

多次元データ構造を用いた音楽の類似検索

2Q-4

中川 由香 中村 泰明
広島市立大学 情報科学部

1. 序論

近年、マルチメディア技術が身近になるにつれ、自動作曲や演奏の表情付けなど音楽情報処理に関する研究¹⁾が盛んになっている。しかし、マルチメディアデータの検索は、データが記号による記述でないことや、音声や画像そのものが多義的であることから困難となっている。

本稿では、楽曲の特徴をベクトルとして表現し、類似性を定義した上で、あるメロディを入力としたとき、これに類似したメロディを含む楽曲を、データベースの中から検索するシステムを構築する。

2. 特徴ベクトルによる類似検索

2.1 特徴ベクトルによる検索手法

楽曲データベースに蓄積されているデータは、タイトル、作曲者等の文字列で表される書誌データと、楽曲そのものの表現である音程、音長などの情報を持つ。ここでは、楽曲の特徴をベクトルで表現し（以下、特徴ベクトルと呼ぶ）、検索キー特徴ベクトルが入力されると、データベース中の検索対象データの特徴ベクトルとの距離を計算し、距離の近いものが検索結果として出力される。すなわち、ベクトル空間内の距離の近さを類似度と見なす。この距離に重みのついた N 次元ユークリッド距離を用いる（但し検索時間のコストを減すため、平方根はとらない）。以下に距離 D を数学的に定義する。

$$D = \sum_{i=1}^N |x_i - s_i|^2 \times w_i \quad (1)$$

ただし、 x_i は検索キー、 s_i は検索対象データを表し、また、 w_i は各次元の相関的な重みを表している。この重みを変えることによって、多様なユーザの検索要求に応えることができるようなシステムを構築した。

Similarity-based Retrieval of Music using the Multi-Dimensional Data Structure.

Yuka Nakagawa and Yasuaki Nakamura

Faculty of Information Science, Hiroshima City University



音程	72	72	69	72	72	69	72	69	65	69	65
音程の差		0	-3	3	0	-3	3	-3	-4	4	-4
音長	24	-12	12	24	-12	12	12	12	12	12	24
音長の比		0.5	1	2	0.5	1	1	1	1	1	2

図 1 楽曲の特徴量表現

表 1 音長における音符の数値化

全音符	96	符点8分音符	18
符点2分音符	72	8分音符	12
2分音符	48	符点16分音符	9
符点4分音符	36	16分音符	6
4分音符	24	32分音符	3

2.2 楽曲の特徴量表現

従来、演奏情報から抽出される特徴量は、自動作曲や自動演奏のための知識ベース作成などに用いられていた。このような特徴量を、楽曲データベースの検索に応用する。

楽曲を構成する要素として、人が音楽を聴くとき、特に印象が強く、楽曲を特徴付ける大きな要素とされる音の高さ（音程）と音の長さ（音長）の二つを特徴量として用いる。

(1) 音程の差

中央のドを 60 とし、半音の差を 1 として数値を割り当てる。休符は、その前の音と同じ数値を割り当てる。これは、音程では、休符の有無で特徴に変わりはないと見なし、休符を無視するためである。これにより連続する 2 音の差を求め、それを特徴量とする（図 1）。

(2) 音長の比

表 1 に従って各音符に数値を割り当てる。休符は、音符にそれぞれマイナスをとったものとする。音程と同様に、このデータを用いて連続する 2 音の比の絶対値を求め、それを特徴量とする（図 1）。

3. 類似音楽検索システム

3.1 検索システムの概要

検索システム構成を図2に示す。検索内容は、楽曲データベースに蓄積されている曲の中から、入力されたメロディに対して、類似メロディを含む曲を検索し、出力するというものである。一つの曲に対して旋律データと曲名、作曲者名、拍子、調で構成される楽曲データベースを作成する。旋律データに対して、特徴量を求め、特徴ベクトルに基づいて多次元空間データ構造 SS-tree^[3]を用いて類似検索を行う。

3.2 特徴ベクトルの算出

曲のイントロだけでなく、どの一部分との類似検索も可能にする。そのため、各曲の曲の始まりからN個の音符をひとまとまりとしたものを、一音ずつずらして特徴ベクトルとする(図3)。これにより、一曲に対する特徴ベクトル数は、ほぼ音符の数と等しくなる。

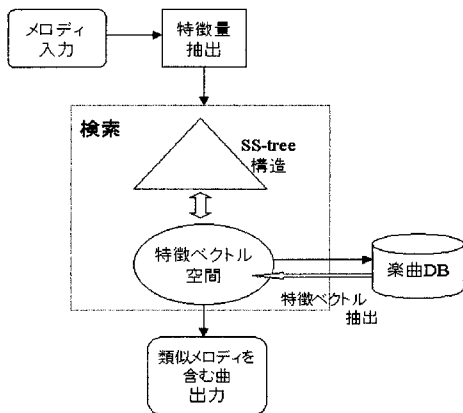


図2 検索システム構成

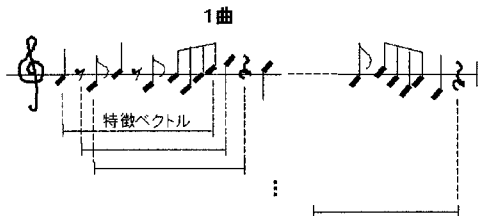


図3 1曲における特徴ベクトル

3.3 多次元データ構造 SS-tree

特徴ベクトル数が莫大になるため、検索を高速化するデータ構造が必要となる。そこで、高次元空間での高速な最近接検索が可能な SS-tree を用いる。

これは特徴空間を階層的な部分領域(包囲球)に



図4 検索結果画面

分割することで、空間を局所的に検索するだけで類似のベクトルを得ることを可能とし、さらにベクトル数が大規模になっても高速性を維持できる。

4. メロディ検索の実行

あるメロディを入力としたときの検索結果画面を図4に示す。入力に対して類似度の高いものから10曲を、入力メロディと類似メロディとの距離と共に求める。また、最も類似度の高い結果の譜面を表示する。本実験では、10音をインデックスとしているため、類似メロディとして10音が表示されている。この結果は、入力されたメロディに対し、非常に近いものが検索されたと言える。

5. 結論

本稿では、特徴量を抽出することによって音楽の類似検索を行うシステムを構築した。音楽に対する解釈は人によって異なることから、ユーザの要求にあわせて類似度の尺度を設定可能とした。

今後は、大規模な楽曲データベースに対する類似検索を実現し、また、重みの決定に関するさらなる検討が必要である。

参考文献

- [1] 赤間浩樹, 紺谷精一, 鬼塚真, 山室雅司, 串間和彦: “多次元空間索引と画像・音楽の内容検索”, bit, vol.32, No.9, pp.47-54, 2000
- [2] 平田圭二, 青柳龍也: “パーピープン: ジャズ和音を生成する創作支援ツール”, 情報処理学会論文誌, vol.42, No.3, pp.633-641, 2001
- [3] D.A.White and R.Jain: “Similarity Indexing with the SS-tree”, Proc. of the 12th Int. Conf. on Data Engineering, pp.516-523, 1996