Ausarbeitung zum Thema

Audio ID
Hintergrund und Anwendungen

von
Jan Hoffmann, Tobias Hoppe und Georg Ruß

Otto-von-Guericke-Universität Magdeburg
Multimedia & Security, Sommersemester 2004

Prolog

Wenn ein Computer zwei Audiostücke vergleicht, dann kann er sie im Normalfall nur als gleich identifizieren, wenn sie bitweise gleich sind. Ein Mensch jedoch erkennt ein Lied anhand der Inhalte (Inhaltsmerkmale) im Vergleich mit seiner Erinnerung. Ist es möglich, diesen Nachteil auszugleichen, indem man dem Computer beibringt, Audiostücke wie ein Mensch zu erkennen?

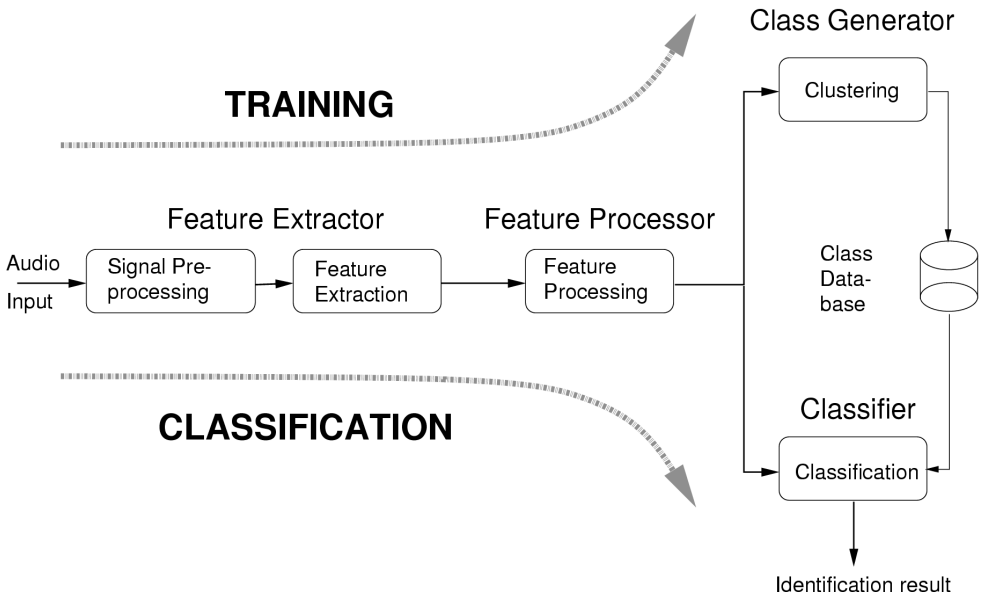
Audio-ID

Audio-ID ist ein solches System, das anhand von Audiodaten (CD-Kopien bis hin zu Mikrofonaufnahmen) Musikstücke identifizieren kann.

Es basiert auf zwei wesentlichen Szenarien (siehe Abbildung).

1.) Training:
Anhand einer Menge von Referenzsignalen wird eine Datenbank (vergleichbar mit dem menschlichen Gedächtnis) aufgebaut, auf die später bei der Klassifikation (s.u.) zurückgegriffen werden kann.

2.) Klassifikation
Ein Audiostück oder ein Ausschnitt desselbigen wird mit Hilfe der Datenbank (s.o.) identifiziert.



Im Folgenden wird ein kurzer Überblick über die einzelnen Schritte dieses Verarbeitungskette gegeben.

1. Signal-Vorverarbeitung

Da von vornherein keinerlei Annahmen über das Format der Audioproben gemacht werden, können diese in beliebigen Formaten vorliegen. Insbesondere spielt es keine Rolle, welche Sample-Frequenz oder Auflösung (Bitrate) das Signal hat, wieviele Kanäle es hat und in welchem Dateiformat es vorliegt.
Das Eingangssignal wird daher als erster Schritt über Resampling- und Downmixalgorithmen in ein Einheitsformat gewandelt. Als solches dient zur Zeit ein 44.100 Hz Monosignal, das sehr verbreitet und für die meisten Anwendungen ausreichend ist.
Am Ende der Vorverarbeitung steht ein wohldefiniertes Signal, das nun den weiteren Verarbeitungsschritten zugeführt werden kann.

2. Merkmalsextraktion

In diesem Schritt soll aus dem soeben in ein einheitliches Format gebrachten Eingangssignal eine Menge von Merkmalen extrahiert werden, die robust gegen verschiedene Veränderungs- bzw. Verzerrungsarten sind wie etwa:

- zeitliche Verschiebung
- Abschneiden
- Lautstärkeänderung
- wahrnehmungsbasiertes Audiokodieren (MP3)
- Veränderung von Frequenzbandlautstärken (Equalizer)
- Pass-Filterung (z.B. andere Samplingfrequenz)
- Kompression des Dynamikumfangs
- Hinzufügen von Rauschen
- Lautsprecher-Mikrofon-Übertragung

Dazu wird das Audiozeitsignal (mit Hilfe einer Ausschnitts-oder Fensterfunktion) segmentiert. Diese Ausschnittsdatensequenzen werden daraufhin per DFT in den Frequenzraum überführt.
Aus dem Spektrum der einzelnen Fenster wird nun jeweils ein Satz verschiedener psychoakustischer Merkmale extrahiert, die dann zusammen jeweils einen Merkmalsvektor bilden. Die Menge dieser Spektralmerkmale ist für diesen Zeitpunkt des Audiosignals charakteristisch und wird in den weiteren Schritten verwendet.

Für diesen Schritt geeignete Merkmale sind:

- Lautheit
  Die Lautheit bezeichnet die Intensität eines Audiostücks. Dabei wird nicht nur die Amplitude, sondern auch die durchschnittliche Lautstärke berücksichtigt. Das kann sowohl für das gesamte Signal oder auch für einzelne Frequenzbänder geschehen. Die Lautheit ist ein recht robustes Merkmal.
- Spektrales Flachheitsmaß
  Das spektrale Flachheitsmaß (SFM) unterscheidet zwischen eher tonähnlichen und eher rauschähnlichen Signalen, wobei das SFM bestimmt, wie tonal das Signal ist. Wie bei der Lautheit wird eine Multiband-Version dieses Maßes benutzt.
- Schärfe oder Spektrale Neigung
  Schärfe enspricht der wahrnehmbaren Brillianz eines Audiosignals, alternativ dazu bestimmt die spektrale Neigung die Steigung des Spektrums von den tiefen zu den hohen Frequenzen auf logarithmierter Skala. Beide Merkmale sind robust

Am Ende dieses Schrittes wurden ausgewählte Merkmale aus den Audiodaten extrahiert.

3. Merkmalsverarbeitung

In diesem Schritt werden die soeben erhaltenen Merkmale für die Erkennung optimiert.
Dazu wird zunächst sichergestellt, dass sich die einzelnen Elemente der Merkmalsvektoren in gleichen Größenordnungen bewegen, indem diese mit den jeweils zugehörigen Standardabweichungen normalisiert werden, nun also Einheitsvektoren sind.
Als nächstes werden Folgen von Werten der verschiedenen Merkmale gruppiert und bilden so einen neuen, höherdimensionalen Merkmalsvektor. Mit dessen Hilfe können nun weitere statistische Daten erhoben werden.

Im Folgenden wird durch Erhöhung der Dekorrelation der Merkmale die Orthogonalität der einzelnen Komponenten erhöht. Durch die damit verbundene Reduktion der Dimensionalität der Merkmalsvektoren können einige überflüssige Berechnungen eingespart werden. Außerdem hat sich gezeigt, dass die Erkennungsleistung dadurch sogar verbessert werden kann.

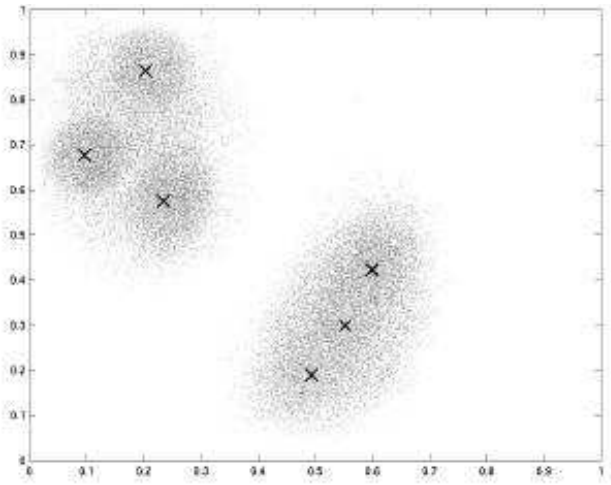
Nach diesem Schritt wurde ein Fingerabdruck des Eingabesignales in Form eines Vektors erzeugt, anhand dessen diese Probe nun identifiziert werden kann.

4. Clustern / Klassifikation

Dieser letzte Schritt wird prinzipiell sowohl für die *Klassifikation* eines Audiosignals als auch schon zuvor für das *Clustering*, den „Trainings-Modus" für das einmalige Aufbauen der Referenzdatenbank benötigt.

Als Eingabe dient der soeben erzeugte Fingerabdruck des aktuellen Audiosignals in Form einer großen Menge an Merkmalsvektoren.

In diesem Schritt wird diese Menge von Vektoren durch eine deutlich kleinere Menge repräsentativer Code-Vectoren approximiert, die dann ein charakteristisches Codebuch („Fingerabdruck") für die aktuelle Klasse (Audiostück) darstellen, wie die folgende Abbildung veranschaulicht. Dort werden sehr viele (hier 2-dimensionale) Merkmalsvektoren durch 6 Code-Vektoren approximiert.



Wie geschieht dies nun konkret?

Klassifikation
Für die Klassifikation eines Stückes geschieht dies mit Hilfe der vergleichsweise sehr performanten Nächster-Nachbar-Regel.

Clustern

Das Clustern zur Generierung der später der Klassifikation zugrunde liegenden Datenbank erfolgt jedoch sorgfältiger, für dieses Training der Referenzdaten kommt Vektorquantisierung zum Einsatz, die auf dem Prinzip des kleinsten quadratischen Fehlers (*RMSE/ „Root Mean Square Error")* basiert. Mit einem k-means-Clustering-Algorithmus werden die Vektoren nach und nach aufgeteilt, währenddessen der RMSE (Root Mean Square Error) so lange minimiert wird, bis er eine gewisse Schwelle unterschreitet oder eine verlangte Anzahl von fertigen Codevektoren erreicht wurde.
Das *Ergebnis des Clustern* ist letztendlich eine Menge von Codebüchern (je eins pro Audio-Stück)

Ergebnis der Klassifikation

Wie können die im Zuge einer Klassifikation erhaltenen Code-Vektoren nun genutzt werden, um das Audiosignal zu identifizieren?
Dazu wird im Abgleich mit der Datenbank für jede dort abgelegte Klasse der Näherungsfehler im direkten Vergleich der Codevektoren ermittelt. Diejenige Klasse, die den kleinsten akkumulierten Näherungsfehler erzeugt, wird als Ergebnis zurückgegeben. Alternativ kann als Ergebnis der Klassifikation auch eine Liste mit mehreren Klassen zurückgeliefert werden, die am ehesten zu dem verarbeiteten Audiosignal passen

Anwendungen

Musikidentifizierung mit Verknüpfung mit Metadaten

Ein Musikstück kann anhand einer Aufnahme (auch bloß eines kurzen Ausschnitts) identifiziert bzw. sein Fingerabdruck berechnet werden.
Daraufhin können aus einer Internetdatenbank Metadaten wie Titel, Interpret, Komponist oder Text abgefragt werden. Dieser Mechanismus kann und wird voraussichtlich zukünftig auch ID3-Tags oder CDDB-Anfragen ersetzen.

Identifizierung von spezifischen Audioinhalten

In großen Datenbeständen (z.B. P2P-Netzen) kann automatisch nach illegalen Inhalten gesucht werden. Wichtigster Einsatzzweck: Stärkung von intellektuellen Besitzrechten (bspw. Copyright)

Schutz von Inhalten

Diese Technologie stellt einen sehr robusten Weg dar, Audioinhalte zum Schutz ihrer Inhalte zu identifizieren, beispielsweise als Stärkung des Kopierschutzes. Sie ist fälschungssicherer als digitale Wasserzeichen, da keine zusätzlichen Informationen in die Audiodaten eingebettet werden und kommt daher auch ohne Qualitätsverlust aus.

Musikhandel

Auch tragbare Elektronik-Kleingeräte (Handys, PDAs, etc.) können die automatische Signaturerstellung ermöglichen und so beispielsweise den Kauf von Musik vereinfachen.

Sendeüberwachung

Die performante Identifikation von Audiosignalen ermöglicht automatische Überwachung von gespielten Stücken in Rundfunk und Fernsehen. Dies ermöglicht es Rechteverwertungseinrichtungen wie der GEMA, automatisch Gebühren einzufordern. Außerdem können über diese Medien werbende Firmen so leicht automatisch selbst kontrollieren, ob gekaufte Werbeeinspielungen ausgestrahlt werden. Ebenfalls sind auch allgemeine, statistische Analysen des Programms (z.B. Charts) möglich.

Anfragen durch Summen („Query by Humming")

Es genügt einem Nutzer, die Melodie eines gewünschten Stückes in ein Mikrofon zu summen. Nach Verarbeitung dieser Audiodaten könnten dann etwa die zehn bestpassendsten Audiostücke zurückgegeben werden. Diese Variante wäre ebenfalls für den Musikhandel (s.o.) interessant.

Testergebnisse

Die folgenden Tabellen zeigen die Ergebnisse zweier Testläufe.

Erster Testlauf:

Der erste Test wurde anhand eines 1000 Musikstücke aus dem Bereich Rock/Pop umfassenden Referenzdatensatzes durchgeführt. Das Training geschah anhand von 30-Sekunden-Ausschnitten dieser Titel, während die Klassifikation mit 20-Sekunden-Fragmenten durchgeführt wurde.

Die Tabelle zeigt die Ergebnisse für die zwei untersuchten Merkmale Lautheit und spektrales Flachheitsmaß (SFM) unter verschiedenen Bedingungen. Der linke der beiden Werte beziffert jeweils den Anteil der Titel, die korrekt erkannt wurden, während der rechte aussagt, wie oft der korrekte Titel zumindest unter den Top10 der Treffer zu finden war.

Merkmal:	Lautheit	SFM
Keine Verzerrung	100.0% / 100.0%	100.0% / 100.0%
Ausschnitt (15s)	51.1% / 75.5%	92.3% / 99.6%
Equalisiert	99.6% / 100.0%	14.1% / 29.8%
Dynamikkompression	89.5% / 94.9%	99.0% / 99.3%
MPEG-1/2 Layer 3 @ 96 kbit/s	19.0% / 33.3%	90.0% / 98.6%
Loudsprecher-Mikrofon-Kette	38.3% / 61.7%	27.2% / 59.7%

Zweiter Testlauf:

Der zweite Test wurde unter ähnlichen Bedingungen durchgeführt, hier lag die Anzahl der trainierten Referenzlieder jedoch bei 10.000. Als Merkmal wird in der Tabelle lediglich das spektrale Flachheitsmaß aufgeführt.

Feature:	SFM
Keine Verzerrung	100.0% / 100.0%
Ausschnitt	96.5% / 99.7%
MPEG-1/2 Layer 3 @ 128 kbit/s	97.3% / 98.9%
MPEG-1/2 Layer 3 @ 128 kbit/s & Ausschnitt	87.6% / 96.8%

Referenzen

[1] Eric Allamanche, Jürgen Herre, Oliver Hellmuth, Bernhard Fröba, Markus Cremer.
AudioID: Towards Content-Based Identification of Audio Material. In: 110th AES Convention, Amsterdam, 2001.

[2] Oliver Hellmuth, Eric Allamanche, Jürgen Herre, Thorsten Kastner, Markus Cremer und Wolfgang Hirsch.
Advanced Audio Identification Using MPEG-7 Content Description. In: 111th AES Convention, New York, 2001.

[3] Thorsten Kastner, Eric Allamanche, Jürgen Herre, Oliver Hellmuth, Markus Cremer und Holger Grossmann.
MPEG-7 Scalable Robust Audio Fingerprinting. In: 112th AES Convention, München, 2002.

[4] Oliver Hellmuth, Eric Allamanche, Markus Cremer, Holger Grossmann, Jürgen Herre und Thorsten Kastner.
Using MPEG-7 Audio Fingerprinting in Real-World Applications. In: 115th AES Convention, New York, 2003.

[5] J.A.Haitsma.
Audio Fingerprinting, „A New Technology to Identify Music“ Nat.Lab. Unclassified Report 2002/824, 2002, Koninklijke Philips Electronics N.V. 2002.

[6] Max Rauner.
Der Sinn der Musik. In: Technology Review, 3/2004, S. 37-41.
oder http://www.heise.de/tr/artikel/44683