Computer Vision for Music Identification

Philip Kurmann, philip@kman.ch

10 Juni 2017

Abstract

Das Paper "Computer Vision for Music Identification" von Yan Ke, Derek Hoiem und Rahul Sakthankar beschreibt einen Algorithmus zur Identifikation von Musik Titel anhand einer Audioaufnahme. Dafür werden Methoden aus der Bilderkennung eingesetzt, um mittels Filter im Spektrogramm einer Audioaufnahme Ankerpunkte zu finden.

Beim Einsatz in der Praxis besteht die Schwierigkeit, dass es sich bei der Suchanfrage um eine qualitativ schlechte oder verrauschte Tonaufnahme handeln kann. Der vorgestellte Algorithmus kann mit diesem Problem umgehen und zeichnet sich weiter durch eine hohe Erkennungsrate, eine hohe Präzision, den zuverlässigen Umgang mit kurzen Suchanfragen sowie mit einer schnellen Antwortzeit aus.

Das System ist sehr effizient beim Indexieren und skaliert daher gut. Dadurch eignet es sich, auch grosse Musikbibliotheken zu durchsuchen.

Computer Vision for Music Identification

Um Musik mittels Bilderkennungsverfahren zu erkennen, müssen die Schallwellen erst in Bilder umgewandelt werden. Hierzu werden die Songs in kurze, überlappende Stücke geteilt. Mittels der "Short-Term Fourier Transformation" (STFT) werden die Teilstücke anschliessend in Spektrogramme umgewandelt, welche 33 logarithmische Frequenzbänder zwischen 300 und 2000 Hz repräsentieren.

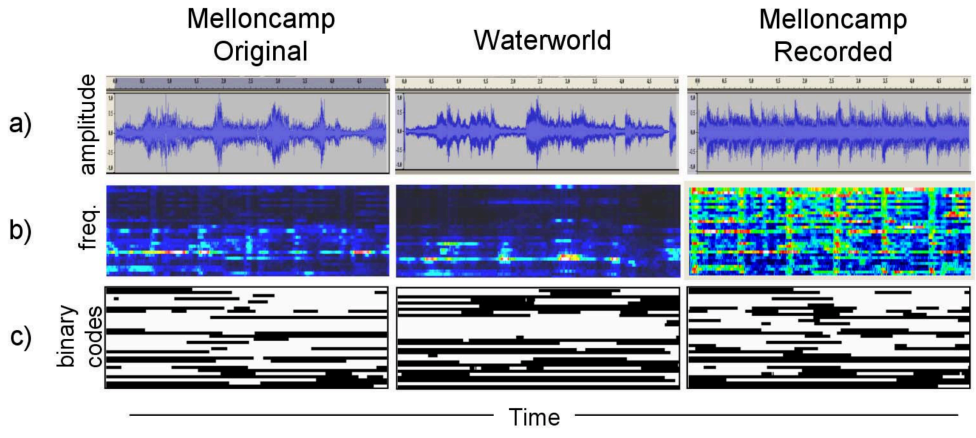


Figure 1: Darstellungen von Audiosignalen (Ke, Hoiem und Sukthankar 2005, p. 2)

Frühere Musikidentifkations- Methoden betrachteten 1-D Signale. Durch den beschriebenen Ansatz werden diese so als 2-D Bilder betrachtet. Bereits ohne weitere Filterung kann in diesen Spektrogrammen eine gewisse Ähnlichkeit von verrauschten bzw. verzerrten Aufnahmen mit den Originalen erkannt werden (siehe

Abbildung 1b). Einfaches Vergleichen von Spektrogrammen wäre jedoch sehr zeitaufwändig und ist daher nicht geeignet, um eine grosse Kollektion zu durchsuchen. Darüber hinaus ist es schwierig, aufgrund der hohen Informationsdichte Korrelationen zu erkennen. Daher wird eine kleine Anzahl Filter auf das Bild des Spektrogramms angewendet. Die Filter werden mittels dem Pairwise Boosting Algorithmus selektioniert. Das Ziel ist, durch die Filter Störgeräusche möglichst effizient zu eliminieren, dabei jedoch gleichzeitig die relevanten Informationen zu erhalten. Bei richtiger Selektion der Filter entstehen so zuverlässige Fingerprints, bei gleichzeitig starker Reduktion der Datenmenge.

Filter

Das Anwenden der Filter auf das Spektrogramm bewirkt, dass der Algorithmus weniger anfällig auf Störungen reagiert. Die Filter des beschriebenen Systems weisen folgende Charakteristiken auf: Basis Frequenz von 1-33, Bandweite von 1-33 und Zeit von 1 Frame (11.6ms) bis 82 Frames (951ms). Anhand dieser Parameter erhält man ca. 25'000 unterschiedliche Filter, wovon mittels Pairwise Boosting 32 Filter ausgewählt werden. Mögliche Filter sind in Abbildung 2 dargestellt.

Jeder Filter besitzt einen eigenen Treshold. Übersteigt der Wert des Filters an einer bestimmten Stelle im Spektrogramm den Treshold, wird eine 1 zurückgegeben. Andernfalls eine -1. Durch Anwenden der Filter auf ein Fenster von 2048 Samples, erhält man einen Deskriptor, ein ein M-Bit Vektor, welcher robust gegenüber Störgeräuschen und Verzerrungen ist. Durch wiederholtes und überlappendes Anwenden der Filter erhält man die Deskriptoren, M-Bit Vektoren, welche robust gegenüber Störgeräuschen und Verzerrungen sind (siehe Abbildung 1c). Um den Song zu erkennen werden mehrere sequenzielle Deskriptoren benötigt.

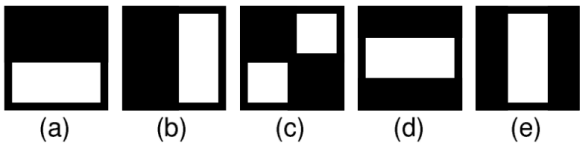


Figure 2: Mögliche Filter (Ke, Hoiem und Sukthankar 2005, p. 2)

Pairwise Boosting

Die Filter müssen so gewählt werden, dass das original Signal und seine verzerrte Version sehr ähnliche Deskriptoren liefern. Hingegen sollen Signale von unterschiedlichen Songs stark unterschiedliche Deskriptoren ergeben. Um den Song zu finden, wird anschliessend die Wahrscheinlichkeit ausgerechnet, ob ein Deskriptor der Anfrage und ein Deskriptor aus der Bibliothek vom selben Song stammen. Die Auswahl der Filter wird mittels Pairwise Bossting durchgeführt, eine Weiterentwicklung von Adaboost.

Das Ziel ist es, einen Klassifikator $H(x_1, x_2) \rightarrow y = \{-1, 1\}$ zu lernen, wobei x_1 und x_2 zwei Histogramme sind und y beschreibt, ob die Bilder der Audio-Teilstücke von der selben Quelle stammen. Der Klassifikator ist ein Ensemble von M schwachen Klassifikatoren, $h_m(x_1, x_2)$, welcher jeder eine Konfidenz, c_m, besitzt. Die schwachen Klassifikatoren sind aus einem Filter f_m und einem Threshold t_m zusammengesetzt. Die Formel lautet:

$$h_m(x_1, x_2) = sgn[(f_m(x_1)t_m)(f_m(x_2)t_m)]$$

Einfach ausgedrückt, werden die Audio-Teilstücke durch den schwachen Klassifikator als identisch eingestuft, wenn sie sich auf der selben Seite des Tresholds befinden.

Die besten Filter werden mittels Pairwise Boosting selektioniert. Hierbei handelt es sich um einen iterativen Machine Learning Algorithmus, bei dem nur für die passenden Teilstücke eine Anpassung der Gewichtung durchgeführt wird. In Pseudocode wird der Algorithmus folgendermassen beschrieben:

Pairwise Boosting Algorithmus

input: Sequenz von n Beispielen $\langle(x_{11}, x_{21})\rangle..\langle(x_{1n}, x_{2n})\rangle$, jedes mit label $y_i \in \{-1, 1\}$

Initialisierung: $w_i = \frac{1}{n}, i = 1..n$

for m=1..M

1. Finde die Hypothese $h_m(x_1, x_2)$, welche den gewichteten Fehler über die Verteilung w minimiert, wobei $h_m(x_1, x_2) = sgn[(f_m(x_1) - t_m)(f_m(x_2) - t_m)]$ für den Filter f_m und den Treshold t_m
2. Berechne den gewichteten Fehler: $err_m = \sum_{i=1}^{n} w_i \cdot \delta(h_m(x_{1i}, x_{2i}) \neq y_i)$
3. Konfidenzwert berechnen: $c_m = log(\frac{1-err_m}{err_m})$
4. Gewichte der passenden Paare anpassen: Wenn $_i = 1$ und $h_m(x_1, x_2) \neq y_i$, dann $w_i \leftarrow w_i \cdot exp[c_m]$
5. Gewichte normalisieren, so dass $\sum_{i:y_i=-1}^{n} w_i = \sum_{i:y_i=1}^{n} w_i = \frac{1}{2}$

Schlusshypothese:

$H(x_1, x_2) = sgn(\sum_{m=1}^{M} c_m h_m(x_1, x_2))$

Abfrage

Um grosse Musiksammlungen zu indizieren, ist es wichtig, dass das System skaliert. Ursprünglich wollte man den Retrieval-Prozess mittels Suche in einem Multidimensionalen Raum lösen und dafür Locality-Sensitive Hashing (LSH) einsetzen. Es hat sich jedoch gezeigt, dass es einen sehr einfachen Ansatz gibt, welcher ähnlich gute Resultate hervorbringt, jedoch die Laufzeit drastisch verbessert. Hierzu werden die Signaturen der Audio Teilstücke indexiert. Um bei der Abfrage alle Resultate innerhalb der Hamming Distanz 2 zu finden, geschieht diese in 3 Schritten: Im ersten Schritt wird die Hashtabelle mit dem Descriptor der Aufnahme durchsucht. Dies ergibt Resultate mit der Hamming Distanz 0. Danach wird die Abfrage M-mal wiederholt, jeweils mit einem anderen BIT, das gedreht wurde. Dies ergibt die Resultate mit Hamming Distanz 1. Danach werden alle Kombinationen mit 2 gedrehten Bits abgefragt. Dies ergibt die Resultate mit Hamming Distanz 2.

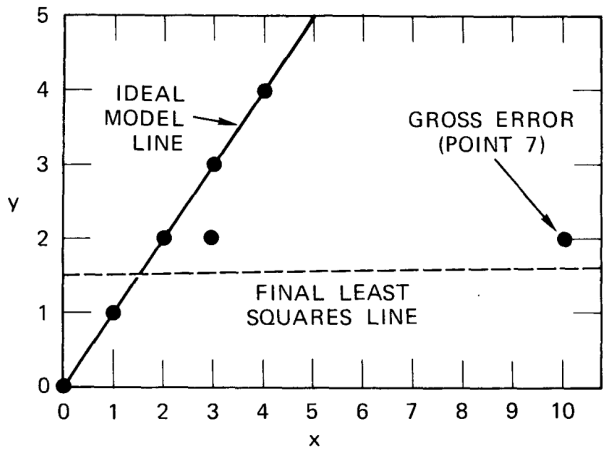
Nachdem alle Near-Neighbor Teilstücke innerhalb der Hamming Distanz 2 gefunden wurden, muss man den Song identifizieren. Dies wird anhand einer geometrischen Verifikation durchgeführt, um die zeitliche Ausrichtung der gefundenen Deskriptoren in den Vergleich miteinzubeziehen. Der verwendete Algorithmus, für diese geometrische Verifikation, heisst RANSAC. Er iteriert über die Zeitausrichtung der ermittelten Kandidaten und liefert als Resultat die Fehleranzahl (EM-Score), welche als Distanzmetrik dient, um den korrekten Song zu bestimmen. Dank RANSAC wird die Abfrage stabil gegenüber Deskriptoren, welche vorwiegend aus Störgeräuschen bestehen.

Random Sample Consensus (RANSAC)

RANSAC ist ein iterativer Algorithmus zur Schätzung eines Modells innerhalb einer Reihe von fehlerbehafteten Messwerten und wurde 1985 von Martin Fischler und Robert Bolles entwickelt. Er weisst eine hohe Robustheit gegenüber Ausreissern auf und findet deshalb Verwendung in diversen Machine Learning Verfahren, bei denen der Least Squares Ansatz schlecht funktioniert.

Voraussetzung für RANSAC ist, dass mehr Datenpunkte vorliegen, als für die Bestimmung des Modells benötigt werden. Die Vorgehensweise zur Bestimmung eines guten Modells mittels RANSAC kann folgendermassen zusammengefasst werden:

1. Zufällige Auswahl von Datenpunkte, die zur Bestimmung des Modells notwendig sind. (Für eine Gerade beispielsweise 2 zufällige Datenpunkte).
2. Ermittlung der Modellparameter (Bei einer Geraden beispielsweise a und b der Geradengleichung $y = ax + b$)



POINT	x	y
1	0	0
2	1	1
3	2	2
4	3	2
5	3	3
6	4	4
7	10	2

Figure 3: Angleichung einer Geraden an eine Punktwolke mit hohem Ausreisser (Fischler und Bolles 1981, p. 382)

3. Ermittlung der Anzahl Punkte, die weniger weit weg vom Modell sind, als ein bestimmter Schwellwert. (eM-Score)
4. Schritte 1-3 n mal wiederholen und das beste Modell bestimmen.

Schlussfolgerung

Da ich einen starken Bezug zum Thema Musik habe, war ich fasziniert als ich vor mehr als 10 Jahren Shazam, ein Musikerkennungsdienst, entdeckt habe. “Computer Vision for Music Identification” erklärt auf anschauliche Weise, wie zuverlässige Systeme zur Identifizierung von Songs gebaut werden können. Durch dieses Paper und der Konsultation von weiteren Quellen, verstehe ich nun, wie ein Service zur Musikerkennung funktionieren kann.

Die Anwendung von Bilderkennungsverfahren zur Identifikation von Musik ist ein effektives Verfahren. Es zeigt eindrücklich, wie durch die Adaption von bewährten Methoden auf ein neues Anwendungsgebiet hervorragende Resultate erzielt werden können. Interessant ist auch einmal mehr zu sehen, dass die Wahl des richtigen Algorithmus grosse Auswirkungen auf die Effizienz haben kann (siehe Abschnitt Abfrage).

Das vorliegende Paper ist aus dem Jahr 2007. Obwohl man das Problem der Musikerkennung gelöst hat, wird an diesem Thema weiter geforscht. Mittlerweile wurde diese Methode, welche hier vorgestellt wurde, auch weiterentwickelt (siehe Jang u. a. 2009).

Referenzen

Fischler, Martin A. und Robert C. Bolles. 1981. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM* 24, Nr. 6: 381–395.

Jang, Dalwong, Chang D. Yoo, Sunil Lee, Sungwoong Kim und Ton Kalker. 2009. Pairwise Boosted Audio Fingerprint.

Ke, Yan, Derek Hoiem und Rahul Sukthankar. 2005. Computer Vision for Music Identification.