahrensbeschreibung ist Gegenstand von Kapitel 4.

3.2. Grundbegriffe

Es folgen nun einige Grundbegriffe, die nicht als vollständige Definition im Sinne der Akustik, sondern zum weiteren Verständnis der Arbeit gedacht sind.

Musik ist die gesetzmäßige Anordnung von Klängen. Ein **Klang** ist die Folge regelmäßiger Luftschwingungen – im Gegensatz zum Geräusch, das aus unregelmäßigen Luftschwingungen gebildet wird. Die kleinste Einheit der Musik ist der Ton. Ein **Ton** ist ein Klang mit einer bestimmten Dauer, Höhe, Lautstärke und Klangfarbe [Enc97].

- Die **Tonhöhe** ist physikalisch messbar und hängt von der Schwingungsfrequenz ab. Je größer die Frequenz einer Schwingung ist, desto höher der Ton [Enc97].
- **Lautstärke** ist die Stärke einer Tonwahrnehmung; die physikalisch passende Größe ist die Amplitude [Bro78]. Im Medium Luft entspricht die Amplitude einer Schallwelle der Stärke der sich bewegenden Luftmoleküle.

- Die **Klangfarbe** ist der zusätzliche Aspekt neben den vorher genannten Größen, den ein Klang inne hat. Klänge bestimmter Musikinstrumente weisen unabhängig von ihrer Grundfrequenz Gebiete verstärkter Teilschwingungen auf, die an feste Frequenzbereiche gebunden sind. Mit zunehmender Klangstärke verlagert sich bei gleicher Tonhöhe das Intensitätsmaximum auf Teilschwingungen höherer Ordnungszahlen [Bro78]. Erzeugt man auf zwei verschiedenen Instrumenten einen Ton mit gleicher Tonhöhe und selber Lautstärke, so klingen sie verschieden, da nicht nur der reine Ton erklingt, sondern auch abhängig vom Instrument Anteile von höheren Frequenzen erklingen. Hauptsächlich sind dies Vielfache der Grundfrequenz, so genannte Obertöne. Die unterschiedlichen Klangfarben werden somit durch die Obertöne unterschieden. Die physikalische Repräsentation der Klangfarbe ist daher auch die Wellenform.

Es gibt drei grundlegende Ordnungssysteme für Musik: Harmonie, Melodie und Rhythmus.

Harmonie ist das gleichzeitige Erklingen mehrerer Klänge, das Gefüge der Töne [Bro78].

Melodie ist die lineare Anordnung von Tönen, d.h. eine geordnete Abfolge von Tönen in bestimmten Tonhöhen und -dauern, die in einem zeitlichen Kontext so miteinander verbunden sind, dass sie eine zusammenhängende musikalische Äußerung ergeben [Enc97].

Rhythmus steht in enger Beziehung zur Melodie, da auch er in Relation zur Zeit steht. Rhythmus gibt die Dynamik der Musik bezogen auf die Zeit an [Enc97]. Komponenten des Rhythmus sind Takt, Metrum und Tempo.

- **Takt** gruppiert die Notenwerte zu einer Einheit, die als gleichmäßig wiederkehrendes Bezugsschema wechselnden rhythmischen Gestalten zugrunde liegt (z.B. Dreivierteltakt) [Bro78].
- Das **Metrum** wiederum regelt innerhalb des Taktes die Betonung. Beispielsweise ist im Dreivierteltakt das erste Viertel stark, das zweite gar nicht und das dritte schwach betont [Enc97].
- **Tempo** gibt die absolute Dauer der Notenwerte an [Bro78]. Notenwerte stellen nur Geschwindigkeitsrelationen untereinander her, es fehlt jedoch eine exakte Grundgeschwindigkeit, die das Tempo darstellt.

3.3. Digitalisierte Musik

Musik ist ein analoges Signal und wird durch ein Medium übertragen. Zur Verarbeitung am Computer muss es zuerst aufgezeichnet und danach digitalisiert werden.

Das wohl bekannteste Gerät zur Aufnahme von Musik ist das Mikrofon. Beim Mikrofon werden die analogen Audiowellen durch eine Membran aufgefangen und dann, je nach Art des Mikrofons, in einen Stromimpuls umgesetzt. Beim elektrodynamischen Mikrofon beispielsweise ist an der Membran eine Spule befestigt, die sich frei schwebend in einem Magnetfeld befindet. Beginnt die Membran zu schwingen, so wird in der Spule durch Induktion Strom erzeugt [Bro78]. Das resultierende Signal ist jetzt zwar eine elektrische Größe, allerdings noch immer analog. Bevor man diskrete Größen erhält, muss daher das Signal durch einen Analog/Digital-Wandler (A/D-Wandler) laufen. Die Amplitude des Audiosignals wird im A/D-Wandler zu festgelegten Zeitpunkten periodisch gemessen und der Messwert als digitale Zahl ausgegeben. Dieser Vorgang heißt „sampling“. Wie oft das Signal abgetastet wird besagt die Sampling Rate (SR), die in Abtastungen pro Sekunde (Hz) (z.B. 44100 Hz) gemessen wird. Die SR ist normalerweise ein Vielfaches der Spitzenfrequenz des abgetasteten analogen Signales. Das erhaltene digitalisierte Audiosignal wird allgemein als Pulse Code Modulation (PCM) bezeichnet. Wie auf digitalen Maschinen üblich, wird der Zahlenbereich des A/D-Wandlers durch ein Vielfaches der Zahl 2 vorgegeben und ist ganzzahlig [Wha00]. Der durch den A/D-Wandler ausgegebene Wert ist eine 16-bit Zahl und wird durch den Zahlenbereich

$$[-32768, 32767]$$

abgedeckt. Abbildung 3.2 zeigt ein typisches PCM-Audiosignal der Länge 5 Sekunden. Zum besseren Verständnis wird nachfolgend immer eine kontinuierliche SR von 44100 Hz angenommen[1].

Musik kann auf Festspeichermedien (z.B. Festplatten) in vielen verschiedenen Formaten vorliegen (WAV, AU, MP3,...). Wie in Kapitel 4.1 ausgeführt, ist diese Arbeit allerdings Dateiformat unabhängig. Auf die unterschiedlichen Audiodateiformate wird im Rahmen dieser Arbeit daher nicht eingegangen.

Es gibt noch ein anderes Repräsentationsformat von Musik, das völlig anders als ein PCM Signal aufgebaut ist. Das Musical Instruments Digital Interface (MIDI) Dateiformat enthält keine analoge Klangkurve, wie sie durch Mikrofonaufnahmen entsteht, sondern ist eine Gerätebeschreibung. Bei diesem speziellen Format wird festgelegten Instrumenten mitgeteilt, wel-

[1] Diese Abtastrate entspricht der Qualität einer Audio CD Aufnahme

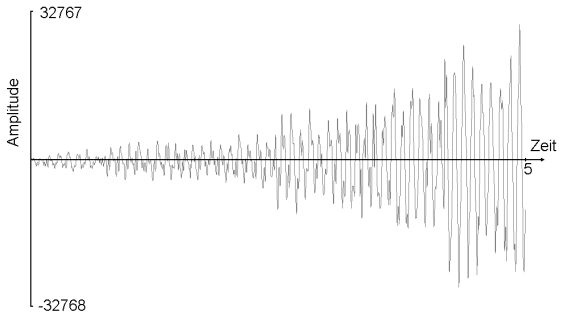

Abbildung 3.2.: PCM Audiosignal

che Note sie zu welchem Zeitpunkt wie lange spielen sollen. Nachdem bei einem normalen Musikstück mehr als ein Instrument erklingt, besteht eine MIDI-Datei aus mehreren Spuren für unterschiedliche Instrumente. Eine Abbildung von Gesang ist nicht möglich. Diese Beschreibung entspricht intern viel mehr einem Text, respektive einer Partitur, als einem Klang.

3.4. Neuronale Netze

Ein Neuronales Netz besteht aus vielen Neuronen (engl: *units*)[2]. Ein Neuron hat mehrere gerichtete gewichtete Eingänge von Vorgängerneuronen und einen Ausgang. Übersteigt die Summe der Werte der Eingangsgewichte einen gewissen Schwellwert, so wird das Neuron aktiviert – es feuert – und gibt ein Signal an seinen Ausgang weiter [Spe96]. Die Hauptaufgabe eines Neuronalen Netzes besteht darin zu lernen. Dieses wird durch Änderungen in den Gewichten der Neuronen möglich.

Es gibt drei mögliche Arten, wie ein Neuronales Netz lernen kann [Spe96]:

Überwachtes Lernen (engl: *supervised learning***)** – Der vom Benutzer erwünschte Ausgabewert wird mit dem Ausgabewert des Netzes verglichen. Die berechnete Abweichung wird als Fehler an das Netz übermittelt. Ziel ist die Minimierung des Fehlers während des Trainings.

Unüberwachtes Lernen (engl: *unsupervised learning***)** – Das Netz wird wiederholt mit Eingaben versorgt. Es wird dem Netz selbst überlassen eine Ordnung zu finden. Die SOM arbeitet nach diesem Gesichtspunkt.

[2] *unit* ist abgeleitet vom Begriff *neural processing units*, allerdings hat sich in der englischsprachigen Fachliteratur dieser Begriff durchgesetzt

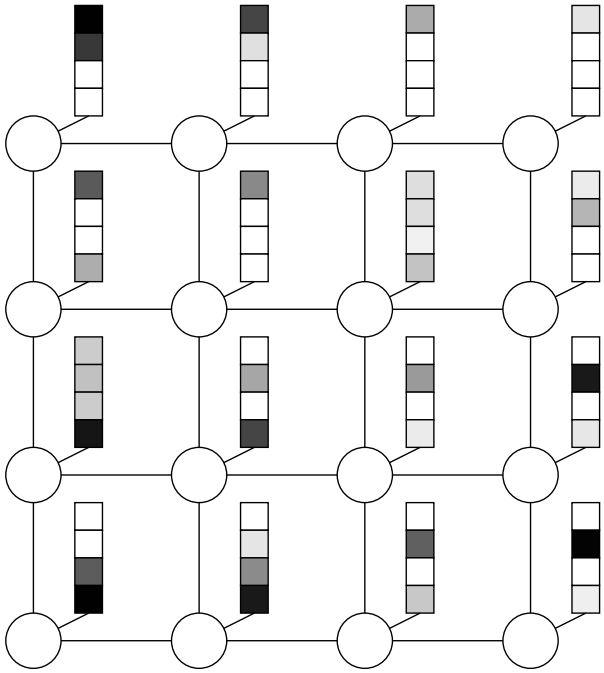

Abbildung 3.3.: Architektur einer 4×4 SOM

Verstärkendes Lernen (engl: *reinforcement learning***)** – Hier wird dem Netz nur mitgeteilt, ob ein Lernschritt gut oder schlecht war. Da man hier zwar eine Rückgabe an das Netz gibt, aber keine explizite Korrekturanweisung, ist dies eine Mischform aus den beiden anderen Arten.

Die SOM, auch *Self-Organizing Feature Map* oder *selbstorganisierende Karte* genannt, ist ein Vertreter des unüberwachten Lernens. Das Konzept der SOM stammt von Teuvo Kohonen [Koh81]. Bei der SOM sind die Neuronen in einem zweidimensionalen Gitter, der Karte (engl: *map*), angeordnet. Jedes Neuron besitzt einen so genannten Gewichtsvektor (engl: *weight vector*). Da sich die Neuronen nicht im Gitter bewegen können, werden durch den Lernvorgang die Gewichte der Gewichtsvektoren verändert. Die der SOM zugrunde liegende Motivation beschreibt Kohonen selbst folgendermaßen [Koh81]:

> According to the views acquired by many physiologists and psychologists, the most important task of the brain, in addition to the control of autonomous functions, is to form various kinds of modles or maps of the environment, sensory experiences, as well as even more abstract occurrences. Such an ability would already account for the majority of cognitive and memory processes.

Abbildung 3.3 zeigt die Architektur einer 4×4 SOM. Jedes Neuron (Kreis) besitzt einen

4-dimensionalen Gewichtsvektor, dargestellt durch ein 4×1 Rechteck. Jedes Neuron ist mit seinen direkten Nachbarn verbunden, dadurch bleiben nachbarschaftliche Beziehungen durch den Lernvorgang (engl: *training*) erhalten. Die Gewichtsvektoren der SOM und die Eingabevektoren (engl: *feature vectors*) haben dieselbe Dimension (in Abbildung 3.3 beträgt die Dimension 4). Die Dimension der SOM beträgt 2, da jedes Neuron durch Angabe der Spalte und Zeile eindeutig identifiziert werden kann. Dadurch, dass die Eingabevektordimension viel größer sein kann als die des Netzes, ist das Ergebnis eine Abbildung des *n*-dimensionalen Eingaberaumes auf eine 2-dimensionale Karte [Spe96].

Lernalgorithmus der SOM [Bay95, Mer95, MR00, Spe96]:

1. Initialisierung der Gewichtsvektoren: Die einzelnen Gewichte der Vektoren werden mit zufälligen Werten befüllt.

2. Zufällige Wahl eines Eingabevektors: Ein beliebiger Vektor aus dem Eingaberaum wird ausgewählt.

3. Ähnlichkeitsberechnung: Der Abstand des Trainingsvektors zu den Gewichtsvektoren wird errechnet.

4. Suche des Neurons mit dem ähnlichsten Gewichtsvektor: Das Gewinnerneuron mit dem minimalen Abstand zum Trainingsvektor wird ermittelt.

5. Adaption des Gewinnerneurons: Beim Training wird der Gewichtsvektor des Gewinnerneurons um die Lernrate (engl: *learningrate*, *learnrate*) dem Eingabevektor angeglichen. Bei Lernrate 1 wird der Gewichtsvektor dem Trainingsvektor gleich gesetzt, bei Lernrate 0 erfolgt keine Adaption.

6. Adaption der Nachbarn: In einem durch eine über die Zeit abnehmende Nachbarschaftsfunktion beschriebenen Bereich befindliche Neuronen werden ebenfalls an den präsentierten Eingabevektor angepasst.

7. Adaptionsparameter verringern: Lernrate und Nachbarschaftsradius sinken. Je länger das Training dauert, desto weniger Änderung erhält das Netz.

8. Wiederhole alle Schritte (bis auf den Ersten), bis alle Eingabevektoren der Karte zugewiesen wurden (engl: *mapping*).

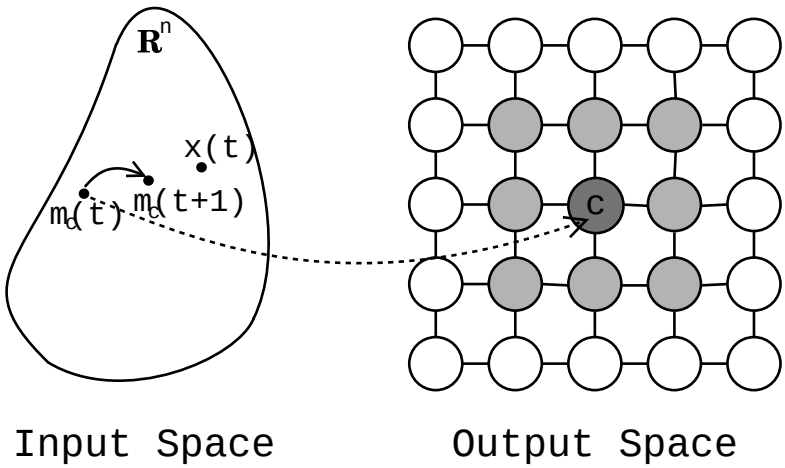

Abbildung 3.4.: Training eines Vektors $x(t)$ auf eine 5×5 SOM

Abbildung 3.4 zeigt das Training eines Eingabevektors $x(t)$ aus dem n-dimensionalen Eingaberaum (engl: *input space*) $\mathbf{R}^n$ auf eine 5×5 SOM (engl: *output space*). Das Neuron c mit Gewichtsvektor $m_c(t)$ wird zum Winner, d.h. der Gewichtsvektor $m_c(t)$ liegt dem Eingabevektor $x(t)$ am nächsten. Der Gewichtsvektor $m_c(t)$ wird nun um die Lernrate in Richtung des Eingabevektors $x(t)$ verschoben und resultiert im neuen Gewichtsvektor $m_c(t+1)$. Ist die Lernrate ungleich eins, stimmt der Eingabevektor mit dem adaptierten Gewichtsvektor nicht überein. Der Abstand des Eingabevektors $x(t)$ zum adaptierten Gewichtsvektor $m_c(t+1)$ wird daher berechnet. Diese Größe heißt Quantisierungsfehler (QF) (engl: *quantisationerror*) [MDR00] des Vektors $x(t)$ auf Neuron c. Zusätzlich zum Gewinnerneuron c (dunkelgrau) werden durch die Nachbarschaftsfunktion angrenzende Neuronen (hellgrau) adaptiert. Durch diese Änderung bilden sich Bereiche (engl: *cluster*) und die Karte bekommt eine Ausrichtung im Ganzen. Durch diese Adaption der Nachbarn ist die SOM Topologie erhaltend.

Abbildung 3.5 zeigt beispielhaft das Ergebnis eines SOM Trainings. Die Abbildung zeigt eine Karte einer 10×10 SOM. Im Hintergrund ist die gesamte Karte sichtbar, während am rechten und linken Bildrand drei Cluster vergrößert dargestellt sind. Kapitel 5 enthält die Karten zu den in dieser Arbeit präsentiertem Verfahren.

Es existieren eine ganze Fülle verwandter Lernalgorithmen. **Learning Vektor Quantisation (LVQ)** [Koh97] besitzt keine Nachbarschaftsfunktion und die Neuronen gehören vorher festgelegten, während des Trainings nicht mehr änderbaren, Klassen an. Da Klasseninformationen in dem hier behandelten Fall allerdings vorher nicht bekannt sind, ist dieser Ansatz

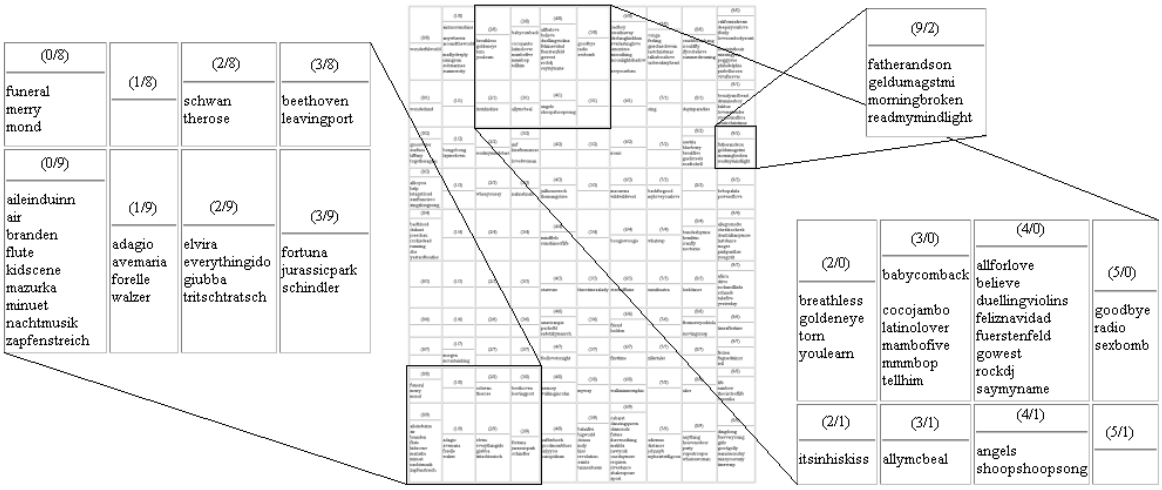

Abbildung 3.5.: Eine 10×10 SOM mit drei hervorgehobenen Clustern

hier nicht anzuwenden [Bay95]. Es existieren auch eine ganze Anzahl an Erweiterungen für den eigentlichen SOM Algorithmus. Beim **Growing Grid (GG)** [BM93] wird nach einer fixen Anzahl von Trainingsschritten ein neues Neuron hinzugefügt bzw. entfernt. Da Daten oftmalig Hierarchien enthalten, gibt es die **Hierarchical Feature Map (HFM)** um diese abzubilden. Hier werden unabhängige SOMs auf einer Hierarchie trainiert. Da bei der HFM sowohl die Tiefe der Hierarchie als auch die Ausmaße der SOM vor Trainingsbeginn festgelegt werden müssen, wurde die **Growing Hierarchical Self-Organizing Map (GHSOM)** entwickelt [MDR00]. Die GHSOM unterliegt keiner der beiden Einschränkungen und kann sowohl in die Breite (GG) als auch in die Tiefe (HFM) wachsen.

4. Feature Extraktion und Clusterung

Das folgende Kapitel beschäftigt sich mit dem hier gebrachten Ansatz Musik zu klassifizieren. Das Verfahren wird erläutert und am Computer nachgebildet, wobei PC bedingte Einschränkungen aufgezeigt werden. Abbildung 4.1 gibt einen Verfahrensüberblick des im folgenden Absatz dargestellten Ablaufs, wobei Zustände durch Kästen und Verfahren durch Rauten dargestellt werden.

Musik, in Form eines PCM-Signals, wird durch den Mediaplayer X-Multimedia System (XMMS) abgespielt. XMMS vollführt intern bereits eine Fouriertransformation, wodurch Klangspektren von Momentaufnahmen erzeugt werden. Die Klangspektren werden zu fünfsekundigen Segmenten gruppiert und bestimmte Frequenzbänder ausgewählt. Diese Segmente durchlaufen eine Lagrange-Interpolation um äquidistante Stützstellen für eine darauf folgende Fast Fouriertransformation (FFT) zu erhalten. Aus den Koeffizienten der Fouriertransformation werden SOM Eingabevektoren erzeugt. Die SOM verarbeitet nun diese Segmente zu einer Segmentkarte. Ähnliche Klangcharakteristik aufweisende Segmente werden in benachbarten Bereichen der Karte gefunden. Aufgrund der Lage der Segmente auf der Segmentkarte wird, um zu einer Karte von Musikstücken zu gelangen, abermals eine SOM trainiert. Lieder

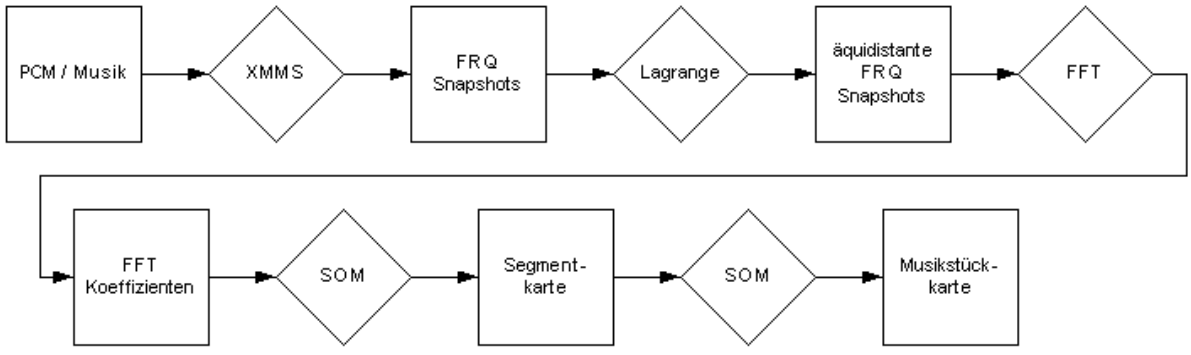

Abbildung 4.1.: Verfahrensübersicht

mit ähnlicher Segmentverteilung werden in benachbarten Bereichen der resultierenden Musikstückkarte gefunden.

Kapitel 4.1 erklärt die Gewinnung des Signals aus XMMS, Kapitel 4.2 zeigt die Erstellung der Vektoren für die Self-Organizing Map (SOM), in Kapitel 4.3 wird die Karte der Musiksegmente und im darauf folgenden Kapitel 4.4 die Karte von Musikstücken generiert.

4.1. Signalextraktion

Wie schon in Kapitel 3.3 erwähnt, liegt Musik im so genannten PCM-Format am Computer vor. Doch dies ist eigentlich nur das Format, mit dem der Sound-Prozessor angesprochen wird. Musik ist, wie auch andere Programme und Daten, auf Festspeichermedien (z.B. Festplatten) gespeichert. Für Musik gibt es noch dazu eine ganze Fülle von Dateiformaten. Der erste Schritt des vorgestellten Algorithmus ist logischerweise das Öffnen der Datei. Leider legt dieser Schritt das ganze Verfahren auf ein einziges Dateiformat fest. Heutige Programme erledigen das Dateimanagement durch Auslagerung in **Plugins**. Plugins sind eigenständige Unterprogramme, die für autonome Aufgaben im Programm benutzt werden und mit dem Hauptprogramm durch wohl definierte Schnittstellen kommunizieren. Durch diesen modularen Aufbau können mehrere Dateiformate unterstützt werden und eine Erweiterung auf neue, andere Audiodateiformate ist einfach möglich. Der Aufwand für die Eigenentwicklung eines solchen Programms würde allerdings wenig Sinn machen, da fast jedes am PC laufende Abspielprogramm für Musikformate (engl: *Mediaplayer*) heute so aufgebaut ist. Da als Entwicklungsplattform Linux ausgewählt wurde, fiel die Wahl auf den inzwischen meist verwendeten Open-Source Mediaplayer XMMS. Nachdem Änderungen im Programmcode vorzunehmen sind, ist die Wahl von Open-Source Software, wo der Quellcode verändert werden darf, essentiell.

XMMS ist ein Mediaplayer unter dem X-Window System, das auf Unix Plattformen hauptsächlich in Verwendung ist. XMMS besteht aus **Input-Plugins**, die für das Öffnen der Dateien zuständig sind, **Output-Plugins**, die die Ausgabe auf den Sound-Prozessor bzw. auf andere Medien bewerkstelligen, und **Visualization-Plugins**, die zur Unterhaltung der Benutzer auf den Audiodaten beruhende visuelle Ausgaben auf dem Bildschirm produzieren. Um die PCM Werte zu erhalten wird die Ausgabe, bevor sie den Sound-Prozessor erreicht, abgefangen und in eine Datei umgeleitet.

Nachdem XMMS im Quellcode vorliegt und modular aufgebaut ist, kann der Datenstrom

an vielen Stellen angezapft werden. Welche Stelle dafür ideal ist, wird nun analysiert.

4.2. Vorverarbeitung

Inhalt dieser Arbeit ist ein klang-dynamischer Ansatz für jede Art von Musik. Während die Dynamik bzw. Änderung des Klangs genauso wie die Melodie über die Zeitachse $d(t)$ läuft, bewegt sich der Klang auf der Frequenzebene $k(F)$. Ziel ist also die Konstruktion einer Funktion wie in Ausdruck 4.1.

$$k(F) \circ d(t) \longrightarrow kd(Ft) \tag{4.1}$$

Nachdem eine exakte mathematische Funktion weder für Klang noch für dessen Dynamik existiert, dient die eingeführte Symbolik nicht zur Erstellung eines mathematischen Beweises bzw. einer Theorie, sondern ausschließlich zum Verständnis des Lesers. Zur Konstruktion der Funktion 4.1 ist es essentiell Vektoren durch eine Vorverarbeitung zu erstellen.

Der Zweck der Vorverarbeitung ist es, der Self-Organizing Map (SOM) zu ermöglichen, leichter Strukturen zu finden. Wie Teuvo Kohonen sagte [Kas00]:

> The SOM just makes a description of data,
> if data is garbage, then nothing happens!

Ein gültiger SOM Eingabevektor muss folgenden Gesichtspunkten genügen [Bay95]:

- Alle Eingabedaten (engl: *features*) müssen numerisch sein
- Die Eingabedaten müssen zu Vektoren gleicher Länge zusammengefasst sein
- Die Wertebereiche aller Eingaben sollten gleich und beschränkt sein.

Der folgende Abschnitt behandelt somit die Erstellung gültiger Eingabevektoren, ausgehend von dem in Kapitel 3.3 beschriebenen Audiosignals.

4.2.1. Signalaufbereitung

Ausgangssignal ist ein Signal über die Zeit $pcm(t)$ mit den in Kapitel 3.3 beschriebenen Eigenschaften. Als erster Schritt steht, wie bei Wold (siehe Kapitel 3.1), eine Transformation in die Frequenzebene.

$$pcm(t) \longrightarrow pcm(F)$$

Die Mathematik lehrt, dass jede periodische stetige Funktion $f(x)$ sich als Summe von Sinus- und Kosinusschwingungen, die Fourierreihe (FR) (siehe Ausdruck 4.2), darstellen lässt [Bar94, Ram96].

$$f(x) \sim \frac{a_0}{2} + \sum_{k=1}^{m} \left(a_k \cos kx + b_k \sin kx\right) \qquad ,wobei \tag{4.2}$$

$$\begin{aligned} \tfrac{a_0}{2} &= \tfrac{\int_a^b f(x)dx}{b-a} \\ a_k &= \tfrac{\int_a^b f(x)\cos kx\ dx}{\int_a^b (\cos kx)^2 dx} \\ b_k &= \tfrac{\int_a^b f(x)\sin kx\ dx}{\int_a^b (\sin kx)^2 dx} \end{aligned}$$

a_1, b_1 ist die Grundschwingung (auch 1. Harmonische), die weiteren Koeffizienten sind jeweils ein Vielfaches der Grundschwingung und werden Oberwellen genannt. Beispielsweise hat eine langsam variierende Funktion nur niederfrequente spektrale Komponenten. Die Komponenten einer sich schnell ändernden Funktion hingegen sind über einen weiten Frequenzbereich verteilt [But98]. Die FR ist, wie aus Ausdruck 4.2 ersichtlich, die Summe einzelner Frequenzen [Sei96]. Unter Nutzung dieses Wissens wird nun eine Transformation des auf die Zeitachse bezogenen Signals in die Frequenzebene mittels der Fouriertransformation durchgeführt. Die für die weitere Rechnung interessanten Daten sind hauptsächlich die Koeffizienten der FR. Wie in 4.2 gezeigt, können die Koeffizienten durch Integration berechnet werden. Die Bestimmung der einzelnen Integrale ist jedoch ein aufwändiges Unterfangen, weshalb sich ein für den PC optimiertes Verfahren eingebürgert hat, die Fast Fouriertransformation (FFT). Bei der FFT wird das Problem der Integration dadurch gelöst, dass das „divide and conquer"-Prinzip angewandt wird. Eine FR kann durch die Summe zweier voneinander unabhängiger FR geschrieben werden. Bei der FFT werden die Eingangsdaten (Anzahl N) in zwei unabhängige Bereiche halbiert ($N/2$). Eingangswerte mit geraden Indizes $(N_2, N_4, \ldots)$ werden in einer FR dargestellt, jene mit ungeraden Indizes kommen in die zweite FR. Dieses Verfahren wird bis zum einfach zu lösenden Problem ($N = 2$) wiederholt [Bri95, Jam95].

Ein weiterer Grund, warum XMMS als Quelle für das Ausgangssignal herangezogen wird ist, dass XMMS eine Schnittstelle besitzt, die bereits ein solches Signal liefert. Die Gruppe der Visualization-Plugins verfügt über zwei Schnittstellen, die vom Input-Plugin gespeist werden [XMM00].

```
gint16[1] pcm_data[2][512]
gint16 freq_data[2][256]
```

[1] ganzzahliger 16-bit breiter Datentyp

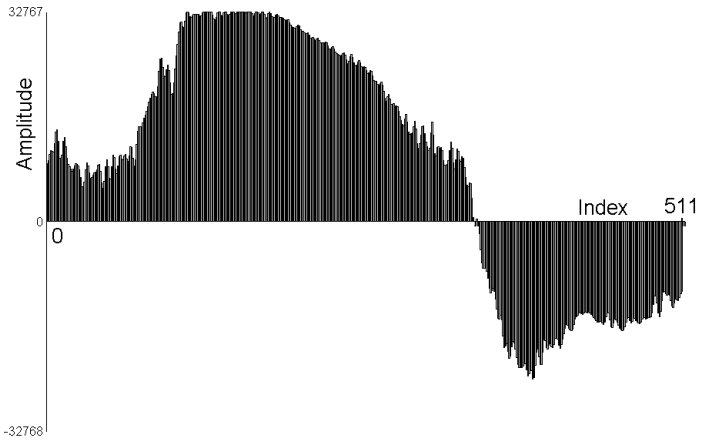

Abbildung 4.2.: PCM-Datenarray

pcm_data bezeichnet eine Gruppe von 512 hintereinander liegenden Samples des PCM Signals und wird von nun an mit **PCM-Daten** bezeichnet. Abbildung 4.2 zeigt ein beispielhaftes PCM-Daten Array. Auf der vertikalen Achse sind die Amplituden der auf der horizontalen Achse 512 aufgetragenen Werte des Arrays.

freq_data beinhaltet 256 Fourier Koeffizienten einer durch XMMS durchgeführten FFT an den PCM-Daten. XMMS liefert den Betrag der komplexen FR, weshalb man nicht 512 Koeffizienten, sondern 256 positive Betrags-Koeffizienten erhält. Diese neue Datenreihe wird im weiteren als **FRQ-Daten** bezeichnet. Abbildung 4.3 zeigt ein FRQ-Datenarray. Auf der vertikalen Achse ist der Wert der 256 einzeln auf der horizontalen Achse aufgetragenen Arrayinhalte verzeichnet.

Genauso wie bei Feiten et al. wird in weiterer Folge eine fixe Anzahl hintereinander liegender PCM-Samples genommen und für diese Werte eine FFT durchgeführt. Die Anzahl der Samples ergibt sich aus folgender Überlegung:

Musik besteht meist aus zwei Kanälen. 512 Samples je Kanal bedeuten 1024 Samples, die bei einem Lesezugriff zu lesen sind. Ein Sample besteht aus einer 16-bit Zahl, somit sind pro Lesezugriff 2048 Bytes zu lesen (Rechnung 4.3). Eine Auswahl von 512 aufeinander folgenden Samples deckt somit eine Dauer von etwa 10 Millisekunden (ms) innerhalb des Musikstücks ab (Rechnung 4.4).

$$\frac{2 \cdot 512 \cdot 16}{8} = 2048 \tag{4.3}$$

$$\left(\frac{1}{44100}\right) 512 = 0.0116 \tag{4.4}$$

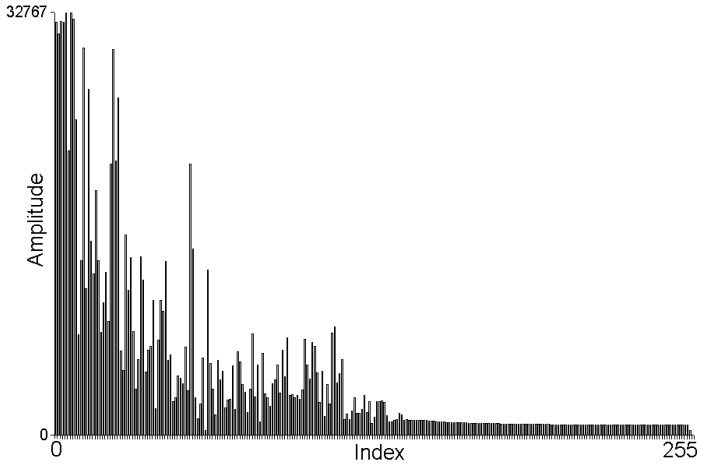

Abbildung 4.3.: FRQ-Datenarray

Es wurde somit ein spezielles Visualization-Plugin implementiert, das die beiden oben beschriebenen Datenreihen aus einer beliebigen Audiodatei extrahiert und in eine separate Datei speichert. Im nächsten Schritt werden dann aus dieser Datei die FRQ-Daten für einen Kanal (linker Kanal) extrahiert und nach Entfernen der Nullwerte am Beginn und Ende der Datei in eine neue Datei gespeichert[2]. Diese Datei wird zur weiteren Verarbeitung genutzt.

4.2.2. Signaltransformation

Alle bisherigen Arbeiten enden hier und generieren direkt aus diesen Koeffizienten einen Eingabevektor. Kohonen selbst schlägt in seinem Buch vor, Fourierkoeffizienten als Merkmale zur Generierung von SOM Eingangsvektoren zu benutzen [Koh97]:

> ... one can use the elements of the ... Fourier amplitude-spectrum as features.

Das erhaltene Frequenzspektrum (FRQ-Daten) stellt eine Momentaufnahme dar (engl: *snapshot*) und ist für die Klanganalyse geeignet. Um auf die Zeitachse zurückzukehren wird eine Gruppe von hintereinander liegenden Snapshots ausgewählt. In Experimenten stellte sich heraus, dass die besten Ergebnisse bei einer Snapshot Gruppe von insgesamt 5 Sekunden erzielt werden konnten.

Leider erweist sich die Entnahme einer fünfsekundigen Probe als schwierig, da der komplexe Aufbau von XMMS keine zeitlichen Schranken für die Berechnung erlaubt. Einerseits

[2]Diese Schritte konnten nicht im Plugin durchgeführt werden, da die Abarbeitung einer Abspielschleife, d.h. die Speicherung eines PCM bzw. FRQ Arrays, zu harten Zeitbedingungen erfolgt.

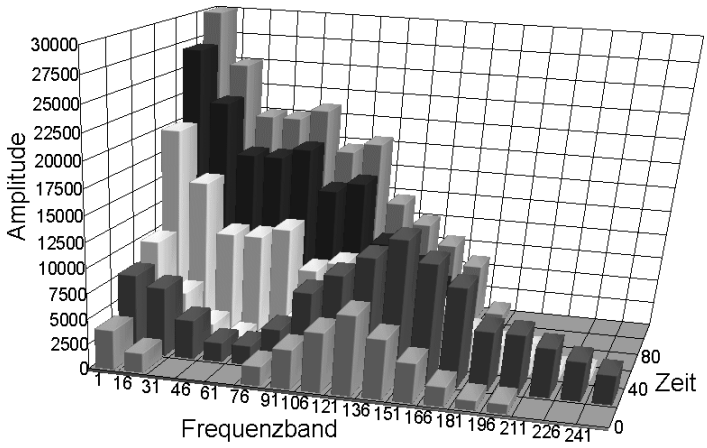

Abbildung 4.4.: gesamt über die Frequenz betrachtete Frequenzspektren

entstehen Timingprobleme durch die Plugin-Architektur, andererseits durch die Prozessorlast. Da Linux auch kein Echtzeitbetriebssystem ist, kann der XMMS Prozess auch von anderen Prozessen unterbrochen werden. Um exakte Zeitpunkte zu erhalten wird in das programmierte Plugin ein Timer eingebaut, der die real verstrichene Zeit zwischen zwei Sample-Arrays misst. Weiters wird durch die Auswahl einer sehr schnellen CPU und Beendigung aller nicht benötigten Programme dafür gesorgt, dass XMMS immer genügend Rechenzeit zur Verfügung steht. Die tatsächliche Prozessorlast lag bei den Experimenten während des Abspielens der Musik dauerhaft bei unter 10 Prozent. Die zeitlichen Probleme sind damit vorerst gelöst, doch bedarf es einer späteren Berücksichtigung der eingeführten Zeitstempel. Bei den Experimenten zeigte sich, dass XMMS etwa alle 20 ms ein Datenarray liefert. Für fünf Sekunden ergeben sich somit 250 Datenwerte.

Durch die am Computer gegebene Beschränkung des Speicherplatzes war allerdings noch eine weitere Optimierung notwendig. In Experimenten zeigte sich, dass es nicht nötig war jedes einzelne Frequenzband zu betrachten, sondern dass hier eine deutliche Datenreduktion wenig Auswirkungen auf die Ergebnisse hat. Daher wurden nicht alle 256 Frequenzbänder betrachtet, sondern nur jedes fünfzehnte, wodurch insgesamt 17 Bänder betrachtet werden (siehe auch Kapitel 4.2.2). Abbildung 4.4 zeigt 6 aufeinander folgende Snapshots (20 ms Abstand) in 17 Frequenzbändern $(1, 16, \ldots, 241)$ mit jeweils einer Amplitude bis zu 32767.

Durch diese Zusammenfassung sind die Daten jetzt in Form einer im Ausdruck 4.5 gezeigte zweidimensionale 250×17 - Datenreihe gegeben.

$$\begin{array}{cccccc} t_1: & k_{1\,1} & k_{1\,16} & k_{1\,31} & \cdots & k_{1\,241} \\ t_2: & k_{2\,1} & k_{2\,16} & k_{2\,31} & \cdots & k_{2\,241} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ t_{250}: & k_{250\,1} & k_{250\,16} & k_{250\,31} & \cdots & k_{250\,241} \end{array} \tag{4.5}$$

In den Spalten befinden sich die Fourierkoeffizienten und in den Zeilen die einzelnen Samples. Die Zeilennummern t repräsentieren die Zeitstempel, die darauf hinweisen, dass die Abstände zwischen den einzelnen Samples nicht regelmäßig sind. Das Ganze ist somit wieder eine Funktion über die Zeit. Bei diesem Schritt kann man jetzt abbrechen und aus diesen Datenreihen einen Eingabevektor erzeugen, doch wird hier noch einen Schritt weiter gegangen. Um, wie Kohonen und andere auch, Spektren von der SOM gruppieren zu lassen, wird nochmals eine Fouriertransformation vollzogen. Diesmal ist jedoch nicht ein Audiosignal die zu transformierende Funktion, sondern eine Funktion, die aus den Spalten aus Ausdruck 4.5 durch Interpolation erstellt wird. Da die einzelnen Frequenzbänder für unterschiedliche Frequenzbereiche stehen, kann man sie getrennt betrachten. Somit erhält man 17 Funktionen die zu interpolieren sind. Da allerdings nur wieder diskrete Eingangswerte für die darauf folgende FFT interessant sind, wird die Funktion selbst nicht berechnet, sondern nur deren Funktionswerte.

Zur Anwendung kommt das Lagrange-Verfahren (Ausdruck 4.7). Bei diesem Verfahren hängen die Polynome nur von den Stützstellen x_i ab (siehe Ausdruck 4.6).

$$P_n(x_i) = f(x_i), \qquad (i = 0, 1, \ldots, n) \tag{4.6}$$

Für das Lagrange-Verfahren existieren sehr schnelle und bekannte Implementierungen, wie sie in [NR92, NR97] zu finden sind und hier zur Anwendung kommen.

$$P_n(x) = \sum_{i=0}^{n} \left(\prod_{j=0, j\neq i}^{n} \frac{x - x_j}{x_i - x_j} \right) f(x_i) \tag{4.7}$$

Ergebnis der Interpolation sind 256 äquidistante Funktionswerte pro Funktion, an denen nachfolgend eine FFT durchgeführt wird. Resultat sind 256 reale positive und negative Fourierkoeffizienten.

$$\begin{array}{cccc} f_{1\,1} & f_{1\,2} & \cdots & f_{1\,256} \\ f_{16\,1} & f_{16\,2} & \cdots & f_{16\,256} \\ \vdots & \vdots & \vdots & \vdots \\ f_{241\,1} & f_{241\,2} & \cdots & f_{241\,256} \end{array} \tag{4.8}$$

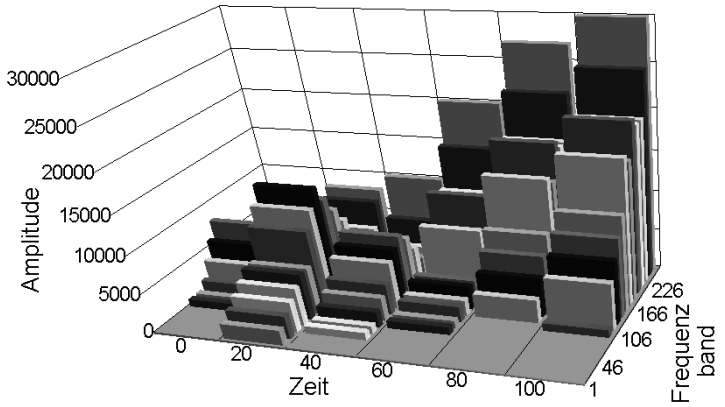

Abbildung 4.5.: getrennt über die Zeit betrachtete Frequenzspektren

Ausdruck 4.8 sieht auf den ersten Blick bekannt aus, doch enthalten die Spalten jetzt zusammengehörige Fourierkoeffizienten der einzelnen interpolierten Funktionen in den Zeilen. Abbildung 4.5 zeigt die Ausgangslage vor der Lagrangeinterpolation. Sie zeigt ein ähnliches Bild wie Abbildung 4.4, doch ist die Abbildung gedreht. Statt wie in 4.4 eine Funktion über Frequenzbänder zu betrachten, wird in 4.5 eine Funktion über die Zeit betrachtet.

Aus diesem Zahlenfeld wird nun ein SOM Eingabevektor erstellt. Da die SOM nur einstellige Vektoren akzeptiert, wird das zweidimensionale Feld in einen Zeilenvektor umgewandelt.

$$f_{1\,1}\ f_{1\,2}\ \cdots\ f_{1\,256}\ ,\ f_{16\,1}\ f_{16\,2}\ \cdots\ f_{16\,256}\ ,\ \cdots\ ,\ f_{241\,1}\ f_{241\,16}\ \cdots\ f_{241\,256} \tag{4.9}$$

Ausdruck 4.9 symbolisiert nun den Vektor für ein fünfsekundiges Stück eines Liedes. Einer dieser Vektoren besteht somit aus $256 \cdot 17 = 4352$ Werten (engl: *features*). Jedes Lied wird, wie oben beschrieben, in fünfsekundige Intervalle eingeteilt. Aus den resultierenden Vektoren wird das Eingabevektorfile erstellt. Das letzte Intervall eines Liedes wird sich in den meisten Fällen nicht genau auf 5 Sekunden ausgehen. Eine Möglichkeit wäre, das letzte Intervall mit Nullen auf die Länge von fünf Sekunden aufzufüllen. Nach Anwendung der SOM würden allerdings solche Stücke durch ihre Ähnlichkeit alle auf einem Knoten landen. Diese Ähnlichkeit wäre allerdings durch das Verfahren erzeugt worden und somit nicht in den Ursprungsdaten enthalten. Diese Lösung ist nicht erstrebenswert. Solche Schlusssegmente werden daher ignoriert. Ein Lied mit einer Länge von 4 Minuten wird also durch $(4 \cdot 60)/5 = 48$ Vektoren dargestellt. Um die Datenrate noch weiter zu senken wurde nur jedes zweite fünfsekundige Segment gewählt. In Experimenten erwies sich auch das als eine erhebliche Performancesteigerung.

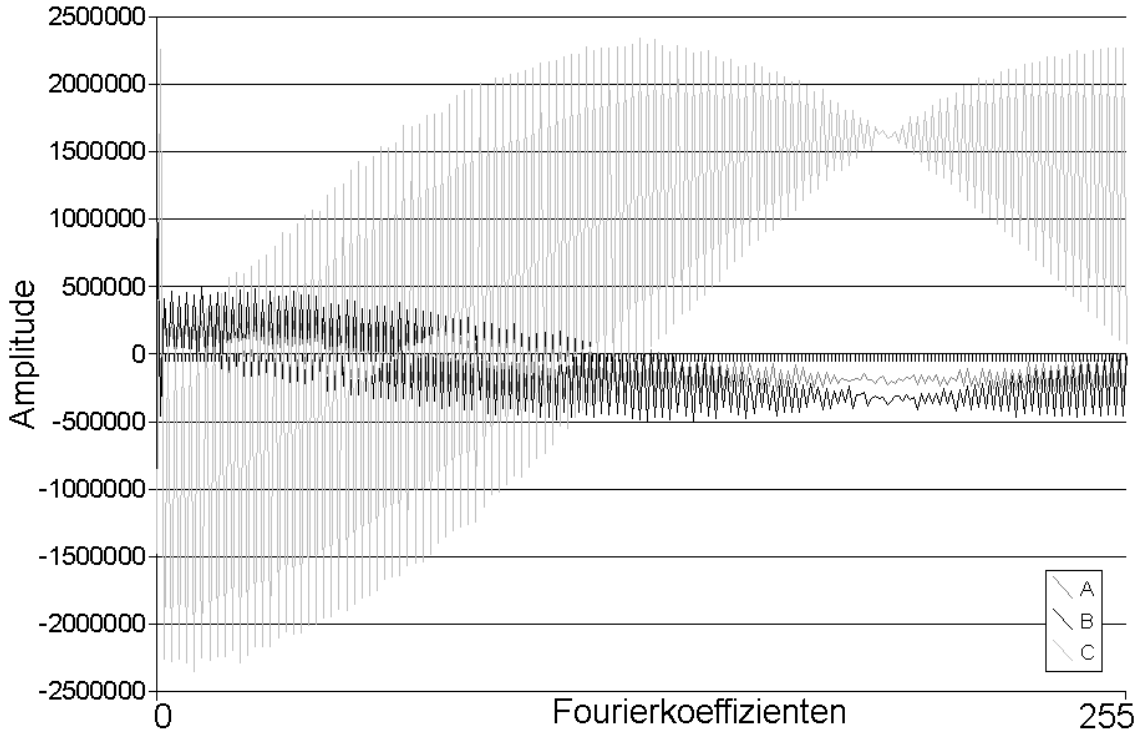

Abbildung 4.6.: drei fast gleiche Stücke auf Ähnlichkeit untersucht

Noch einmal sei daran erinnernt, dass das hier gezeigte Verfahren nicht Melodie verarbeitend ist, sondern die Änderung bzw. Dynamik des Klanges untersucht. Segmente, die in der Karte nebeneinander abgebildet sind, haben somit ähnliche klangliche Dynamik und nicht eine ähnliche Melodie. Zur Illustration soll Abbildung 4.6 dienen.

Die Grafik zeigt drei fast gleiche fünfsekundige Musikstücke, die aus dem Refrain des Lieds *Breathless (The Corrs)* stammen. Auf der horizontalen Achse sind die 256 Fourierkoeffizienten aufgetragen, und auf der vertikalen Achse deren Wert. Stück *A* beinhaltet ausschließlich Refrain. Stück *B* läuft Stück *A* um eine halbe Sekunde voraus und beinhaltet auch nur Refrain. Stück *C* läuft Stück *A* um eine Sekunde voraus und beinhaltet sowohl Refrain, als auch einen kurzen Teil einer Strophe, die eine gänzlich andere Instrumentierung besitzt. Abbildung 4.6 zeigt die Fourierkoeffizienten, die zur Vektorbildung für die Segmentkarte herangezogen werden. Während Stück *A* und Stück *B* gleich klingen, da dieselbe Instrumentierung und derselbe Gesang vorherrscht, obwohl sie versetzt sind, zeigt Stück *C* ein gänzlich anderes Verhalten der Koeffizienten. Dies zeigt eindeutig, dass das gezeigte Verfahren unabhängig von der Melodie nur auf die Dynamik des Klanges abzielt.

In einem letzten Vorverarbeitungsschritt erfolgt eine Nullpunktverschiebung über den gesamten Bereich. Da die Vektoren Kurvenbeschreibungen darstellen, muss man nur den kleinsten Koeffizienten über alle Vektoren wählen und dessen Absolutbetrag zu allen Koeffizienten dazuzählen um alle Kurven in den positiven Bereich zu übertragen. Zum Schluss werden die Vektoren noch auf den Bereich $[0,1]$ Längen normiert.

Überlegungen zum Datenvolumen

An dieser Stelle wird nun auf den Speicherplatz und die angesprochenen Optimierungen eingegangen.

Ein Koeffizient aus dem Vektorfile ist eine reelle positive Zahl – der Hauptspeicherbedarf liegt somit bei 32 bit = 4 Byte, doch auf Festspeichermedien benötigt diese Zahl weit mehr. Das Vektorfile ist ein Textfile, somit wird jedes Zeichen durch ein Byte dargestellt. Die Zahl 12345678.123456 benötigt also 15 Byte Speicherplatz. Um die Schätzung einfach zu halten wird der Speicherbedarf für einen Koeffizienten mit diesen 15 Byte beziffert. Zusätzlich zu diesen 15 Byte kommt noch ein Trennzeichen (Leerzeichen) von der Länge eines Bytes. Der Bedarf für eine Komponente eines Zeilenvektors liegt somit bei 16 Byte Speicherplatz auf dem Festspeichermedium. Ohne die oben angenommene Berücksichtigung von nur 17 Frequenzbändern wäre die Anzahl der Koeffizienten 65535. Somit läge der Speicherbedarf für einen Zeilenvektor bei $(16 \cdot 65535 = 1048560)$ 1 Megabyte (MB). Ist die durchschnittliche Länge eines Liedes 4 Minuten 10 Sekunden, so liegt der Speicherbedarf für ein einziges Lied bei 50 MB. Eine Liedsammlung mit nur 20 Liedern würde somit bereits ein Vektorfile von 1 Gigabyte (GB) füllen. Rein vom Speicherplatz der heutigen Festspeichermedien (Festplatten mit über 50 GB Speicher gibt es heute schon im Einzelhandel zu kaufen) würde das keine unlösbaren Probleme darstellen. Linux (Kernel 2.2) allerdings verhindert Dateigrößen über 2 GB[3]. Die Laufzeit des SOM Programms würde auch beträchtlich sein. Die hier verwendete Implementierung der SOM lädt außerdem das Vektorfile komplett in den Hauptspeicher. Ohne die oben beschriebenen Optimierungen wäre die Implementierung zur Klassifikation einer großen Liedsammlung nach heutiger Sicht erschwert. Durch die hier beschriebenen Optimierungen werden nur mehr etwa 3 % der ursprünglichen Datenmenge herangezogen. Rechnung 4.10 zeigt noch mal, dass nur jedes zweite Segment, von 17 aus 256 ausgewählten Frequenzbereichen, betrachtet wird.

$$\frac{\frac{17}{256}}{2} = 3\% \; der \; Datenmenge \tag{4.10}$$

[3]seit Kernel 2.4 ist diese Grenze auf 64 GB angehoben worden

Der Mediaplayer XMMS liefert nur alle 20 ms ein Datenarray, allerdings umfaßt dieses Array nur 10 ms, daher liegt tatsächlich die Datenrate noch niedriger, bei etwas über einem Prozent. Es wurden auch Versuche mit nur neun Frequenzbändern, unter Berücksichtigung von nur jedem dritten fünfsekundigen Segment, durchgeführt. Die tatsächliche Datenrate liegt hier bei weit unter einem Prozent. Es hat sich dabei zwar die gute Tendenz der Ergebnisse bestätigt, doch waren diese, verglichen mit den mit etwas umfangreicherem Datenvolumen durchgeführten Experimenten beschriebenen, wesentlich schlechter. Trotz dieser geringen Datenrate ist das verarbeitete Datenvolumen noch immer beträchtlich und ging jeweils bis an die Grenzen der Implementierung (siehe Kapitel 5.1.1).

4.3. Clusterung von Musiksegmenten

Die Vektoren der Segmente werden nun mittels der SOM gruppiert. Segmente mit ähnlicher klanglicher Dynamik werden in benachbarten Bereichen abgebildet. Die Ausgabe der SOM ist eine zweidimensionale Tabelle – die Karte. Die erhaltene Karte ist eine Karte von Musiksegmenten, sie heißt von nun an **Segmentkarte**.

Segmente, die ähnliche Dynamik aufweisen, liegen nahe beieinander. Das bedeutet, dass z.B. aufeinanderfolgende Segmente ein- und desselben Liedes üblicherweise auf den selben Knoten oder benachbarten Knoten abgebildet werden, da sie ähnliche Klangcharakteristika aufweisen. Lieder, die zwischen verschiedenen Genres springen, liegen auf mehreren Bereichen verteilt. Beispielsweise sind bei modernen Pop-Songs Strophe und Refrain teilweise sehr verschieden, was eine einheitliche Clusterung erschwert. Diese Karte erreicht bereits das gesteckte Ziel Musik zu klassifizieren. Durch die große Anzahl an Segmenten ist sie jedoch sehr unübersichtlich und daher nur schwer zu lesen. Durch die beträchtliche Größe der Karte, und da jedes Lied mehrmals eingetragen ist, ist eine nicht automationsgestützte Auswertung eher aufwändig. Um ganze Musikstücke anhand der Ähnlichkeit der Klangcharakteristika ihrer Segmente zu gruppieren, wobei jedes Musikstück nur einen einzigen Eintrag auf der Ausgabeseite besitzt, kann man neuerlich eine SOM anwenden.

4.4. Clusterung von Musikstücken

Basierend auf der Segmentkarte wird eine weitere Karte trainiert. Zur Generierung des Eingabevektorfiles für die Musikstückkarte werden die geometrischen Maße der Segmentkarte

als Grundlage genommen. Eine 2×4 SOM bedeutet, dass ein Vektor der Musikstückkarte $2 \cdot 4 = 8$ Attribute besitzt. Auf der Musikstückkarte ist jedes Musikstück nur genau einmal vertreten, unabhängig von seiner Länge. Die Anzahl der Vektoren der Musikstückkarte ist daher gleich der Anzahl der betrachteten Musikstücke. Die einzelnen Attribute des Eingabevektors werden aufgrund der Lage der Segmente eines Stückes auf der Segmentkarte gebildet. Den Eingabevektor kann man vorerst wie in Ausdruck 4.8 zur Einfachheit wieder als Tabelle betrachten. Ein Beispiel für einen 2×4 SOM Eingabevektor zeigt Ausdruck 4.11 .

$$\begin{matrix} l_{1\,1} & l_{1\,2} & l_{1\,3} & l_{1\,4} \\ l_{2\,1} & l_{2\,2} & l_{2\,3} & l_{2\,4} \end{matrix} \tag{4.11}$$

Auch dieser Vektor muss vor Verwendung im SOM Vektorfile als Zeilenvektor notiert werden, Ausdruck 4.12 .

$$l_{1\,1}\, l_{1\,2}\, l_{1\,3}\, l_{1\,4}\,,\, l_{2\,1}\, l_{2\,2}\, l_{2\,3}\, l_{2\,4} \tag{4.12}$$

Die einzelnen Komponenten l sind nun keine Fourierkoeffizienten, sondern neue berechnete Werte. Die Berechnung der Komponenten geschieht wie folgt:

Liegen beispielsweise ein Segment desselben Liedes A in der Segmentkarte auf Knoten S(0/1)[4], zwei auf S(1/3), und keines auf den anderen Segmenten, so lautet der Zeilenvektor für die Musikstückkarte für dieses spezielle Stück $(0\ 1\ 0\ 0\ ,\ 0\ 0\ 0\ 2)$. Um die Ergebnisse der Segmentkarte zu verunschärfen, wird zu den unmittelbaren Nachbarn noch ein Wert $(1/4)$ hinzugezählt. Somit ergibt sich ein Wert von $1 + 8 \cdot 1/4 = 3$, welcher auf insgesamt neun Knoten verteilt wird. Diese Summe bleibt immer gleich. Je besser ein Segment geclustert ist, desto besser wird das Zentrum gewichtet, und desto niedriger die Nachbarn. Ist ein Segment schlecht geclustert, so wird eine breite Streuung bis schließlich zur Gleichgewichtung vorgenommen. In der Implementierung wird der Quantisierungsfehler (QF), der eine Maßzahl für die Qualität des Mappings ist, benutzt um diese Werte zu berechnen. Vektoren mit niedrigem QF liegen besser auf dem Knoten, als Vektoren mit hohem QF. Diese Streuung wird vorgenommen um nachbarschaftliche Beziehungen zu förden. Durch die Mitgewichtung der Nachbarn werden Gebiete, wo die Segmente eines Liedes direkt nebeneinander liegen aufgewichtet, wodurch Clusterbildung gefördert wird.

Der QF wird durch die euklidische Distanz $d(m,s)$ zwischen dem n-dimensionalen Gewichtsvektor m und dem jeweiligen Segmentvektor s auf diesem Knoten S(x/y) gebildet (Ausdruck 4.13). Die Ausdrücke m_i und s_i stehen für die Komponenten der Vektoren m und s in

[4]Knoten S(x/y) liegt auf der Segmentkarte in Spalte x und Zeile y

der *i*-ten Dimension.

$$QF_{x\,y}(s) = d(m,s) = \sqrt{\sum_{i=1}^{n} \left(m_i - s_i\right)^2} \tag{4.13}$$

Ist der QF $QF_{x\,y}(s)$ des derzeit betrachteten Segmentvektors *s* minimal $min\,[QF_{x\,y}(w)]$ auf diesem Knoten, d.h. ist das betrachtete Musiksegment sehr ähnlich bzw. am ähnlichsten dem Gewichtsvektor *m* des Neurons S(x/y), so wird für dieses Segment selbst 0.9 hinzugezählt und alle anderen Nachbarn gleichsam um $3 - 0.9 = 2.1$ erhöht (*w* stellt den Segmentvektor mit dem minimalsten QF auf diesem Knoten dar, um das deutlich zu zeigen wird er immer wie folgendermaßen dargestellt: $min\,[QF_{x\,y}(w)]$). Das bedeutet, zu jedem der Nachbarn wird $2.1/9 = 0.2333$ hinzugezählt, was in etwa $1/4$ entspricht. Übersteigt der QF ein gewisses Niveau (ist er siebenmal so groß wie das Minimum auf diesem Knoten), so wird eine Gleichgewichtung vorgenommen. Der Wert des Knotens und der aller Nachbarn wird um 0.3 erhöht. Die genaue Berechnung der Attribute $u_{x\,y}(v)$ und deren Nachbarn $u_{x\pm1\;y\mp1}(v)$ des neuen Vektors *v* für die Musikstückkarte ist in Rechnung 4.14 und 4.15 gezeigt.

$$u_{x\,y}(v) = 1 - \frac{\frac{QF_{x\,y}(s)}{min\left[QF_{x\,y}(w)\right]}}{10} \tag{4.14}$$

$$u_{x\pm1\;y\mp1}(v) = \frac{3 - u_{x\,y}(v)}{9} \tag{4.15}$$

$u_{x\,y}(v)$ bezeichnet jenen Wert der auf die Komponente $l_{x\,y}$ des Vektors *v* (Ausdruck 4.11) aufsummiert wird. Um die Komponente $l_{x\,y}$ des Vektors *v* zu bilden müssen alle $u_{x\,y}(v)$ aufsummiert werden. $u_{x\pm1\;y\mp1}(v)$ steht für die jeweiligen Nachbarn und repräsentiert somit expandiert: $u_{x+1\;y-1}(v)$, $u_{x+1\;y}(v)$, $u_{x+1\;y+1}(v)$, $u_{x\;y-1}(v)$, $u_{x\;y+1}(v)$, $u_{x-1\;y+1}(v)$, $u_{x-1\;y}(v)$, $u_{x-1\;y-1}(v)$.

Zum besseren Verständnis folgt ein weiteres Beispiel: Stück *A* besitzt folgende Verteilung auf einer 2×4 SOM: 1 Segment auf S(0/1) mit QF 4 und 2 Segmente auf S(1/3) mit QF 1 und 2. Stück *B* besitzt ein Segment auf S(0/1) mit QF 1, ein weiteres auf S(0/2) mit QF 3 und noch eines auf S(1/3) mit QF 9. Zuerst wird der minimale QF für jeden Knoten berechnet, Rechnung 4.16 .

$$min\,[QF_{x\,y}(w)] = \begin{pmatrix} 0 & 1 & 3 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.16}$$

Nun wird Vektor *b* mittels Formel 4.14 und 4.15 berechnet.

Für S(0/1):

$$u_{0\;1}(b) = 1 - \frac{\frac{1}{1}}{10} = 0.9$$

$$u_{0\pm1\ 1\mp1}(b) = \frac{3-0.9}{9} = 0.2333 \tag{4.17}$$

Wie aus Rechnung 4.17 hervorgeht, wird zum Attribut (0/1) im Vektor für die Musikstückkarte 0.9 und für die Nachbarn (0/0), (1/0), (1/1), (1/2), (0/2) je 0.2333 hinzugezählt.

Für S(0/2):

$$\begin{aligned} u_{0\ 2}(b) &= 1 - \frac{\frac{3}{3}}{10} = 0.9 \\ u_{0\pm1\ 2\mp1}(b) &= \frac{3-0.9}{9} = 0.2333 \end{aligned} \tag{4.18}$$

Wie aus Rechnung 4.18 hervorgeht, wird zum Attribut (0/2) im Vektor für die Musikstückkarte 0.9 und für die Nachbarn (0/1), (1/1), (1/2), (1/3), (0/3) je 0.2333 hinzugezählt.

Für S(1/3):

$$\begin{aligned} u_{1\ 3}(b) &= 1 - \frac{\frac{9}{1}}{10} = 0.1 \\ u_{1\pm1\ 3\mp1}(b) &= \frac{3-0.1}{9} = 0.3222 \end{aligned} \tag{4.19}$$

Rechnung 4.19 hätte man nicht durchführen müssen, da der QF siebenmal größer ist als das Minimum. Die Vektorkomponenten der Knoten (1/3), (1/2), (0/2), (0/3) werden daher alle um 0.3 erhöht.

Vektor *b* für die Musikstückkarte lautet somit wie im Ausdruck 4.20.

$$u(b) = \begin{pmatrix} 0.2333 & 1.1333 & 1.4333 & 0.5333 \\ 0.2333 & 0.4666 & 0.7666 & 0.5333 \end{pmatrix} \tag{4.20}$$

Um die Länge der Stücke zu nivellieren werden die resultierenden Vektoren auf Länge 1 normiert.

Nun wird wiederum eine SOM mit diesen Vektoren trainiert. Das Resultat ist eine Karte, auf der jedes Musikstück nur genau einmal vorkommt. Auf der Segmentkarte ähnlich verteilte Musikstücke liegen in benachbarten Bereichen, während Stücke mit einer völlig anderen Verteilung in anderen Bereichen zu finden sind. Natürlich werden nicht nur Stücke deren Segmente alle auf einem Cluster liegen auf einen gemeinsamen Bereich gemappt, sondern auch solche, die ähnliche Sprungverhalten zeigen, z.B. zwischen zwei Clustern hin und her springen (Strophe und Refrain).

4.5. Zusammenfassung

In diesem Kapitel wurde detailliert das Verfahren zur Musikklassifikation mittels einer SOM beschrieben. Dazu wurden aus dem Mediaplayer XMMS Klangspektren entnommen. Aus diesen Klangspektren wurden Frequenzbereiche extrahiert und zu fünfsekundigen Segmenten zusammengefasst. Die einzelnen Frequenzbänder wurden über die Zeitachse interpoliert um äquidistante Stützstellen für eine darauf folgende FFT zu erhalten. Aus den erhaltenen Fourierkoeffizienten wurden Eingabevektoren für die SOM erstellt und daraus eine Karte von Segmenten von Musikstücken erstellt. Ähnlich klingende Segmente liegen auf der Karte in nachbarschaftlichen Bereichen, während divergierend klingende Segmente weiter voneinander entfernt liegen. Um statt Segmente von Musikstücken ganze Musikstücke zu gruppieren, wurde aus der Segmentkarte unter Berücksichtigung des Quantisierungsfehler (QF) neuerlich ein Vektorfile für einen weiteren SOM-Lauf generiert. Die daraus gewonnene Musikstückkarte enthält für jedes Musikstück genau einen Eintrag. Ähnlich auf der Segmentkarte verteilte Musikstücke liegen in benachbarten Bereichen.

5. Experimente

Das in Kapitel 4.2 vorgestellte Verfahren wird im nun folgenden Kapitel an zwei verschiedenen Musiksammlungen angewandt. Die **bekannte Musiksammlung** in Kapitel 5.1 stellt eine bunte Mischung aus Pop-Musik der letzten Jahrzehnte, bekannter klassischer Musik, sowie aus Hits der letzten Jahre dar. Jeder Leser sollte zumindest ein paar Lieder am Titel erkennen und sich an die Melodie erinnern. Durch den Bekanntheitsgrad der Lieder kann sich jeder Leser selbst ein Bild von der Funktionsweise und den Ergebnissen des Verfahrens machen. Die zweite Kollektion, die **freie Musiksammlung** in Kapitel 5.2, besteht aus eher unbekannten Titeln, die aus dem Internet frei, ohne Abführen von zusätzlichen Abgaben, heruntergeladen werden können. Diese Sammlung zeigt das eigentliche Ziel des Verfahrens auf. Ohne die Lieder manuell vorzusortieren bzw. ohne Angabe von Metainformationen, werden ähnlich klingende Musikstücke indentifiziert und nahe beieinander gruppiert.

5.1. Bekannte Musiksammlung

Die bekannte Musiksammlung besteht aus insgesamt 230 Lieder und hat eine Gesamtspielzeit von über 14 Stunden. Unter den Stücken befindet sich Austro-Pop (*I am from Austria (Rainhard Fendrich)*), Sommer Hits (*Macarena(Los Del Rio), Around the world (ATC)*), Klassik (*Donauwalzer (Strauß)*), Musicals (*Les Miserables, Cabaret*), Filmmusik aus *Zurück in die Zukunft, Drei Musketiere, Jurassic Park, Robin Hood*, Pop-Musik von *Madonna, Cher, Beatles, Sting, Bryan Adams, Tina Turner, Jennifer Lopez, The Corrs, Coolio, Spice Girls, Take That, Robbie Williams* und Klassiker von *Chuck Berry, Louis Armstrong, Elvis, Fats Domino, Ella Fitzgerald, Cat Stevens*, um nur einige zu nennen. Tabelle 5.1 zeigt einen Ausschnitt aus der bekannten Musiksammlung. Die gesamte Tabelle befindet sich im Anhang A .

Tabelle 5.1.: Auszug aus der bekannten Musiksammlung

Karte	Titel	Version
adagio	Adagio aus Klarinettenkonzert	Mozart
aileinduinn	Ailein Duinn	Volkslied
allforlove	All For Love	Brain Adams, Rod Steward, Sting
allymcbeal	Searching My Soul	Vonda Shepard
angels	Angels	Robbie Williams
anywhereis	Anywhere Is	Enya
avemaria	Ave Maria	Schubert
badblood	Bad Blood	Ministry
beethoven	5.Sinfonie 1.Satz	Beethoven
believe	Believe	Cher
branden	Brandenburgische Konzert 2 Andante	Bach
breathless	Breathless	The Corrs
everythingido	Everything I Do	Bryan Adams
fatherandson	Father And Son	Cat Stevens
fuguedminor	Toccata und Fuge in D-Moll	Bach
funeral	Begräbnismarsch (Piano Sonate 2)	Chopin
gowest	Go West	Pet Shop Boys
ironic	Ironic	Alanis Morissette
itsinhiskiss	It’s in his kiss	Vonda Shepard
macarena	Macarena	Los Del Rio
morgen	Morgenstimmung	Edvard Grieg
morningbroken	Morning Has Broken	Cat Stevens
readmymindstars	If You Could Read My Mind	Stars on 54
sexbomb	Sexbomb	Tom Jones
tannenbaum	Oh, Tannenbaum	Volkslied
tellhim	Tell Him	Vonda Shepard
therose	The Rose	Bette Midler
torn	Torn	Natalie Imbruglia
whenyousay	When you say nothing at all	Ronan Keating
wildwildwest	Wild Wild West	Will Smith

5.1.1. Training

Nach der Verarbeitung der Lieder durch XMMS verblieben 230 Dateien mit insgesamt 460 MB (Bzip2)[1] zur weiteren Verarbeitung. Im ersten Schritt erfolgte die Entfernung von stillen Passagen am Anfang und Ende der Datei. Darauf folgend wurden die Lieder in die einzelnen Frequenzkanäle aufgesplittet und jeder fünfzehnte beginnend mit Kanal 1 ausgewählt. Das resultierte in 17 Dateien für jedes Lied, somit 3910 Dateien mit insgesamt 375 MB (Bzip2). Nach der Zerlegung in fünfsekundige Segmente und Auswahl jedes Zweiten erhält man 85374 Dateien, mit insgesamt 254 MB. Lagrange-Interpolation und FFT werden direkt hintereinander ausgeführt. Nach Verschiebung der Werte in den positiven Bereich und Normierung der Vektoren erhält man ein 444 MB großes Vektorfile. Durch die Zusammenfassung der 17 Frequenzbereiche pro Segment erhält man 5022 Vektoren, jeder Vektor mit 4352 Dimensionen.

Nach etwa zweistündigem Training[2] über insgesamt 15066 Iterationen einer 22×22 Karte, beginnend mit Lernrate 0.8, erhält man die Segmentkarte. Aus der Segmentkarte wird mittels den in Kapitel 4.4 beschriebenen Verfahren ein neues Vektorfile zum Training der Musikstückkarte vorbereitet.

Durch die Dimension der Segmentkarte ergibt sich die Dimension des neuen Vektorfiles. Das Vektorfile der Musikstückkarte besteht aus 230 484-dimensionalen Vektoren und hat eine Größe von weniger als einem Megabyte (MB). Nach einem wenige Minuten dauerndem Training erhält man eine 10×10 Karte – die Musikstückkarte.

5.1.2. Evaluierung

Abbildung 5.1 auf Seite 41 zeigt die gesamte Musikstückkarte in einer Abbildung. Mit einigen Worten beschrieben zeigt sich auf der Musikstückkarte folgendes Bild: Unten links formiert sich ein großer Klassik Cluster (*Brandenburgisches Konzert, Eine kleine Nachtmusik, Mondscheinsonate*), in der rechten unteren Ecke findet sich rockige Musik (*Timewarp, Good Golly meets Molly*). Mitte links auf der Karte ist ein Hardrock Knoten (*Bad Blood, Du hast*). Je weiter man aufwärts schaut, desto näher kommt man an Disco-Musik, endend in der Mitte ganz oben mit *Go West, Rock DJ, Sexbomb*. Ruhige, langsame Musik ist in der rechten oberen Ecke zu finden (*Everybody loves Somebody, Love me Tender, White Christmas*). Natürlich gibt es auch auf dieser Karte sehr saubere Knoten, die nur ein einziges Genre beinhalten,

[1] Bei derart gekennzeichneten Dateigrößen wurden die einzelnen Dateien um Speicherplatz einzusparen mit Bzip2 komprimiert

[2] Der Hauptspeicherbedarf liegt bei etwa 800 MB und die Prozessorlast bei über 90 Prozent

(0/0)	(1/0)	(2/0)	(3/0)	(4/0)	(5/0)	(6/0)	(7/0)	(8/0)	(9/0)
wonderfulworld	aintnosunshine anywhereis aroundtheworld madlydeeply risingsun submarines summercity	breathless goldeneye torn youlearn	babycomback cocojambo latinolover mambofive mmmbop tellhim	allforlove believe duellingviolins feliznavidad fuerstenfeld gowest rockdj saymyname	goodbye radio sexbomb	badboy breathaway dschinghiskhan everlastinglove lemontree missathing moonlightshadow seeyouwhen	conga feeling griechischwein lastchristmas talkaboutlove unbreakmyheart	crashboombang icouldfly ifyoubelieve summerdreaming	californiadream deepisyourlove ifonly lovesombodysomt lovsisintheair missingyou peggysue philadelphia piubellacosa vivaforever
(0/1)	(1/1)	(2/1)	(3/1)	(4/1)	(5/1)	(6/1)	(7/1)	(8/1)	(9/1)
wonderland		itsinhiskiss	allymcbeal	angels shoopshoopsong			sing	dayinparadise	beautyandbeast drummerboy kaktus lovemetender stormsinafrica whitechristmas
(0/2)	(1/2)	(2/2)	(3/2)	(4/2)	(5/2)	(6/2)	(7/2)	(8/2)	(9/2)
grossvater surfusa tiffany togetheragain	bongobong laymedown	readmymindstars	imf kissfromarose lovedwoman			ironic		austria blueberry breakfree gaelicreels roadtohell	fatherandson geldumagstmi morningbroken readmymindlight
(0/3)	(1/3)	(2/3)	(3/3)	(4/3)	(5/3)	(6/3)	(7/3)	(8/3)	(9/3)
alltoyou help letsgetloud sanfrancisco singalongsong		whenyousay	nahnehnah	jailhouserock themangotree		macarena wildwildwest	backforgood myloveyourlove		bebopalula poweroflove
(0/4)	(1/4)	(2/4)	(3/4)	(4/4)	(5/4)	(6/4)	(7/4)	(8/4)	(9/4)
badblood duhast joeschau rockisdead running she yesterdbeatles				mindfiels sunshineoflife		boogiewoogie	whatsup	bundeshymne herzilein icanfly nocturne	allegromolto cheektocheek donttalkanymore lastdance mcgee pinkpanther yougotit
(0/5)	(1/5)	(2/5)	(3/5)	(4/5)	(5/5)	(6/5)	(7/5)	(8/5)	(9/5)
				starwars	threetimesalady	eternalflame	zarathustra	lorddance	africa drive rocknrollkids schneib takefive yesterday
(0/6)	(1/6)	(2/6)	(3/6)	(4/6)	(5/6)	(6/6)	(7/6)	(8/6)	(9/6)
				americanpie pachelbl radetzkymarsch		friend holdon		fromnewyorktola movingonup	timeaftertime
(0/7)	(1/7)	(2/7)	(3/7)	(4/7)	(5/7)	(6/7)	(7/7)	(8/7)	(9/7)
	morgen mountainking			feellovetonight		firsttime	zillertaler		frozen fuguedminor tell
(0/8)	(1/8)	(2/8)	(3/8)	(4/8)	(5/8)	(6/8)	(7/8)	(8/8)	(9/8)
funeral merry mond		schwan therose	beethoven leavingport	memory walkingincohn	myway	walkininmemphis		alice	life rainbow thecircleoflife veronika
(0/9)	(1/9)	(2/9)	(3/9)	(4/9)	(5/9)	(6/9)	(7/9)	(8/9)	(9/9)
aileinduinn air branden flute kidscene mazurka minuet nachtmusik zapfenstreich	adagio avemaria forelle walzer	elvira everythingido giubba tritschtratsch	fortuna jurassicpark schindler	aufderhoeh goodmornblues onlyyou panoptikum	bahnfrei bigworld donau indy kiss revolution saints tannenbaum	cabaret dancingqueen diamonds future ihavenothing matilda newyork onedaymore requiem riverdance shakespeare sport	adiemus distance johnnyb myheartwillgoon	anything heavensdoor party supertrouper whatsawoman	dingdong foreveryoung girls goodgolly manicmonday missyoucrazy timewarp

Abbildung 5.1.: gesamte Musikstückkarte

wie auch weniger gelungene Klassifizierungen, wo auf einem Knoten ein Durcheinander von Musikrichtungen herrscht. Um solche Knoten zu verstehen lohnt sich ein Blick auf die Segmentkarte. Tabelle 5.2 zeigt einen Klassikcluster aus der Segmentkarte. Trotz der überwiegenden klassischen Musik (*adagio, air, avemaria, beethoven, branden, elvira, flute, forelle, fortuna, funeral, indy, leavingport, lorddance, mazurka, merry, mond, morgen, mountainking, nachtmusik, nocturne, pachelbl, radetzkymarsch, riverdance, schindler, schwan, shakespeare, tannenbaum, walzer*) in diesem Cluster, liegen hier auch Segmente wie *crashboombang.3*[3] (S(13/21), Achter von oben), die zwar sicher keine klassischen Titel sind, die jedoch zumindest im jeweiligen Segment ähnliche Klangdynamiken besitzen. Viele Intros, erkennbar an den niedrigen Segmentnummern, bzw. Fade-outs (hohe Segmentnummern) liegen meist zusammen mit klassischen Stücken. Dieser Effekt ist durch die völlig andere, meist sehr eingeschränkte (z.B. nur Gitarre), Instrumentierung während dieser Passagen erklärbar. Dieses Mapping schlägt sich dann meist auch im Quantisierungsfehler (QF) der jeweiligen Segmente nieder. Beispielsweise besitzt auf S(13/21) *tannenbaum.1* und *morgen.15* den niedrigsten QF, während hingegen der von *everythingido.67*, *whenyousay.37* und *readmymindstars.3* etwa siebenmal so groß ist.

Im folgenden werden nun exemplarisch einige gute Knoten auf der Musikstückkarte behandelt und Begründungen für deren Lage gegeben.

Zuerst wird links unten der große Klassikknoten betrachtet. Abbildung 5.2 zeigt den entsprechenden Kartenbereich nochmal vergrößert.

Knoten (0/8)[4] enthält den *Begräbnismarsch (Chopin)* sowie die *Mondscheinsonate (Beethoven)* und *Merry Peasant (Schumann)*. Alle drei sind sehr ruhige Klavierstücke und liegen auf der Segmentkarte in nahezu den selben Bereichen. Die Segmente aller drei Lieder liegen hauptsächlich, wie in Tabelle 5.2 ersichtlich, auf S(13/21) und S(21/21). Knoten (0/9) enthält fast nur Klassikstücke und ist auch sehr sauber. Unter anderem finden sich hier *Air* und das *Brandenburgische Konzert No.2* von *Bach*, sowie abermals je ein Stück von *Schumann*, *Chopin* und *Mozart* (*Fremde Länder und Menschen (Schumann), Mazurka in a-Moll (Chopin), Eine kleine Nachtmusik (Mozart)*). Alle erwähnten Stücke sind wiederum sehr ruhig und haben starke Anteile an den beiden oben bereits erwähnten Koordinaten der Segmentkarte. Das Lied *Ailein Duinn*, ein schottisches Volkslied, scheint auf den ersten Blick nicht zu den anderen zu passen. Doch bei genauerer Betrachtung entpuppt sich dieses Stück als

[3] *.3* bezeichnet die Segmentnummer, die genaue Erläuterung befindet sich im Anhang C

[4] Knoten (x/y) liegt auf der Musikstückkarte in Spalte x und Zeile y

Tabelle 5.2.: Ein Klassikcluster aus der Segmentkarte bei S(12/21) und S(13/21)

S(12/21)	S(13/21)
anywhereis.43	adagio.79
bigworld.3	air.25
diamonds.9	air.27
elvira.39	avemaria.19
everythingido.35	beethoven.19
fatherandson.5	beethoven.27
flute.35	branden.23
forelle.43	crashboombang.3
funeral.5	dayinparadise.1
indy.25	elvira.43
ironic.3	elvira.45
leavingport.37	everythingido.33
lorddance.3	everythingido.67
memory.3	fatherandson.3
morgen.23	feellovetonight.5
mountainking.17	flute.31
nachtmusik.37	forelle.15
nocturne.9	forelle.33
pachelbl.19	forelle.35
pachelbl.21	fortuna.9
pachelbl.33	fortuna.43
radetzkymarsch.5	funeral.7
shakespeare.19	funeral.21
walzer.15	goodmornblues.21
	heavensdoor.1
	holdon.5
	holdon.49
	indy.33
	leavingport.11
	mazurka.23
	merry.1
	mond.3
	mond.17
	mond.25
	mond.35
	mond.49
	morgen.15
	mountainking.7
	nachtmusik.55
	nocturne.1
	nocturne.35
	pachelbl.25
	radetzkymarsch.13
	readmymindstars.3
	riverdance.1
	schindler.43
	schwan.7
	summercity.25
	tannenbaum.1
	therose.9
	walkingincohn.47
	whenyousay.37

(0/8)	(1/8)	(2/8)	(3/8)
funeral merry mond		schwan therose	beethoven leavingport
(0/9)	**(1/9)**	**(2/9)**	**(3/9)**
aileinduinn air branden flute kidscene mazurka minuet nachtmusik zapfenstreich	adagio avemaria forelle walzer	elvira everythingido giubba tritschtratsch	fortuna jurassicpark schindler

Abbildung 5.2.: Klassikknoten links unten

sehr ruhige, bedächtige Ballade, die von einer einzigen Frauenstimme mit Begleitung einer Harfe und Geige vorgetragen wird. Auf dem Nachbarknoten (1/9) finden sich unter anderem zwei Stücke von *Schubert* (*Ave Maria, Themen und Variationen der Forelle*) sowie abermals ein Stück von Mozart (*Klarinettenkonzert - Adagio*). Auch hier zeigt die Segmentkarte aller dieser Stücke starke Anteile im rechten unteren Sektor. Knoten (2/9) enthält nun neben einem Stück von *Mozart* auch dynamischere Musik, wie die *Tritsch Tratsch Polka (Strauß)*. Auch zwei Stücke mit Gesang finden sich hier *Vesti La Giubba (Domingo)* sowie *Everything I do (Bryan Adams)*. Natürlich ist *Everything I do* keine klassische Musik, doch spielt hier die meiste Zeit über ein Orchester. Die Segmentkarte weist zwar auch hier noch eine Verteilung im linken unteren Eck auf, doch zeigt sich eine breiter Streuung, vor allem beim Stück von *Bryan Adams*, was aber verständlich ist, zumal es auch viel länger als die anderen ist (über 6 Minuten). Noch interessant in diesem Cluster ist Knoten (3/9), der zwei Stücke von *John Williams* enthält (Hauptthema aus *Jurassic Park* und *Schindlers Liste*). Auch Knoten (2/8) ist erwähnenswert, da hier ein Stück aus *Schwanensee (Tschaikowsky)* direkt bei *The Rose (Bette Midler)* liegt. *The Rose* wird fast nur von Klavier, Geige und einer sehr ruhigen Stimme getragen. Alles in allem kann gesagt werden, dass der Klassikknoten sehr sauber ist. Dieser Cluster streicht die Topologieerhalung der SOM gut hervor. Beginnend von links, mit sehr ruhiger Musik, steigt die Dynamik nach rechts gehend langsam an. Die SOM bildet diesen Verlauf der Realität richtig ab.

Als nächstes wird Knoten (8/6) in Abbildung 5.3 betrachtet.

Sowohl *From New York to L.A. (Stephanie McKay)* als auch *Moving On Up (M-People)* sind eher schnellere Stücke, die man als Disco-Musik bezeichnen kann. Diagonal darunter

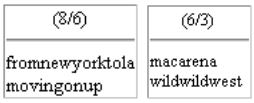

Abbildung 5.3.: Zwei Knoten mit Disco-Musik

Abbildung 5.4.: Ruhige Musik, rechts Mitte

auf (9/7) ist wieder ein Beispiel für einen Pop-Song, der zu klassischen Stücken gefallen ist. *Frozen (Madonna)* wird fast nur von Geigen getragen, was von der klanglichen Zuordnung Sinn ergibt. Abbildung 5.3 zeigt ebenfalls Knoten (6/3) mit *Macarena (Los Del Rio)* und *Wild Wild West (Will Smith)*, die ebenfalls moderne Disco-Musik darstellen.

Knoten (9/2), vergrößert in Abbildung 5.4, zeigt eine Kollektion ruhiger Lieder. Zwei Lieder von *Cat Stevens* (*Father And Son*, *Morning Has Broken*) sind neben einem Lied von *Ludwig Hirsch (Gel Du Magst Mi)* und *If You Could Read My Mind (Gordon Lightfoot)* zu finden. Alle vier sind Balladen, getragen von Stimme und meist nur einer Gitarre.

Gleich darüber befindet sich ein Cluster aus großteils ruhiger Musik (Abbildung 5.5). Hervorgehoben gehört hier insbesonders Knoten (9/1), der von *White Christmas* bis *Love me tender* sehr rein ist. Ob *Storms in Africa (Enya)* hier dazupasst, bleibt dem Geschmack des Lesers überlassen. Allerdings sollte man versuchen, *Enya's* Musik selbst in ein Genre einzuordnen, was wohl mehr als schwierig wird.

Im mittleren Bereich ganz links befindet sich ein Knoten, auf den fast alle Hardrock-Stücke gemappt wurden. Abbildung 5.6 zeigt *Bad Blood (Ministry)*, *Du Hast (Rammstein)* und *Rock is Dead (Marilyn Manson)* auf einen Knoten liegend. Obwohl *Bad Blood* sehr gut geclustert ist, mehr als die Hälfte des Liedes liegen auf einem einzigen Knoten in der Segmentkarte, ist der Knoten doch nicht besonders sauber. Die Segmente 5-13, 17, 21-33, 39-41, 47 und 51-55 liegen alle auf Knoten S(12/15) auf der Segmentkarte. Problematisch ist bei allen flotten extrem dynamischen Stücken, dass sich aus der Theorie der Fourierreihe (FR) heraus ergibt, dass die Koeffizienten stark schwanken und somit auch das aus der Interpolation resultierende Polynom. Noch einmal sei erwähnt, dass hier mit etwa 3 Prozent der zur Verfügung stehenden Datenmenge gearbeitet wird. Aussagen über Verbesserungen bei Erhöhung der Datenmenge

Abbildung 5.5.: Cluster mit ruhiger Musik, oben rechts

Abbildung 5.6.: Hardrock, links außen, Mitte

(2/0)	(3/0)	(4/0)	(5/0)
breathless goldeneye torn youlearn	babycomback cocojambo latinolover mambofive mmmbop tellhim	allforlove believe duellingviolins feliznavidad fuerstenfeld gowest rockdj saymyname	goodbye radio sexbomb
(2/1)	(3/1)	(4/1)	(5/1)
itsinhiskiss	allymcbeal	angels shoopshoopsong	

Abbildung 5.7.: Flotte Musik oben Mitte

können seriös nicht beantwortet werden, da die Implementierung den Rahmen dieser Arbeit sprengen würde.

Noch ein letzter Cluster wird in Abbildung 5.7 gezeigt. Hier findet man *Believe (Cher)*, *Go West (Pet Shop Boys)* und *Rock DJ (Robbie Williams)* auf (4/0) gleich neben *Sexbomb (Tom Jones)* auf (5/0) und *Mambo No 5 (Lou Bega)* auf (2/0). Auch ein sehr gutes Mapping ist *Torn (Natalie Imbruglia)* mit *Breathless (The Corrs)* auf (2/0). Ganz interessant ist die Anordnung von einer bestimmten Sängerin auf diesem Cluster. Insgesamt befinden sich drei Lieder (*Searching My Soul, Tell Him, It's in his kiss*) von *Vonda Shepard* in diesem Cluster. Alle drei sind auf einer CD und mit ein- und derselben Gruppe und Instrumentierung gespielt. Hinzu kommt noch, dass *It's in his kiss* von *Cher's Shoop Shoop Song* gecovert ist, der ebenfalls in diesem Cluster zu finden ist.

Hier soll jetzt ein Schlussstrich unter die Clusteranalyse gezogen werden. Jeder Leser ist natürlich aufgefordert die Karte länger zu betrachten, da sicher noch viele interessante Mappings zu finden sind. Allerdings gibt es auch noch auf der Segmentkarte einiges zu finden.

5.1.3. Ein weiterer Blick auf die Segmentkarte

Bei genauer Betrachtung der Segmentkarte fällt auf, dass die meisten Lieder, vor allem Pop-Musik, nicht allzu gut geclustert sind. Klassische Musik hingegen liegt meist sehr sauber geclustert. Dies gilt natürlich dann auch für die Musikkarte, da sie auf der Verteilung der Segmente auf der Segmentkarte beruht. Untersucht man den Grund, so stößt man bei Betrachtung der entsprechenden Teile der Segmentkarte sehr schnell auf Lösungen. Vor allem der Anfang und das Ende der Lieder seien hier erwähnt. Heutzutage ist es bei Pop-Musik üblich, eine Intro (so wird der Beginn eines Liedes genannt) zu schreiben, die sich völlig anders als der Rest

des Liedes anhören kann. Beispielsweise ist *See You When You Get There (Coolio)* ein Rap-Song, in dem innerhalb der ersten 25 Sekunden ausschließlich Violinen spielen. Natürlich wurde somit das erste und das dritte Segment in den Klassikcluster gemapped, während sonst kein weiteres Segment in diesem Sektor liegt. Auch der Schluss eines Liedes klingt bei vielen Liedern anders als der Hauptteil des Liedes selbst, wodurch sich hier wieder ein Problem für die nachfolgende Musikstückkarte ergibt. Ist das Lied zusätzlich nur von kurzer Länge, so bleiben durch die Reduktion der Segmente nur mehr sehr wenige Segmente zur Klassifikation zurück.

Eine weitere Möglichkeit besteht darin, dass sich Refrain und Strophe unterscheiden. *Ironic (Alanis Morissette)* beispielsweise hat sowohl eine ruhige Intro, wie auch einen langsamen Ausgang; die Strophen sind eher ruhig gehalten, während der Refrain ziemlich rockig ist. Daraus resultiert, dass die Segmente des Liedes über die ganze Karte verstreut sind. Betrachtet man die Segmente allerdings einzeln für sich und jede fünfsekundige Passage getrennt, so bekommt man einen Einblick in den Aufbau des Liedes. Tabelle 5.3 zeigt in der ersten Spalte den Index des betrachteten fünfsekundigen Segments. Anhang C gibt eine kleine Umrechnungshilfe für die Indizes. Die Spalte „Lied“ zeigt das, nach Meinung zweier Testpersonen, „Genre“ des fünfsekundigen Liedes selbst an, während hingegen Spalte „Segment“ das „Genre“ nach seiner Lage auf der Segmentkarte angibt. Um die Tabelle einfach zu halten wurden nur vier Buchstaben zur Kodierung verwendet: U steht für eine ruhige, S für schnellere, R für rockige und M für Übergänge zwischen den Passagen.

Wie man aus Tabelle 5.3 sieht, ist es nicht unbedingt nötig das Lied zu hören um dessen Art zu erkennen. Durch diesen „Nebeneffekt“ erhält man quasi Einblick in den Liedaufbau an sich. Wie man sieht, vereinigt ein einziges Lied mehrere „Genres“ in sich. Diese Multi-Charakteristik zeigt auf, wieso eindeutige Clusterung schwer fällt.

Dass die Self-Organizing Map (SOM) allerdings auch Verbindungen aufzeigt, die man sonst nicht sieht, zeigt folgendes Beispiel. Die zwei Lieder *Mission Impossible (Adam Clayton, Larry Mullen)* und *Macarena (Los Del Rio)* haben auf den ersten Blick wenig gemeinsam. Trotzdem sind auf der Segmentkarte zweimal Überkreuzungen zu finden. Bei genauem Hinhören fällt auf, dass das Bassmotiv, das die ganze Zeit im Hintergrund zu hören ist, dasselbe ist. An einer bestimmten Stelle (*Imf.9, macarena.23*), kann man das Bassmotiv sehr deutlich erkennen. Zu diesem Zeitpunkt sind nur wenige andere Instrumente zu hören. Hier zeigen sich ganz klar die Stärken der SOM, da sie durch ihre Architektur diese Beziehung aufspürt und auf der Karte richtig abbilden kann.

Tabelle 5.3.: Genreanalyse von Ironic

Index	Lied	Segment
1	U	U
3	U	U
5	U	U
7	M	R
9	R	M
11	R	R
13	S	S
15	S	S
17	M	U
19	R	R
21	R	R
23	U	M
25	U	U
27	U	U
29	U	U
31	U	U
33	M	M
35	R	M
37	R	R
39	U	U
41	U	U
43	U	U

Tabelle 5.4.: Auszug aus der freien Musiksammlung

Karte	Titel	Interpret	Genre
abbey	Abby of Metal	Paralysing Prophecy	Metal
acidmachine	Acid Machine	Ritual	Metal
cosmotron	Cosmotron	Stefan Poiss	Electronic
falling	Falling	Stefan Poiss	Electronic
galaxies	Galaxies	Stefan Poiss	Electronic
mindkiller	Mindkiller	Stefan Poiss	Electronic
moyo	Moyo	Stefan Poiss	Electronic
parsec	Parsec Main Theme	Stefan Poiss	Electronic
pushthatthing	Push that thing	Dave Aude	Electronic
shelter	Shelter	AV Shelter	Electronic
thesedays	Some of these days	Paul Burch and the WPA Ballclub	Country

5.2. Freie Musiksammlung

Tabelle 5.4 zeigt die zweite bearbeitete Musiksammlung. Sie besteht aus so genannter „freier“ Musik, d.h. die Komponisten bzw. die Interpreten verzichten auf Tantiemen aus ihren Werken, wodurch die Musik ohne Abgabe von Gebühren frei verbreitet und abgespielt werden darf. Die Musik stammt zum überwiegenden Teil aus dem Internet [Aud00]. In Tabelle 5.4 ist eine Spalte „Genre“ zu finden. Diese Spalte zeigt das auf Audiogalaxy.com angegebene Genre.

5.2.1. Training

Die insgesamt 127 Lieder haben eine Gesamtspielzeit von über 8 Stunden. Nach der Verarbeitung der Lieder durch XMMS verblieben 127 Dateien mit insgesamt 293 MB (Bzip2) zur weiteren Verarbeitung. Nach der Entfernung von stillen Passagen am Anfang und Ende der Datei werden die Lieder in die einzelnen Frequenzkanäle aufgesplittet und jeder fünfzehnte beginnend mit Kanal 1 ausgewählt. Das resultierte in 17 Dateien für jedes Lied, somit 2159 Dateien mit insgesamt 226 MB (Bzip2). Nach der Zerlegung in fünfsekundige Segmente und Auswahl jedes Zweiten erhält man 50575 Dateien, mit insgesamt 157 MB. Lagrange-Interpolation und FFT werden direkt hintereinander ausgeführt. Nach Verschiebung der Werte in den positiven Bereich und Normierung der Vektoren erhält man ein 263 MB großes Vektorfile. Durch die Zusammenfassung der 17 Frequenzbereiche pro Segment erhält man 2975 Vektoren, jeder Vektor mit 4352 Dimensionen.

Nach etwa einstündigem Training über insgesamt 14875 Iterationen einer 18×18 Karte,

beginnend mit Lernrate 0.8, erhält man die Segmentkarte. Aus der Segmentkarte wird mittels den in Kapitel 4.4 beschriebenen Verfahren ein neues Vektorfile zum Training der Musikstückkarte vorbereitet.

Durch die Dimension der Segmentkarte ergibt sich die Dimension des neuen Vektorfiles. Das Vektorfile der Musikstückkarte besteht aus 127 324-dimensionalen Vektoren und hat eine Größe von weniger als einem Megabyte (MB). Nach wenige Minuten dauerndem Training erhält man eine 10×10 Karte – die Musikstückkarte.

5.2.2. Evaluierung

Abbildung 5.8 auf Seite 52 zeigt die gesamte Musikstückkarte in einer Abbildung. Es gilt wieder dasselbe, wie für die Karte aus Kapitel 5.1; Stücke, die nahe beieinander liegen, haben einen ähnlichen Klangverlauf. Unter Zuhilfenahme der Genrekennung von Audiogalaxy.com sieht diese Karte nicht so sauber aus wie die in Kapitel 5.1. Dies ist vor allem dadurch zu erklären, dass die Stücke generell von der Instrumentierung her sehr ähnlich sind. So fehlt beispielsweise die markante Trennung in einen Cluster mit Orchestermusik, die ein sehr charakteristisches und von den übrigen Genres stark unterschiedliches Klangbild aufweist. Dennoch ist eine Gruppierung in dominante Cluster, wie beispielsweise Hardrock (Abbildung 5.9) vorhanden.

Die Segmentkarte sieht so aus, dass fast alle Stücke auf einen insgesamt etwa 12 Knoten umfassenden Cluster gemappt wurden, der über mehrere Zeilen und Spalten geht. Die Musikstückkarte zeigt das auch ganz eindeutig durch die breite Streuung der einzelnen Musikstücke.

Abbildung 5.9 zeigt einen Cluster mit Hardrock-Musik. Knoten (8/0) zeigt drei Lieder, die aus sehr unterschiedlichen Passagen bestehen. Sowohl ruhigere Musik als auch sehr wilde Passagen sind in diesen Stücken gemischt. In Knoten (8/1) befindet sich unter anderem *Push that thing (Dave Aude)*, das laut Audiogalaxy.com als *elektronische Musik* ausgewiesen wird. Hier sind die Schwächen derartiger Klassifikationssysteme zu erkennen, da es sich zwar um elektronisch, sprich per Synthesizer, erzeugte Musik handelt, doch in Wirklichkeit ein Hardrock-Stück ist. Die SOM ließ sich nicht täuschen, da sie auf solche Metainformationen keinerlei Zugriff hat.

Abbildung 5.10 zeigt einen Knoten mit Rock-Musik. *Some of these days (Paul Burch)* wird zwar in Audiogalaxy.com als Country ausgewiesen, hat aber eindeutig ein, zumindest für Country-Musik, unübliches Schlagzeug, das eher zu Rock-Musik passt. Hört man auf Knoten (0/0) *Shelter (AV Shelter)* und *Galaxies (Stefan Poiss)* direkt hintereinander, so sind

(0/0)	(1/0)	(2/0)	(3/0)	(4/0)	(5/0)	(6/0)	(7/0)	(8/0)	(9/0)
acidmachine galaxies shelter	abbey beboxdual between thegrain	wishfuldrink	blonde nipples tripleoptics	aroundblock	tiny	thesword	quienerestu	blazinggowns burningofangels totheshores	americantrio booze breeds darkness righe showdown somethingvag
(0/1)	(1/1)	(2/1)	(3/1)	(4/1)	(5/1)	(6/1)	(7/1)	(8/1)	(9/1)
bravenewworld drfunk thesedays		woman	calicogirl	balloonrace			derailed	bullfrogtheme diane hookitup pushthatthing	
(0/2)	(1/2)	(2/2)	(3/2)	(4/2)	(5/2)	(6/2)	(7/2)	(8/2)	(9/2)
oracle sandyloam				handthatfeeds wootieboogie				critic torment	orient
(0/3)	(1/3)	(2/3)	(3/3)	(4/3)	(5/3)	(6/3)	(7/3)	(8/3)	(9/3)
animaanimale	blindmics	lockyouup meistamoneski	owlservice	badblood	hole	outofmyway timeoflife		thingslook	
(0/4)	(1/4)	(2/4)	(3/4)	(4/4)	(5/4)	(6/4)	(7/4)	(8/4)	(9/4)
camdentownrain snakes	elcordobes			reagansdead swinglow	beatemall october		crazytrain	rad	
(0/5)	(1/5)	(2/5)	(3/5)	(4/5)	(5/5)	(6/5)	(7/5)	(8/5)	(9/5)
backontrack gongdance hardside hurtsobad smile	callmecrazy	soupdujour		gorgeousfriend		unsound	discoguitar		cleopatra holocaust marylee strawberry
(0/6)	(1/6)	(2/6)	(3/6)	(4/6)	(5/6)	(6/6)	(7/6)	(8/6)	(9/6)
saramala	singingtheblues	hirake honolulunights leavemealone	doppler		nuttmeg	cantopraexu somethingwrong		pounds rebel	
(0/7)	(1/7)	(2/7)	(3/7)	(4/7)	(5/7)	(6/7)	(7/7)	(8/7)	(9/7)
mindkiller moyo			tuskegee			daddydizzy	lonely		borders wildcatboogie
(0/8)	(1/8)	(2/8)	(3/8)	(4/8)	(5/8)	(6/8)	(7/8)	(8/8)	(9/8)
cosmotron mambo sunnypasture			roulette		situation	dreamon fingernails	thecat	christopher milwaukee roots	countrygirl
(0/9)	(1/9)	(2/9)	(3/9)	(4/9)	(5/9)	(6/9)	(7/9)	(8/9)	(9/9)
binoculars falling parsec	bill newone outtaspace steelcitystrut	blancanina losbilbilicos tenderlu		dedicated hickorywind nufonia waterfront	bd montysgroove vsledzadja	aroundyou	rabenwald samba	goodleader	

Abbildung 5.8.: gesamte Musikstückkarte

(8/0)	(9/0)
blazinggowns burningofangels totheshores	americantrio booze breeds darkness righe showdown somethingvag
(8/1)	(9/1)
bullfrogtheme diane hookitup pushthatthing	
(8/2)	(9/2)
critic torment	orient

Abbildung 5.9.: Hardrock Knoten oben rechts

(0/0)	(1/0)
acidmachine galaxies shelter	abbey beboxdual between thegrain
(0/1)	(1/1)
bravenewworld drfunk thesedays	

Abbildung 5.10.: Rock Knoten oben links

Abbildung 5.11.: Synth-Musik Knoten unten links

Gemeinsamkeiten deutlich hörbar.

Abbildung 5.11 zeigt zum Schluss noch einen besonderen Cluster. *Mindkiller, Moyo* auf (0/7), sowie *Cosmotron* auf (0/8) sowie *Falling* und *Parsec Main Theme* auf (0/9) sind alle von *Stefan Poiss* für das Spiel PARSEC geschrieben.

Abschließend gesagt, scheint diese freie Musiksammlung nicht so gut geclustert zu sein, wie die in Kapitel 5.1 . Doch durch das Fehlen von Klassik und das starke Auftreten von Hardrock ähnlichen Stücken, die wiederum gut geclustert sind, ist die Musik nicht so vielfältig wie die der bekannten Musiksammlung. Die manuelle Klassifizierung von Audiogalaxy.com ist zumindest bei einigen Stücken recht fragwürdig, vor allem die Gruppe *Electronic* in Tabelle 5.4 lässt keinerlei Rückschlüsse auf das tatsächliche Genre zu.

6. Zusammenfassung und Ausblick

In dieser Arbeit wurde ein System zur automatischen Gruppierung von Musik vorgestellt. Ein Mediaplayer (XMMS) wurde verwendet, um aus Musikstücken Klangspektren zu extrahieren. Die extrahierten Klangspektren wurden zu fünfsekundigen Segmenten gruppiert. Einzelne Frequenzbänder wurden aus den Segmenten genommen und aus Werten eine Funktion interpoliert, die die Dynamik der Musik auf dem gewählten Frequenzband beschreibt. Die erzeugte Funktion wurde einer Fast Fouriertransformation (FFT) unterzogen, um wiederum Frequenzspektren der interpolierten Funktion zu erhalten. Diese neuen Spektren bilden die Merkmale der Eingabevektoren des verwendeten Neuronalen Netzes. Die Self-Organizing Map (SOM), ein unüberwacht lernendes, Topologie erhaltendes Neuronales Netz, erzeugt eine Abbildung aus einem hochdimensionalen Eingaberaum einen zweidimensionalen nachbarschaftsbeziehungserhaltenden Ausgaberaum – die Karte. Die SOM wird dazu verwendet, Segmente verschiedener Musikstücke, deren klangliche Dynamik ähnlich ist, zu gruppieren. Ähnlich klingende Segmente werden in benachbarten Regionen der Karte gefunden, während anders klingende Segmente in weiter auseinanderliegenden Bereichen liegen. Die erhaltene Segmentkarte wird zur Erstellung einer Musikstückkarte herangezogen, auf der anstatt vieler Segmente von Musikstücken jedes Musikstück nur mehr genau einmal vorkommt. Die Vektoren zur Erstellung der Musikstückkarte werden durch die Lage der Segmente auf der Segmentkarte bestimmt. Auf der erhaltenen Musikstückkarte sind Musikstücke mit ähnlicher klanglicher Dynamik nebeneinander abgebildet, während gänzlich anders klingende Musikstücke weiter voneinander entfernt sind.

Trotz der Vielfalt der in den Experimenten verwendeten Musikstücke sind die erhaltenen Ergebnisse sehr beachtlich. Eine Sammlung aus sehr bekannter Musik konnte erfolgreich geclustert werden. Orchestermusik mit ihrer sehr eigenen Charakeristik erwies sich als besonders gut zu gruppieren. Es wurden Ähnlichkeiten zwischen Liedern aufgezeigt, die über offensichtliche Verwandtschaften hinaus gehen. Die SOM eröffnete ebenfalls Einblick in den

Liedaufbau heutiger Pop-Stücke, sowie die starken Unterschiede zwischen Refrain und Strophen. Weiters wurde eine Sammlung von wenig bekannten Musikstücken untersucht, bei der die SOM trotz weniger markanter Merkmale Unterscheidungen vornahm. Lieder wurden nicht aufgrund ihres beim Vertreiber angegebenen „Genres“, sondern aufgrund ihrer Dynamik erkannt. Dies führte zu teilweise divergierenden Angaben, die die Schwächen manueller Klassifizierungen herausstreicht.

Obwohl die Ergebnisse schon sehr gut sind, können weitere Adaptierungen vorgenommen werden um die erhaltenen Resultate noch zu verbessern. Eine mögliche Verbesserung läge darin mehr Daten zu betrachten. Wie schon des Öfteren erwähnt, verwendet das derzeitige Verfahren ungefähr 3 Prozent der von XMMS gelieferten Daten. Experimente mit kleineren Testmengen ergaben, dass die Clusterung besser wird, je mehr Daten verwendet werden. Je näher die Frequenzbereiche beieinander liegen, im besten Fall aufeinander folgen, desto ähnlichere Muster ergeben sich auf der SOM. Diese Verbesserungsmöglichkeit scheitert derzeit noch am Implementierungsaufwand. Eine andere Möglichkeit besteht darin, an Parametern, wie etwa der Länge der Segmente mit allen Datensätzen zu experimentieren.

Eine Verbesserungsmöglichkeit würde sich auch bei der Quelle (XMMS) befinden. Erst durch die Unzulänglichkeit von XMMS Signale zu exakten Zeitpunkten zu liefern, muss eine Funktion interpoliert werden. Durch direkten Zugriff auf das Audiosignal kann man exakt gezeitete Samples entnehmen. Eine Interpolation ist dann hinfällig. Durch diesen Schritt ist das Verfahren allerdings vom Dateiformat abhängig, was eine wenig erfreuliche Einschränkung ist. Durch die Vielzahl der heute existierenden Konvertierungsprogramme für Audiodateiformate kann man natürlich vor Beginn des Verfahrens alle Musikstücke auf das für das Verfahren gewählte Dateiformat (z.B. MP3) umwandeln. Um direkten Zugriff auf die Audioquelle zu bekommen kann man die *MPEG maaate* Bibliothek benutzen [Maa01].

Eine weitere Möglichkeit zur Optimierung besteht darin, statt der Fast Fouriertransformation (FFT) eine andere Transformation zu wählen, die eher den Gegebenheiten von Musiksignalen gerecht wird, z.B. Wavelets [YS98]. Auch kann man in der Vorverarbeitung, wie viele andere auch (z.B. Feiten), psychoakustische Filter benutzen, wie sie Ellis entwickelt.

Eine rasche Vereinfachung wäre beispielsweise, die FFT Koeffizienten nach der Interpolation nicht direkt zur Vektorbildung heranzuziehen, sondern den Betrag der komplexen Zahlen zu bilden. Hierdurch verringert sich die Anzahl der betrachteten Merkmale um die Hälfte, und eine Nullpunktverschiebung ist nicht mehr erforderlich.

Auch kann man versuchen, an den fünfsekundigen Segmenten, statt jeden Frequenzbereich einzeln einer FFT zu unterziehen, eine zweidimensionale FFT durchzuführen, wodurch man

die Frequenzbereiche nicht mehr unabhängig voneinander betrachten würde.

Ein gänzlich anderer Versuch wäre die Arbeit von Dixon und Goto – das Beat-Tracking System – mit diesem Ansatz zu verknüpfen. Eine Möglichkeit bestünde darin, das Tempo auf die Frequenzkurve aufzumodulieren oder die Form des Vektors durch ein zusätzliches Attribut oder ein alle Attribute beeinflussendes Verfahren zu ändern.

Eine andere Möglichkeit, die Dynamik in der Frequenz herauszufinden, wäre einfach, die Beträge der Koeffizienten einer FFT aufzusummieren und daraus Schlüsse zu ziehen. Anstatt wie hier zum Schluss die Koeffizienten der FFT zu untersuchen, könnte man auch die Differenz der aufeinander folgenden Spektren interpolieren und klassifizieren, um nur die Änderung losgelöst von ihrem Grundwert zu betrachten.

Abschließend kann man sagen, dass dieses Gebiet, das noch bis vor kurzem im Dornröschenschlaf lag, inzwischen von allgemeinem Interesse ist. Durch die Vielzahl von Audiosignalen und Musikstücken, die es heute vor allem durch immer billigere Hardware gibt, ist es natürlich essentiell, Verfahren zu finden, die ohne manuellen Eingriff inhaltsbasierte Gruppierungen vornehmen. Trotz der in dieser Arbeit präsentierten Ergebnisse ist es eher unwahrscheinlich, dass ein einzelner Ansatz, wie etwa hier die Dynamik der Spektren zu verfolgen, ein endgültiges Lösungssystem erbringt. Viel mehr die Verknüpfung aller Verfahren (Spektralanalyse, Ohrmodell, Dynamik der Spektren, Beat, Melodieanalyse) ist vermutlich der Schlüssel zur erfolgreichen Applikation.

7. Danksagung

Als erstes möchte ich meinem Betreuer Andi Rauber danken. Er stand mir, wann immer ich Anleitung bedurfte, mit Rat und Tat zur Seite. Obwohl er an notorischem Zeitmangel leidet, kam ich mir bei ihm niemals als einer von vielen, sondern als wichtigster von wenigen vor. Weiters danke ich Dieter Merkl für seine Betreuung.

Mein Dank gebührt auch Winfried Auzinger, der mir schnell und unkompliziert Informationen über Interpolation und Fouriertransformation gab. Für die Übersendung von Material möchte ich auch Tamas Ungvary danken. Für die Diskussion der mathematischen Algorithmen stellte sich mein langjähriger Freund und Kollege Milan Andjelkovic zur Verfügung. Stefan Poiss gab die Erlaubnis, seine Musik für die Experimente in der freien Musiksammlung zu verwenden. Mit Michael Dörfler hatte ich eine wertvolle Diskussion vor Beginn der Arbeit, die ihr die richtige Richtung gegeben hat.

Da diese Arbeit den Abschluss meiner universitären Laufbahn darstellt, möchte ich die Gelegenheit nutzen, auch weiteren Menschen Dank zu sagen, die zwar nichts mit dieser Arbeit zu tun hatten, mich bis hierhin jedoch geführt haben.

Als erstes meinen Eltern, die mir nicht nur durch ihre Finanzierung ein sorgenfreies Studium ermöglichten. Meinem Bruder Gregor, da er versuchte einen Schritt vor mir zu bleiben. So konnte ich niemals nachlassen.

Weiters unterstützten mich einigen Studienkollegen: Alexander Dusch, Martin Gomez, Thomas Hinterstoisser, Michael Jank, Edith Kemeny, Georg Neugschwandtner.

Uschi Göltl, die meine allererste wissenschaftliche Arbeit betreute und mich dadurch auch beeinflusste, ein technisches Studium zu wählen.

Beruflich sei vor allem Michael Simon genannt, da er mir den ersten „echten“ Job in meinem Leben gab. Er und die Integro Family zeigten mir, dass Arbeit Spaß machen kann. Bei Markus Hadek lernte ich in zwei Monaten praxisorientiert mehr, als in meinem ganzen

Studium davor. Bei Alex Redlein und dem IUCCIM, konnte ich den universitären Instituts-Alltag miterleben.

Allen diesen Personen gebührt mein aufrichtiger Dank. Privat möchte ich noch Eva Wiedenhofer danken, vor allem da sie von Danksagungen gar nichts hält. Zu erwähnen wären noch ein paar meiner Freunde: Bozidar Nedic zeigt mir seit Jahren, wie man Firma, Studium und Privatleben unter einem Hut bringt; Milan Andjelkovic hat mich durch seine hervorragenden Leistungen auch zu besseren Erfolgen ermuntert; Stephan Nedwed demonstrierte mir, dass ein guter Student auch auf der Straße seine Interessen zu vertreten hat. CRS und die Leute die dahinter stehen haben sicherlich mein kreatives Potential erweitert und meinen Mut gefördert.

Als letztes möchte ich allen österreichischen Steuerzahlern dafür danken, dass sie durch ihre Abgaben mir eine gute und für mich ohne Gebühren verbundene Ausbildung gewährten. Ich werde in meinem späteren Leben durch meinen höheren Verdienst und die damit verbundenen höheren Abgaben ihnen diese Investition wieder zurückerstatten.

A. Bekannte Musiksammlung

Tabelle A.1 enthält die komplette Liste aller Lieder der bekannten Musiksammlung.

Karte	Titel	Version
adagio	Adagio aus Klarinettenkonzert	Mozart
adiemus	Adiemus	Adiemus
africa	Africa	Toto
aileinduinn	Ailein Duinn	Volkslied
aintnosunshine	Ain’t no sunshine	Lighthouse Family
air	Air aus Orchestersuite 3	Bach
alice	Living Next Door To Alice	Smokie
allegromolto	Allegro Molto	Brahms
allforlove	All For Love	Brain Adams, Rod Steward, Sting
alltoyou	All To You	The Rounder Girls
allymcbeal	Searching My Soul	Vonda Shepard
americanpie	American Pie	Don McLean
angels	Angels	Robbie Williams
anything	Anything goes	Tony Bennett
anywhereis	Anywhere Is	Enya
aroundtheworld	Around the world	ATC
aufderhoeh	Auf der höh	Volkslied
austria	I am from austria	Rainhard Fendrich
avemaria	Ave Maria	Schubert
babycomback	Baby come back	The Equals
backforgood	Back for good	Take That
badblood	Bad Blood	Ministry
badboy	Bad Boy	Gloria Estefan
bahnfrei	Bahn frei - Polka schnell	Strauß
beautyandbeast	Die Schöne und das Biest	Jana Werner, Peter Hofmann
bebopalula	Bebopalula	Gene Vincent
beethoven	5th Symphony 1st Movement	Beethoven
believe	Believe	Cher
bigworld	Big Big World	Emilia
blueberry	Blueberry Hill	Fats Domino
bongobong	Bongo Bong	Manu Chao
boogiewoogie	Boogie Woogie Bugle Boy	Bette Midler
branden	Brandenburgische Konzert 2 Andante	Bach
breakfree	I Want To Break Free	Queen
breathaway	Take My Breath Away	Berlin
breathless	Breathless	The Corrs
bundeshymne	Bundeshymne	Mozart
cabaret	Cabaret	Liza Minelli
californiadream	California Dreaming	Mamas and the Papas
cheektocheek	Cheek to cheek	Ella Fitzgerald
cocojambo	Coco Jambo	Mr President
Karte	**Titel**	**Version**

Karte	Titel	Version
conga	Conga	Gloria Estefan
crashboombang	Crash Boom Bang	Roxette
dancingqueen	Dancing Queen	ABBA
dayinparadise	Another Day In Paradise	Phil Collins
deepisyourlove	How Deep Is Your Love	Take That
diamonds	Diamonds are a girls best friend	Marilyn Monroe
dingdong	Ding Dong	EAV
distance	From A Distance	Bette Midler
donau	Donauwalzer	Strauß
donttalkanymore	We Don’t Talk Anymore	Cliff Richard
drive	Drive	The Cars
drummerboy	Little drummer boy	Bing Crosby, David Bowie
dschinghiskhan	Dschinghis Khan	Dschinghis Khan
duellingviolins	Duelling violins	Ronan Hardiman
duhast	Du Hast	Rammstein
elvira	Andante Klavirkonzert no.21 „Elvira Madigan“	Mozart
eternalflame	Eternal Flame	Bangles
everlastinglove	Everlasting Love	Gloria Estefan
everythingido	Everything I Do	Bryan Adams
fatherandson	Father And Son	Cat Stevens
feeling	You’ve lost that lovin’ feeling	Rightous Brothers
feellovetonight	Can You Feel The Love Tonight	Elton John
feliznavidad	Feliz Navidad	Jose Feliciano
firsttime	The First Time	Robin Beck
flute	Flötenkonzert in G minor - andante	Buffardin
forelle	Trout - Quintet - Themen und Variationen	Schubert
foreveryoung	Forever young	Rod Stewart
fortuna	O Fortuna Imperatrix Mundi	Carl Orff
friend	You’ve Got A Friend	Carole King
fromnewyorktola	From New York to L.A.	Stephanie McKay
frozen	Frozen	Madonna
fuerstenfeld	Fürstenfeld	STS
fuguedminor	Toccata und Fugue in D Minor	Bach
funeral	Begräbnismarsch (Piano Sonate 2)	Chopin
future	End Credits	Back To The Future II
gaelicreels	Gaelic Reels	Volkslied
geldumagstmi	Gel Du Magst Mi	Ludwig Hirsch
girls	Girls Girls Girls	Sailor
giubba	Vesti La Giubba	Domingo
goldeneye	GoldenEye	Tina Turner
goodbye	Time To Say Goodbye	Brightman, Bocelli
goodgolly	Good golly miss molly	Little Richard
goodmornblues	Good morning blues	Frank Muschalle
gowest	Go West	Pet Shop Boys
griechischwein	Griechischer Wein	Udo Jürgens
grossvater	Grossvater	STS
heavensdoor	Knockin On Heaven’s Door	Randy Crawford
help	Help!	Beatles
herzilein	Herzilein	Wildecker Herzbuam
holdon	Hold On	Wilson Phillips
icanfly	I Believe I Can Fly	R Kelly
icouldfly	Wish I Could Fly	Roxette
ifonly	If Only	Rod Stewart
ifyoubelieve	If You Believe	Sasha
ihavenothing	I Have Nothing	Whitney Houston
imf	Mission Impossible	Adam Clayton, Larry Mullen
indy	The Raiders March	John Williams
ironic	Ironic	Alanis Morissette
itsinhiskiss	It’s in his kiss	Vonda Shepard
jailhouserock	Jailhouse Rock	Elvis
Karte	**Titel**	**Version**

Karte	Titel	Version
joeschau	Jö schau	Georg Danzer
johnnyb	Johnny b. goode	Chuck Berry
jurassicpark	Jurassic Park	John Williams
kaktus	Mein kleiner gruener Kaktus	Comedian Harmonists
kidscene	Fremde Länder und Menschen	Schumann
kiss	Kiss	Prince
kissfromarose	Kiss From A Rose	Seal
lastchristmas	Last Christmas	Wham
lastdance	Save the last dance for me	The Drifters
latinolover	Latino Lover	Loona
laymedown	As I Lay Me Down	Sophie B Hawkins
leavingport	Leaving Port	James Horner
lemontree	Lemon Tree	Fools Garden
letsgetloud	Let’s get loud	Jennifer Lopez
life	Life	unknown duett
lorddance	Lord of the dance	Ronan Hardiman
lovedwoman	Have You Ever Really Loved A Woman	Bryan Adams
lovemetender	Love Me Tender	Elvis
lovesombodysomt	Everybody loves somebody sometimes	Dean Martin
lovsisintheair	Love is in the Air	John Paul Young
macarena	Macarena	Los Del Rio
madlydeeply	Truly Madly Deeply	Savage Garden
mambofive	Mambo No 5	Lou Bega
manicmonday	Manic Monday	Bangles
matilda	Matilda	Harry Belafonte
mazurka	Mazurka in a Moll	Frederic Chopin
mcgee.mps	Me and Bobby McGee	Kenny Rogers
memory	Memory	Barbara Streisand
merry	The Merry Peasant	Schumann
mindfiels	Mindfields	Prodigy
minuet	Minuet	Boccherini
missathing	I Don’t Want To Miss A Thing	Aerosmith
missingyou	I’ll Be Missing You	Puff Daddy
missyoucrazy	Miss You Like Crazy	Natalie Cole
mmmbop	Mmmbop	Hanson
mond	Mondscheinsonate	Beethoven
moonlightshadow	Moonlight Shadow	Mike Oldfield
morgen	Morgenstimmung	Edvard Grieg
morningbroken	Morning Has Broken	Cat Stevens
mountainking	In der Halle des Bergkönigs	Grieg
movingonup	Moving On Up	M-People
myheartwillgoon	My Heart Will Go On	Celine Dion
myloveyourlove	My Love is Your Love	Whitney Houston
myway	My Way	Frank Sinatra
nachtmusik	Eine kleine Nachtmusik	Mozart
nahnehnah	Nah Neh Nah	Vaya Con Dios
newyork	New York, New York	Frank Sinatra
nocturne	Nocturne norwegisch	Secret Garden
onedaymore	One Day More	Les Miserables
onlyyou	Only You (and you alone)	The Platters
pachelbl	Canon	Pachelbel
panoptikum	Unsquare Dance	Dave Brubeck
party	Let’s have a party	Wanda Jackson
peggysue	Peggy Sue	Buddy Holly
philadelphia	Streets Of Philadelphia	Bruce Springsteen
pinkpanther	Pink Panther Theme	Henry Mancini
piubellacosa	Piu Bella Cosa	Eros Ramazzotti
poweroflove	The Power of Love	Huey Lewis
radetzkymarsch	Radetzkymarsch	Johann Strauß Vater
radio	Radio	The Corrs
Karte	**Titel**	**Version**

Karte	Titel	Version
rainbow	Over the Rainbow	Judy Garland
readmymindlight	If You Could Read My Mind	Gordon Lightfoot
readmymindstars	If You Could Read My Mind	Stars on 54
requiem	Requiem	Mozart
revolution	Do you hear the people sing?	Les Miserables
risingsun	House of the Rising sun	Animals
riverdance	Riverdance	Bill Whelan
roadtohell	The Road To Hell	Chris Rea
rockdj	Rock DJ	Robbie Williams
rockisdead	Rock is Dead	Marilyn Manson
rocknrollkids	Rock'n Roll Kids	Paul Harrington, Charlie McGettigan
running	Keep on running	Spencer Davis Group
saints	Oh When The saints go marchin' in	Luis Armstrong
sanfrancisco	San Francisco	Scott McKenzie
saymyname	Say my name	Destiny's Child
schindler	Schindlers Liste	John Williams
schneib	schneibb scho obar ins Tal	Grenzlandchor Arnoldstein
schwan	Schwanensee (Scene)	Tschaikowski
seeyouwhen	See You When You Get There	Coolio
sexbomb	Sexbomb	Tom Jones
shakespeare	The Beginning of the Partnership	Shakespeare in Love
she	She	Elvis Costello
shoopshoopsong	Shoop Shoop Song	Cher
sing	Sing Sing Sing	Glenn Miller
singalongsong	Sing along song	Tim Tim
sport	Es lebe der Sport	Rainhard Fendrich
starwars	Star Wars	John Williams
stormsinafrica	Storms in Africa	Enya
submarine	Yellow Submarine	Beatles
summercity	Summer in the city	Lovin' Spoonful
summerdreaming	Summer Dreaming	Kate Yanai
sunshineoflife	You Are The Sunshine Of My Life	Stevie Wonder
supertrouper	Super Trouper	A-Teens
surfusa	Surfin' USA	Beach Boys
takefive	Take Five	Dave Brubeck
talkaboutlove	Let's Talk About Love	Celine Dion
tannenbaum	Oh, Tannenbaum	Volkslied
tell	William Tell Overture (conclusion)	Rossini
tellhim	Tell Him	Vonda Shepard
thecircleoflife	The Circle Of Life	Elton John
themangotree	Under The Mango Tree	Tim Tim
therose	The Rose	Bette Midler
threetimesalady	Three Times A Lady	Lionel Richie
tiffany	Breakfast At Tiffany's	Deep Blue Something
timeaftertime	Time After Time	Cyndi Lauper
timewarp	Time Warp	Rocky Horror Picture Show
togetheragain	Together Again	Janet Jackson
torn	Torn	Natalie Imbruglia
tritschtratsch	Tritsch Tratsch Polka	Strauß
unbreakmyheart	Un-Break My Heart	Toni Braxton
veronika	Veronika, der Lenz ist da	Comedian Harmonists
vivaforever	Viva Forever	Spice Girls
Walkinincohn	Walking In Memphis	Marc Cohn
walkininmemphis	Walking In Memphis	Cher
walzer	Walzer op.39 No15	Brahms
whatsawoman	What's A Woman	Vaya Con Dios
whatsup	What's Up	4 Non Blondes
whenyousay	When you say nothing at all	Ronan Keating
whitechristmas	White Christmas	Bing Crosby
wildwildwest	Wild Wild West	Will Smith
Karte	**Titel**	**Version**

Karte	Titel	Version
wonderfulworld	What a Wonderful World	Louis Armstrong
wonderland	Wonderland	Passion Fruit
yesterday	Yesterday	unknown
yesterdbeatles	Yesterday	Beatles
yougotit	You Got It	unknown
youlearn	You Learn	Alanis Morissette
zapfenstreich	Großer Zapfenstreich	Hornsignale
zarathustra	Also sprach Zarathustra	Richard Strauss
zillertaler	Zillertaler Hochzeitsmarsch	Zillertaler Schürzenjäger

Tabelle A.1.: Musiksammlung

B. Freie Musiksammlung

Tabelle B.1 enthält die komplette Liste aller Lieder der freien Musiksammlung. Die Musik stammt zum überwiegenden Teil aus dem Internet [Aud00] von Audiogalaxy.com und Stefan Poiss, der für die Musik des Weltraumspiels PARSEC [Par00] verantwortlich ist und diese für die Experimente zur Verfügung stellte[1]. Die Spalte „Genre“ zeigt das auf Audiogalaxy.com angegebene Genre.

Karte	Titel	Interpret	Genre
abbey	Abby of Metal	Paralysing Prophecy	Metal
acidmachine	Acid Machine	Ritual	Metal
americantrio	American Trilogy	The Delgados	Rock
animaanimale	Anima Animale	Arto Lindsay	Jazz
aroundblock	Once around the block	Badly drawn boy	Rock
aroundyou	All around you	The Lonesome brothers	Country
backontrack	Back on Track	Babatunde Lea	Jazz
badblood	Bad Blood	Son Seals	Blues
balloonrace	The Last great balloon race	Mike Nicolai	Country
bd	Big D	Charybdis!	Punk
beatemall	America beat em all (to the moon)	Li'l Cap'n Travis	Country
beboxdual	BeBox	The 1-4-5s	Punk
between	Betwist or between	The rock a teens	Rock
bill	Bill the Pharmacizt	Hip Hop Congress	Hip-Hop
binoculars	Binoculars	Savath + savalas	Electronic
blancanina	Por Que Llorax la Blanca Nina	Yid Vicious	Folk
blazinggowns	Blazing Gowns	Sophistes	Metal
blindmics	Blind Mics	Khromozomes	Hip-Hop
blonde	Bottle Blonde	The Kiss-Offs	Punk
booze	Booze up and riot	Premium	Metal
borders	Borders	Granfaloon Bus	Country
bravenewworld	Brave new world	The gunga din	Rock
breeds	It Breeds	TOW	Metal
Bullfrogtheme	Bullfrog Theme	Bullfrog	Hip-Hop
burningofangels	The devine burning of angels	The project hate	Metal
calicogirl	Calico Girl	Guy Forsyth	Blues
callmecrazy	Don't call me crazy	Libbi Bosworth	Country
camdentownrain	Camden Town Rain	Mary Lou Lord	Folk
cantopraexu	Canto Pra Exu	Virginia Rodrigues	Folk
christopher	Christopher Says	Brenda Kahn	Folk
cleopatra	Rocknroll Cleopatra	The Wontons	Punk
cosmotron	Cosmotron	Stefan Poiss	Electronic
countrygirl	Country Girl	Analog Brothers	Hip-Hop
crazytrain	Crazy Train	The Flatirons	Country
Karte	**Titel**	**Interpret**	**Genre**

[1] PARSEC wird von Studenten der TU-Wien entwickelt

Karte	Titel	Interpret	Genre
critic	New Rock Critic	Skull Kontrol	Punk
daddydizzy	Don’t make your Daddy Dizzy	Rod Piazza and the Mighty Flyers	Blues
darkness	In the darkness	Orchards and Vines	Rock
dedicated	Dedicated	Dynamic Syncopation	Hip-Hop
derailed	Derailed	Epicure	Punk
diane	Diane	The Hippos	Punk
discoguitar	Disco Guitar Remix	DJ Assault	Hip-Hop
doppler	KC Doppler	Slide Five	Electronic
dreamon	Dream On	TP Allstars	Hip-Hop
drfunk	Dr Funk	Carl Cox	Electronic
elcordobes	El Cordobes	Diaz	Hip-Hop
falling	Falling	Stefan Poiss	Electronic
fingernails	Dirty fingernails	Modest Mouse	Rock
galaxies	Galaxies	Stefan Poiss	Electronic
gongdance	Le Gong Dance	Pitamaha - Music from Bali	Folk
goodleader	He’s a mighty good leader	Beck	Folk
gorgeousfriend	Gorgeous Friend	Cafeteria	Country
handthatfeeds	The Hand that feeds	PITT	Metal
hardside	Tales from the hardside	Omar Santana	Electronic
hickorywind	Hickory Wind	The Gram Parsons Notebook	Country
hirake	Hirake chu-rip	Kks	Folk
hole	God-Shaped Hole	Hayseed	Country
holocaust	Holocaust	Deride	Metal
honolulunights	Honolulu Nights	Auldridge, Brozman, Grisman	Folk
hookitup	Hook It Up	The Donnas	Punk
hurtsobad	It hurts so bad	Susan Tedeschi	Blues
Leavemealone	Blues leave me alone	Kim Wilson	Blues
lockyouup	Lock you up	The Love Dogs	Blues
lonely	Mlonely while you’re gone	Bobby Wynne	Country
losbilbilicos	Los Bilbilicos	Judy Frankel	Folk
mambo	Mambo	Dominic Halpin and the suspects	Jazz
marylee	Hello mary lee	Lazy lester	Blues
meistamoneski	Meista Moneski	Pijall	Hip-Hop
milwaukee	Milwaukee Blues	The Mad Cat Trio	Country
mindkiller	Mindkiller	Stefan Poiss	Electronic
montysgroove	Monty’s Groove	Monty Alexander	Jazz
moyo	Moyo	Stefan Poiss	Electronic
newone	New One	Sheraff	Jazz
nipples	Nipples	Peter Brotzmann	Jazz
nufonia	A Night at the nufonia	Kid Koala	Hip-Hop
nuttmeg	Nuttmeg World	Dirty Walt and the Columbus Sanatation	Blues
october	October	National Skyline	Rock
oracle	The oracle	Bugs	Electronic
orient	Fly the Orient	Tricky Woo	Punk
outofmyway	Get out of my way	Roach Powder	Metal
outtaspace	Dis Go’s N outta space	Michaelangelo	Electronic
owlservice	The Owl Service	Pram	Electronic
parsec	Parsec Main Theme	Stefan Poiss	Electronic
pounds	1000 Pounds	Superchunk	Rock
pushthatthing	Push that thing	Dave Aude	Electronic
quienerestu	Quien Eres Tu	Francisco Aguabella	Jazz
rabenwald	Rabenwald	Mika	Electronic
rad	Rad 180	Godheadsilo	Metal
reagansdead	Reagan’s Dead	The Prima Donnas	Punk
rebel	Im a loner dottie a rebel	The get up kids	Punk
righe	Tra le righe	I Guerrieri	Punk
roots	Roots	Marques Wyatt	Electronic
roulette	Roulette	Vandermark 5	Jazz
samba	Samba	King sunny ade	Folk
sandyloam	Sandy loam	Walt Wilkins	Blues
Karte	**Titel**	**Interpret**	**Genre**

Karte	Titel	Interpret	Genre
saramala	Saramala	Stephan Rigert's Talking Drums	Jazz
shelter	Shelter	AV Shelter	Electronic
showdown	The twilight showdown	Starlight mints	Rock
singingtheblues	Still singing the blues	Randy Garibay	Blues
situation	It's a cheatin' situation	The Wandering Eyes	Country
smile	The shadow of your smile	Sonny Stitt	Jazz
snakes	Snakes	Susan mckeown	Folk
somethingvag	Something vague	Bright eyes	Rock
somethingwrong	Something Wrong	The Hissyfits	Punk
soupdujour	Soup Du Jour	Cookin' with Kurt	Jazz
steelcitystrut	Steel City Strut	Turtle Island String Quartet	Jazz
strawberry	Anjali	Stawberry Mousse	Electronic
sunnypasture	Sunny Pasture	Vic Chesnutt and Mr and Mrs Keneipp	Folk
swinglow	Swing Low swing chariot	Sackcloth and Ashes	Blues
tenderlu	Tenderly	Soulstice	Electronic
thecat	The Cat	People under the stairs	Hip-Hop
thegrain	Against the grain	Lunacy	Metal
thesedays	Some of these days	Paul Burch and the WPA Ballclub	Country
thesword	The way of the sword	Power Symphony	Metal
thingslook	Things are looking up	The Lucky Strikes	Jazz
timeoflife	You're having the time of my life	Jets to brazil	Rock
tiny	Tiny	Tommy Guerrero	Electronic
torment	The Warmth in her torment	Eternal Suffering	Metal
totheshores	To the shores	Andrey Vinogradov	Folk
tripleoptics	Triple Optics	The Funky Precedent	Hip-Hop
tuskegee	Tuskegee Experiment	Soul Prophetic Sound	Hip-Hop
unsound	Unsound	Bettie Serveert	Rock
vsledzadja	VsledZaDja	Olga Arefieva and Kovcheg Band	Folk
waterfront	Waterfront	The black heart Procession	Rock
wildcatboogie	Wildcat Boogie	Wildcat O'Halloran Band	Blues
wishfuldrink	Wishful Drinking	Roger Wallace	Country
woman	Get out my life woman	Bobby Nathan	Blues
wootieboogie	Wootie Boogie	Lavelle White	Blues

Tabelle B.1.: freie Musiksammlung

C. Umrechentafel

Tabelle C.1 stellt eine einfach Umrechenmöglichkeit von Indizes auf der Segmentkarte zur respektiven Stelle im Lied in Sekunden dar. Auf der Segmentkarte folgt auf die Liedbezeichnung eine Nummer getrennt durch einen Punkt (Index Spalte). Diese Nummer stellt einen Zeitindex im Lied dar. Ein Beispiel: *ironic.15* symbolisiert im Lied *Ironic (Alanis Morrissette)* den Ausschnitt von 1:15 bis 1:20 Minuten. Um nicht immer eigenhändig umrechnen zu müssen ist die folgende Tabelle abgedruckt.

Index	m	s
1	0	5
3	0	15
5	0	25
7	0	35
9	0	45
11	0	55
13	1	5
15	1	15
17	1	25
19	1	35
21	1	45
23	1	55
25	2	5
27	2	15
29	2	25
31	2	35
33	2	45
35	2	55
37	3	5
39	3	15
41	3	25
43	3	35
45	3	45
47	3	55
49	4	5
51	4	15
53	4	25
55	4	35
57	4	45
59	4	55

Tabelle C.1.: Umrechentabelle

D. Abkürzungen

A/D-Wandler Analog/Digital-Wandler

ASR Automatic Speech Recognition

FFT Fast Fouriertransformation

FR Fourierreihe

GB Gigabyte

GG Growing Grid

GHSOM Growing Hierarchical Self-Organizing Map

HFM Hierarchical Feature Map

LVQ Learning Vektor Quantisation

MB Megabyte

MIDI Musical Instruments Digital Interface

ms Millisekunden

PCM Pulse Code Modulation

QF Quantisierungsfehler

SOM Self-Organizing Map

SR Sampling Rate

XMMS X-Multimedia System

Literaturverzeichnis

[Ghi95] David Chamberlin, Asif Ghias, Jonathan Logan and Brian C. Smith. Query by Humming: Musical Information Retrieval in an Audio Database. In *Proceedings of the third ACM international conference on Multimedia*, pages 231 – 236, 1995.

[Aud00] AudioGalaxy.com. *AudioGalaxy.com* Free Music. www.AudioGalaxy.com. online, November 2000. Fetched: 16.11.2000.

[Bar94] Hans-Jochen Bartsch. *Taschenbuch mathematischer Formeln: Nachschlagewerk zur höheren Mathematik*. Fachbuchverlag Leipzig-Köln, 1994.

[Bay95] Harald F. Bayer. Über die Anwendung selbstorganisierender Karten, 10 1995.

[Bri95] E. Oran Bringham. *FFT Schnelle Fourier-Transformation*. R. Oldenbourg Verlag, München Wien, 6 ed., 1995.

[Bro78] Riemann Brockhaus. *Musiklexikon*. Schott's Söhne, Mainz, 1978.

[BW99] G.J. Brown and DeLiang Wang. Separation of Speech from Interfering Sounds based on Oscillatory Correlation. *IEEE Transactions on Neural Networks*, 10(3):684 – 697, 1999.

[But98] Tilman Butz. *Fouriertransformation für Fußgänger*. B. G. Teubner Stuttgart, Leipzig, 1998.

[Enc97] Microsoft Corporation. *Microsoft Encarta 97 Enzyklopädie*. CD-ROM, 1997. Lexikon.

[McN99] I.H. Witten, D. Bainbridge, C.G. Nevill-Manning, L.A. Smith, and R. McNab. Towards a Digital Library of Popular Music. In *Proceedings of Fourth ACM Conference on Digital Libraries*, Berkeley, California, USA, 1999.

[VE92] B. L. Vercoe and D. PW Ellis. A Percetual Representation of Sound for Auditorial Signal Separation, May 1992. *presented at the 123rd Meeting of the Acoustical Society of America*, Salt Lake City.

[BDE99] Steven Blackburn, David De Roure, Samhaa El-Beltagy and Wendy Hall. A Multitangent System for content based Navigation of Music. In *Proceedings of seventh ACM international conference (part 2) on Multimedia*, pages 63 – 66, 1999.

[Dix00] S. E. Dixon. A Lightweight Multi-agent Musical Beat Tracking System. In *Proceedings of the Pacific Rim International Conference on Artificial Intelligence*, pages 778 – 788, Melbourne, Australia, 2000.

[Dör00] Michael Dörfler. Interview. spoken, 5. Oktober 2000. www.tripbox.com . Founder of Tripbox.

[Dow99] J. Stephen Downie. Music Retrieval as Text Retrieval (poster abstract): simple yet effective. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 297 – 298, 1999.

[Wol96] D. Keislar, E. Wold, T. Blum and J. Wheaton. Content-based Classification, Search, and Retrieval of Audio. In *IEEE MultiMedia*, volume 3, pages 27 – 36. IEEE MultiMedia, 1996.

[ER95] D. Ellis and D. Rosenthal. Mid-level Representations for Computational Auditory Scene Analysis. In *International Conference on Artificial Intelligence*, Montreal, Quebec, August 1995. Workshop on Computational Auditory Scene Analysis.

[FG94] B. Feiten and S. Günzel. Automatic Indexing of a Sound Database using Self-organizing Neural Nets. *Computer Music Journal*, 18(3):53 – 65, Fall 1994.

[Fo97a] J.T. Foote. An Overview of Audio Information Retrieval. In *ACM Multimedia*, 12 1997.

[Fo97b] J.T. Foote. Content-based Retrieval of Music and Audio. In *Multimedia Storage and Archiving Systems II, Proceedings of SPIE*, volume 3229, pages 138 – 147, 1997.

[Ram96] G. Ramharter *Mathematik für Bauingenieure*. Entwurf zum Skriptum, LVA 102.789, 1996.

[Gje89] R. O. Gjerdingen. Using Connectionist Models to Explore Complex Musical Patterns. *Computer Music Journal*, 13(3):67 – 75, Fall 1989.

[BM93] J. Blackmore and R. Miikkulainen. Incremental Grid Growing - Encoding High-dimensional Structure into a Two-dimensional Feature Map. In *Proc of the IEEE Int'l Conf on Neuronal Networks*, San Francisco, CA, 1993.

[Jam95] J.F. James. *A students's guide to Fourier transforms*. Cambridge University Press, 1995.

[CHL98] Arbee L. P. Chen, Jia-Lien Hsu and C.-C. Liu. Efficient Repeating Pattern Finding in Music Databases. In *Proceedings of ACM 7th international conference on Information and knowledge management*, pages 281 – 288, 1998.

[Kas00] Samuel Kaski. Self-organizing Maps for Data Mining. Lecture, June 2000.

[Koh97] T. Kohonen. *Self-Organizing Maps*. Springer-Verlag Berlin Heidelberg, 2 edition, 1997.

[Koh81] Teuvo Kohonen. Automatic Formation of Topological Maps of Patterns in a Self-organizing System. In *Proceedings of the Second Scandinavian Conf on Image Analysis*, pages 214 – 220, 1981.

[MDR00] D. Merkl, M. Dittenbach and A. Rauber. The Growing Hierarchical Self-organizing Map. In *Proceedings of the International Joint Conference on Neural Networks 2000 (IJCNN'2000)*, Como, Italy, 24. - 27. July 2000.

[MG97] Y. Muraoka and M. Goto. Real-time Rhythm Tracking for Drumless Audio Signals – Chord Change Detection for Musical Decisions. In *IJCAI97 Workshop on Computational Auditory Scene Analysis*, 1997.

[Par00] Clemens Beer, Markus Hadwiger, Andreas Varga, Michael Wögerbauer, and Stefan Poiss et al. *parsec* – there is no safe distance. www.parsec.org . online, 2000.

[Mer95] D. Merkl. Content-based Software Classification by Self-organization. In *Proceedings of the IEEE Int'l Conference on Neural Networks (ICNN'95)*, Perth. Australia, Nov 27 - Dec 1995.

[MR00] D. Merkl and A. Rauber. *Document Classification with Unsupervised Neural Networks*, pages 102 – 121. Physica Verlag, Germany, 2000. ISBN: 3790812994.

[MID96] MIDI Manufacturers Association (MMA). *MIDI 1.0 Specification*. www.midi.org . online, 1996.

[Mus01] musclefish.com. *Muscle Fish*. www.musclefish.com . online, 2 2001.

[PCL94] G. De Poli, P. Cosi and G. Lauzzana. Auditory Modeling and Self-organizing Neural Networks for Timbre Classification. *Journal of New Music Research*, 23:71 – 98, 1994.

[XMM00] Olle Hallnas, Peter Alm, Mikael Alm, Thomas Nilsson, and 4Front Technologies. *XMMS* – Cross-platform Multimedia Player. www.xmms.org . Sourcecode, 2000.

[McN96] Ian H. Witten, Roger J. McNab, Lloyd A. Smith, Clare Henderson, and Sally Jo Cunningham. Towards the Digital Music Library: Tune Retrieval from Acoustic Input. In *Proceedings of the ACM Digital Libraties Conference*, Bethseda, 1996.

[Maa01] Commonwealth Scientific and Industrial Research Organisation. Faq for *MPEG maaate* v 0.1.0. online, February 2001.

[Sei96] F. Seifert. *Elektrotechnische Grundlagen der Informatik*. Skriptum, January 1996.

[MoC98] Gerald Kühne, Silvia Pfeiffer, Rainer Lienhart and Wolfgang Effelsberg. The MoCA Project – Movie Content Analysis Research at the University of Mannheim. In R. Kruse J. Dassow, editor, *Informatik '98: Informatik zwischen Bild und Sprache*. Springer Verlag, Berlin-Heidelberg, 1998.

[EPF96] Wolfgang Effelsberg, Silvia Pfeiffer, Stephan Fischer. Automatic Audio Content Analysis. Technical report, Department for Mathematics and Computer Science, University of Mannheim, 1996.

[Spe96] H. Speckmann. *Dem Denken abgeschaut*. Verlag Vieweg, Braunschweig-Wiesbaden, 1996.

[YS98] A. Youssef and S.R. Subramanya. Wavelet-based Indexing of Audio Data in Audio/Multimedia Databases. In *Proceedings of the International Workshop on Multimedia Database Management Systems*, 1998.

[SW98] A. Sterian and G. H. Wakefield. A model-based Approach to Partial Tracking for Musical Transcription, 1998. Presented at SPIE, San Diego, CA, USA.

[Tse99] Y.H. Tseng. Content-based Retrieval for Music Collections. In *Proceedings of the ACM International Conference on Research and Development in Information Retrieval (SIGIR)*, Berkeley, CA, USA, August 1999.

[UZ98] Alexandra L. Uitdenbogerd and Justin Zobel. Manipulation of Music for Melody Matching. In *Proceedings of the ACM Multimedia 98*, pages 235 – 240, England, Sep. 11-15 1998.

[NR92] W. T. Vetterling, W. H. Press, S.A.Teukolsky and B. P. Flannery. *Numerical Recipes in C The Art of Scientific Computing*. Cambridge University Press, 2 edition, 1992.

[NR97] W. T. Vetterling, W. H. Press, S.A.Teukolsky and B. P. Flannery. *Numerical recipes the art of scientific computing*. CD-ROM, 1997.

[Wha00] whatis.com. *PCM*. online, August 2000. Fetched: 22.1.2001.

[ZK98] T. Zhang and C. Kuo. Content-based Classification and Retrieval of Audio. In *SPIE's 43rd Annual Meeting – Conference on Advanced Signal Processing Algorithms, Architectures, and Implementations VIII*, San Diego, USA, July 1998.

Index