

音楽データライブラリー音声入力検索システム 「口三味線システム」の設計

Designing of a new music library data base system, "Kuchijamisen system",
an idea to look up a tune with melodic lines inputted by voice.

清 道 正 嗣

Masatsugu Seidou

要約：

この論文では、音楽データのライブラリーのデータベースから、音声入力した検索キーを使って特定の曲を見つけ出すためのシステムに関するアイデアを述べている。このシステムでは、検索に用いるフィールド値と検索キーを、音階上での音符の上下運動を表現する形式で符号化する。これによって特に音楽の訓練を受けていない人間でも、このシステムを使うことが可能となる。

Abstract：

This report describes an idea for a data base system to look up a tune from a music library. Melodic lines inputted by voice are used as the searching key in this system. Both the inputted melodic lines and the field values recorded on the data base are encoded to the progressions, which express the movements of notes on the music scale. Because of the encoding, users need not to be trained in singing especially for using the system.

はじめに：

本論文は、94年度学長特別教員研究費による研究「芸術系大学におけるデータベース構築法・教育法に関する研究」の成果報告である。しかし以前に研究報告書に記載したものと同一ではない。報告書に記載したデータベース（DB）・データベースマニピュレーションシステム（DBMS）は、本学で現在入手可能なデータを使って一般的な計算機システムとデータベース構築ソフトによって、実現するというものであった。本論文では、今のところは本学資料館・図書館に存在しないフィールド値を用いるDB・DBMSについて

検討する。

一般の人が芸術系大学の資料館・図書館のDBを使って、CD・楽譜など音楽を記録しているもののライブラリーから求める曲を見つけようとするときに、手助けとなる新たな検索手段を提供できるのではないかと思いついた。それは、「題名も作者も分からないが、こんな曲を誰か知らないか？」という要求に答える能力を、DB・DBMSに与えようというものである。このような要求に応じた検索の実現は、歌詞のあるものに対しては容易である。DBに歌詞を登録しておき、覚えている歌詞を検索キーにして文字列の検索を行うDBMSを作れば良い。しかし歌詞のないものをも検索対象とする場合は、別の方法が必要である。

この点に対して、今回一つの解答を提示する。曲データから得た符号列のフィールドを各レコードにつけ、声によって入力した検索キーを使って検索するというものである。日本では歌って楽器のまねをすることを古来の慣用表現で「口三味線を弾く」と称することから、このシステムを以下では「口三味線システム」と呼ぶことにする。

システムの概要：

以下に「口三味線システム」の概要を示す。このシステムは、三種類の異なる方法で構築することができる。第一の方法は、MIDI楽器・MIDIアンプをコンピュータに接続し、これに音声の入力から数値化までを行わせ、コンピュータではMIDIデータからの変換とDB検索だけを行う。第二の方法では、音声データの入力から数値化までを行う専用のハードウェアとコンピュータを用意し、専用ハードウェアのコントロール・音声データ処理と検索をソフトウェアによって行う。最後の方法は、音声入出力端子を持つコンピュータを使い、音声データ処理と検索をソフトウェアによって実現するというものである。

このうち第一の方法は、音声データのサンプリングに関する技術上の問題を、すべてMIDI機器に任じてしまえるという利点がある。問題は、MIDIの規格から少なくとも12音階で表現できる曲でなければ、検索性データの記録が不可能であるという点である。しかし例外的な曲は回避するようにあらかじめ取り決めておけば、重大な欠点とはならない。

第二の方法のハードウェア部分は、音声入力装置・一時記憶装置・周波数カウンターとコンピュータで作ることができる。この方法では、あらかじめ決められた規格に縛られることがないので、12音階で表現できないデータも処理することができる。しかし専用の器材を必要とするので、試験的に製作するには問題はないが、広く一般に試験運用を行ってもらったり実用に供したりするには、システムの供給に問題がある。

第三の方法は、コンピュータ本体以外にハードウェアを必要としないので安上がりである。第二の方法と同様に、12音階の制限もない。ただ用いるコンピュータには音声入出力端子（あるいはスピーカ・マイク）があることが必要であり、そのハードウェアに関する情報が入手できることが不可欠である。当然ソフトウェアは、機種に大きく依存するという欠点がある。

三種類とも、動作は基本的に同じである。口三味線システムでは、DBの検索は以下に示すように行われる。まず検索者に、検索キーとなる音声データを入力してもらう。このデータを半導体を用いた一時記憶装置（コンピュータの本体中のRAMあるいは、入力装置側に付けた半導体メモリー）に蓄える。記録したデータを再生し検索者に聞かせることにより、検索者の意図したフレーズが録音されたことを確認する。このデータを時間軸に沿ってサンプリングし、周波数あるいはMIDIノートナンバー [1] の数列にする。この数列を後述の Protokol によって処理し、検索キーを生成する。あらかじめ同じ Protokol で作りDBの各レコードの検索用フィールドに記録しておいたデータに対する、この検索キーの一致度を求める。一致度に対してあらかじめ決めておいた判定基準に基づいて、検索結果を表示する。

12音階に制限されない曲を扱うシステムは、自由度が高く技術的側面からも興味深いが、実装が容易ではないことは想像に難くない。またこのような曲に対する検索能力は、一般の利用者による需要がほとんど見込めない。今回は、より簡単に実装でき需要も見込める、12音階で表現できる曲のみを扱うDBシステムについて述べることにする。

検索用データフィールド値・検索キーのデータ形式：

検索用のフィールドに登録しておくデータは、図1に示すように加工する。まず実際の

元のデータ	
音の長さを除去したデータ	
MIDIノートナンバー	78 83 79 78 76 74 72 74 72 71 69
ノートナンバーの差	5 -4 -1 -2 -2 -2 2 -2 -1 -2

図1 検索用データの記号化

例に示した曲データは、J.S.BachのOuverture Nr.3 D-dur BWV1068のAirの第一ヴァイオリンの始めの2小節である。まず音の長さに関する情報を捨て、連続する音の並びにする。MIDIのノートナンバーの定義に従い、各音を符号化する。各符号を数値とみなして始めから順に差を取り、このDBシステムで用いるデータを得る。

曲を人間の口で歌えるように単音の連続したものに加工した後、音程だけを残し音の長さの情報を捨てる。音の高さをMIDIノートナンバーで表し、これを数とみなして順に前の音との差を取り、結果を数列として記録する。この数列は、もとのデータの音の上下運動を、12音階の1音差を“1単位”として示している。検索時に入力されたデータも同様のプロトコルで処理し、検索キーを生成する。

曲データをこのように変換することは、音楽に関する訓練を特別につんでいない人間でも、このシステムを利用できるようにするために必要である。譜面の通りに正確に弾いたり歌ったりする必要のあるシステムでは、音楽家以外には利用できないことになる。今回は声による入力を想定しているので、特に音程とリズムの正確さに可能な限り依存しない形式での符号化が必要である。この点に関して今回の符号化の方法には、以下の2つの利点がある。

まず第一に、人間の奏でるリズムには意識・無意識の別はあってもかなりの揺らぎが存在するので、譜面上の記述と一致しない。このことからリズムを検索の対象に含むことは、不必要に検索精度を落とすことになる。リズムを取り去ることは、このような揺らぎを考慮に入れなくてすむという利点がある。

つぎに、人間は自分が発声できない音域のフレーズを口ずさむとき、しばしば調を移して歌う。ときには曲を、この歌える調で覚えていることがある。また特別に訓練をつんでいない人間にとっては、正確な音程を表現することは困難である。一方、音の上下運動を表現することは、それほど困難ではない。いずれの場合も検索キーの入力値は、譜面と一致しない。しかしこのような入力値は、音程の上下方向にシフトさせると元の曲の相当する部分と一致する。あらかじめ音の高さに依存しない方法で曲を表現しておけば、調を無視することができるので、正しい調を探すプロセスを省き検索に必要な計算時間を節約することができるという利点がある。

このデータ形式は元の曲を不可逆的に変換しているので、検索用データフィールドからの情報だけでは決して元の曲が復元できない。このことはDBの運用面で、「DBの配布を行っても、著作権に対する侵害を回避できる。」という二次的ではあるが便利な特性を持つことを意味している。

DB登録用フィールド値の入手法：

コンピュータミュージックの普及により、今回のようなDBに登録しておくデータの入力作業は、かなり軽減できるようになってきた。様々な曲のMIDIデータが、市販品あるいはパソコン通信から入手可能である。MIDIデータはその規格 [1, 2] から、口

三味線システムのデータに容易に変換できる。またMIDI以外の形式による曲データもかなりあり、それらも今回のデータ形式に容易に変換できる。

検索用フィールドには、計算機資源が許す場合は入手した曲データの全てを入れるということも考えられるが、曲の出だし・主題・サビなどその曲のなかで印象の強い部分を適当な長さに切って用いても十分検索目的には使えるであろう。曲の一部を用いるときには、人間が適当な部分を選び出す。この選択を行うには作業する曲に対する理解が必要で、特別に音楽教育を受けたものでなければ困難であろう。このような点に、本学のような芸術系大学の協力は大きな助けとなるであろう。

既存のデータが利用できない曲の場合、人力による曲データの入力が必要となる。このような場合も本学には音楽学部があるので、曲データ入力のための作業要員の確保は、経済的な問題を除けば容易である。

データの検索法：

このDBの検索には、通常の正規表現を使ったパターンマッチングでは不十分である。なぜなら人間の記憶するフレーズには、音の置換・欠失・挿入がみられることがあり、なおかつ本人にはそれが分からないからである。どこに不確定な要素があるか分からないキーを、正規表現であらわすことはできない。このような検索には、むしろ生命科学系の情報検索システムで相補性・

ホモロジーの検索に使われているアルゴリズム [3] のようなものが適している。これは、二つの連続した記号列から、互いに最も一致する場合の一致度を求めるという方法である。

この方法でキーに対するDBのフィールドの一致度を求め、ある閾値以上をとったもの、あるいは上位データを提示するようにすれば良い。利用

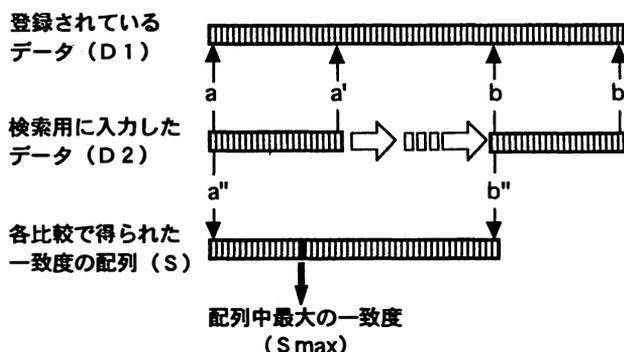


図2 一致度の求め方

入力したデータ (D2) を、一致度を求めるレコードに登録してあるデータ (D1) と、a-a' に示す位置に並べる。対応する要素間で、比較を行い一致している要素の数を求める。この数を検索キーの全要素数で割った値を、一致度 a'' として記録する。この操作を、D2 をデータの下流に向かって (白ぬきの矢印方向) 1 要素分づつシフトさせながら b-b' まで繰り返す。得られた一致度の配列 (S) から、最高値 (Smax) を求め、これをこのレコードの一致度とする。

者の要求に適度な正答率で回答するには、あまり正確な一致度を求めることは検索効率を悪化させるだけで無意味である。ほとんど同じに聞こえるフレーズを持つ曲はそんなに多いわけがないので、利用者への回答を複数提示することにすれば、かなり簡略化した一致度の計算で良いことになる。今回報告するシステムでは生命科学系の情報検索システムとは異なり、図2に示すような入力値への挿入・欠失を考慮しない単純な入力値と登録値の比較で十分に実用に耐えると思う。

回答提示のための選択基準には、閾値を用いることが望ましい。上位データを求めるにはソーティングが必要であるが、これに必要な検索時間はDBのレコード数の増加により加速度的に増大する。そのような処理を行ってまで、利用者の求めるものを最上位に示す必要性はどこにもないからである。

閾値を使うには、あらかじめ基準を設定することが必要である。これには、音の置換・欠失・挿入などの誤りを入力フレーズ中にどの程度許すかを決定すれば、ある程度の判断の基準となる値を得ることができる。今回の一致度の計算法であれば、入力フレーズ中に存在する置換・欠失・挿入の数を決めると、最悪の一致度が単純に計算できる。図3に示すように音の欠失・挿入は、誤りより下流の部分に位置ずれをおこすことから、置換に比

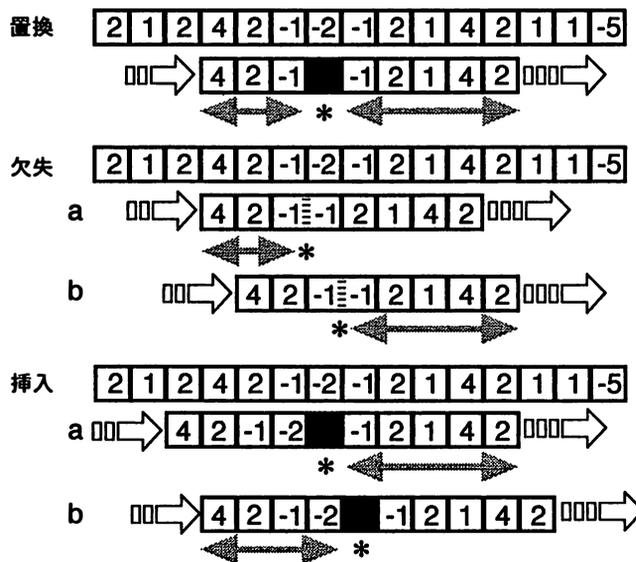


図3 検索キー中の誤りの一致度への影響

置換の場合には、誤りは検索キー中の数に応じて一致度を単調に低下させる。欠失・挿入の場合には、誤りは生じた場所より下流側に要素の位置ずれを引き起こす。このため一致度は、誤りの数が同じでも誤りの生じた場所によって変化する。

各例は、最上りがDBに記録されている符号列、それ以外が入力データから生成された符号列を示している。*は誤りの位置、灰色の両矢印は登録データと一致しているところを示している。

べて一致度により悪い影響を与える。図3-a・bに示すように欠失・挿入では、誤りによって分断された2つの連続した正しい部分の長さは、互いに検索入力全体から相手を引き去った長さとなる。このため最悪の一致度を与える欠失・挿入は、それが検索キー中に均等に現れたときに起きる。誤りによって分断されたキー中で最も長く連続した正しい部分の長さが、この場合に最低となるからである。その一致度は、検索キー中の誤りの数の逆数である。この値は、検索キーの要素数が奇数であるか偶数であるかによってわずかに異なるが、誤りが1個のときは約50%、2個のときは約33%、3個のときは約25%になる。これは最悪の場合なので、同じ誤りの数を持つ大抵の入力は、もっとましな値を得るはずである。

置換とそれ以外の誤りでは、誤りの一致度への影響の大きさが異なるので、閾値の値は試験運用を行って調整すべきである。始めは2～3箇所の欠失・挿入があるときの最悪の場合の値を使って、順次検索に対する回答数を見ながら最適値を求めることが良いだろう。もし試験的に運用した結果、置換に対する判定基準が相対的に甘いことが原因で、回答数が多すぎるという不都合が頻繁に起きる場合には、閾値を置換に対するものとそれ以外用に2つ決め、二段階に回答を提示する必要があるかもしれない。

予想されるシステムの問題点：

検索データを人間の口で歌えるフレーズにするということに対して、予想される問題点が2つある。

あまりにポリフォニックで人によって聞き取り方の異なることを許す（あるいは期待する）ような曲は、このシステムでは対処できない。このような曲からDBの検索用のフィールドに登録しておく実用的なデータを作成することは極めて困難であり、場合によっては実際には作成できない可能性があるからである。このような曲への対応としては、「表現困難」を意味する符号を決めてDBに登録しておき、音声ではなくキーボード・マウスなどで、この符号を利用者に入力してもらうことにするのが良いように思われる。このような曲とは別に、重音が連続していても口三味線で歌える曲は存在する。この場合に歌われるフレーズには、個人差があまりあるようには思えない。このような曲に関しては、前述の例とは異なり、重音を単音に置き換えてしまっても問題はないと思う。

もう1つの問題点は、利用者の入力音声の音階があまりに不安定な場合には、いつまでも利用者が検索用のキーを作成できなかったり、検索を行っても一致度が低すぎて回答が出なかったりする可能性があることである。このような場合に備えて、容易に操作できる入力用の装置（既存の楽器・専用デバイス）を用意することも必要であるかもしれない。

謝辞：

この研究は、1994年度学長特別教員研究費によって援助されました。一海孝光先生・中村洋子先生（愛知県立芸術大学教養教育）には、この論文の原稿を読んでいただき、多くの助言をいただきました。この場を借りて、お礼申し上げます。

参考文献：

1. MIDI 1.0 Document ver.4.1 日本語版 (1989), M I D I 規格協議会
2. Standard MIDI Files 0.06 (1988), Dave Oppenheim
(日本語翻訳版, (OZ (AV00380)・川村優子 (AV02215) 翻訳, NIFTY-Serve FMIDI より入手。))
3. Principles of Protein Structure (1979), G.E.Schulz and R.H.Schirmer, Springer-Verlag GmbH & Co. KG