論　文

WWW 上での歌声による曲検索システム

園田　智也[†]　　後藤　真孝[††]　　村岡　洋一[†††]

A WWW-based Melody Retrieval System

Tomonari SONODA[†], Masataka GOTO[††], and Yoichi MURAOKA[†††]

あらまし　本論文では、WWW上で動作する、歌声の旋律からその曲のタイトルを検索するシステムについて述べる。歌声による検索では、入力の旋律情報（音高・音長）が正確とは限らないため、閾値によってそれらを粗い旋律情報に変換したものを検索キーとし、データベースの曲とのマッチングを行なう。しかし、このための適切な閾値の設定は難しく、特に音長情報においては、有効な検索キーを得ることが困難であった。また、粗い旋律情報では正答の絞り込みも難しい。そこで、本研究では（1）有効な検索キーを得るための最適な閾値を設定する手法、（2）データベースの曲から正答の曲の候補を精度良く絞り込むためのマッチング手法の2つを提案することで、従来手法よりも正答率の高い検索を実現し、WWW上で複数の利用者が活用できるシステムを構築できた。

キーワード　音楽データベース、メロディ検索、WWW アプリケーション、旋律情報

1. まえがき

本論文では WWW ブラウザ上で歌声を入力し、ネットワーク経由で曲データベース (DB) サーバからその曲のタイトルを検索するシステムについて述べる。WWW 上には多くの曲 DB が公開されているが、これまで、それらに対する主な検索手法は歌詞などを文字で入力して行なうものであった。しかし、耳で聞いて記憶している旋律を検索に用いる方がより直感的である。そこで、本研究では歌声の**旋律情報** (音高と音長の 2 つの属性値を持つ音符の系列) を用いた検索を WWW 上で実現することを目的とする。

WWW 上で歌声による曲検索システムを構築するには、様々な利用者に対して、精度良い検索を行なう必要がある。また、ネットワークの負荷を軽減するため、転送するデータサイズは小さい方が望ましい。

従来の曲検索システム [1]～[6] はいずれも 1 台の計算機上で実装され、ネットワーク上での利用は考慮されていなかった。歌声の旋律情報は、DB 中の曲のものとは正確に調・テンポが一致するとは限らないため、文献 [2]～[6] では、旋律の音高や音長の系列を音符間の**相対音高差・相対音長比の系列** に変換してから、検索に用いていた。さらに、利用者の記憶違いや歌唱能力による誤差を許容するため、相対音高差に対しては、前音から「上がった、同じ、下がった」、相対音長比に対しては、「長くなった、同じ、短くなった」などのように、**粗い精度の相対値**を変換したものを検索キーとしていた。

しかし、粗い精度の相対値を用いる場合、閾値によってある範囲の値を同一の値とみなして検索キーを生成するために、その閾値や検索効率に関して以下の問題があった。

（1）　有効な検索キー生成のための適切な閾値の設定が難しかった。特に音長に対しては設定が困難であり、従来研究では有効に用いられていなかった。

（2）　粗い相対値を検索キーとした検索では、DB 中の曲も同様に粗い相対値に変換してマッチングを行なうため、複数の曲で同様の旋律パターンを持つことがあり、正答の曲が精度良く絞り込めないことがあった。

そこで、本研究ではこれらの問題を解決するために、以下の 2 つの手法を提案する。

（1）　曲 DB のすべての曲中に出現する音高・音長の分布を利用して閾値を決定することで、最も効率的に正答を絞り込む検索キーを生成する手法。

（2）　検索キーの相対値の粗さを徐々に細かくしな

[†] 早稲田大学理工学部 , 東京都
School of Science and Engineering, Waseda University, Tokyo, 169-8555 Japan

がら検索を行なうことで、正答の曲を効率良く絞り込むマッチング手法。

以上の手法により、音高・音長の両者を有効に用いた精度の良い検索を実現することができた。また、旋律入力用の WWW ブラウザのプラグインを開発し、歌声の膨大な音響信号を少量の旋律情報に変換したことで、ネットワークにかかる負荷の少ない検索システムを構築することができた。

2. WWW 上での曲検索システム

本システムの概要を図 1 に示す。本システムはサーバ･クライアント型のシステムであり、利用者は WWW ブラウザのクライアント側で曲のメロディ（旋律）を歌うことで、サーバの保有する DB の曲の中から、最も似ている旋律を持つと解釈された曲のタイトルを得ることができる。

以下ではサーバ・クライアントそれぞれの処理の概要について述べる。

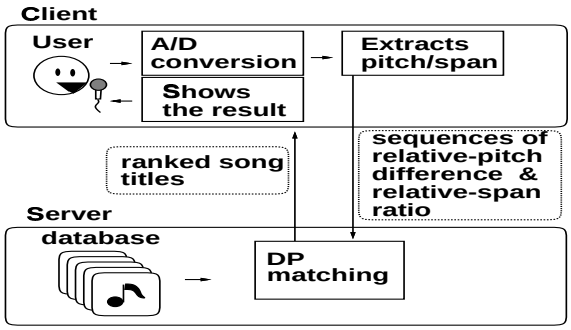

図 1　システムの概要
Fig. 1　Image of the system.

2.1　クライアント (音高・音長抽出)

クライアントにおける処理の中心となる、各音符の「音高 (pitch)・音長 (span) の抽出」について述べる。クライアントは、歌声の音響信号の A/D 変換後、まず、処理の単位時間（フレーム、16msec）ごとに有声音の判定を行なう。次に、有声音のフレームを利用し、各音符の音高・音長の同定を行なう。最後に、音高・音長の系列を相対音高差 (relative-pitch difference)・相対音長比 (relative-span ratio) の系列に変換してサーバに転送する。その際、膨大な音響信号データをデータサイズの小さい旋律情報にクライアント側で変換（具体的には、WWW ブラウザ上のプラグインで変換）することで、ネットワークの負荷を小さくできる。最後にサーバから検索結果である曲のタイトルを受け取ると WWW ブラウザ上に表示を行なう。

クライアントの各処理を以下に示す。

- **歌声入力**

歌声の入力では、利用者はマイクを使用し、自由な音高、自由なテンポ、自由な歌い出し（曲中の位置）で入力することが可能である。歌い方は、歌詞を歌わずに、ta-ta-ta, cha-cha-cha 等のように、1 音符を無声音かつ破裂音で始まり 'a' の有声音で終る 1 モーラに対応させたものに限定する。この形式により、システムは安定して旋律情報を抽出でき、利用者にとっても、この歌い方は無理がないために特に負担とはならない。

- **有声音の判定**

歌っている最中はマイクには周囲の環境音が入らないことを前提に、歌声の音響信号の振幅が一定値以上のとき、有声音と判断する。

- **音高・音長の同定**

連続する有声音のフレームのうち、一番最初のフレームを音符の発音時刻とし、各音符を発音時刻によって区切る。さらに、各音符の発音時刻と、次の音符の発音時刻との時間差（フレーム数）をその音符の音長とする。次に、各音符に対し、音長として定められた区間における各フレームのピッチを、以下の手順で求める。

まず、各フレームに対し、FFT (サンプリング周波数 8 kHz, 512 点) の結果から得られた、音声波形のパワースペクトルを用い、スペクトル包絡のピークを抽出する。抽出した m 個のピークのうちの i 番目のピーク P_i $(0 \leqq i < m)$ (Hz) に対して、P_i がそのフレームでのピッチであることの確信度 $R(i)$ を設け、その初期値を $R(i) = P_i$ としておく。ここで、ピーク P_i と 2 以上の整数 N、十分に小さな値 ϵ (Hz) に対し、

$$NP_i - \epsilon < P_k < NP_i + \epsilon \quad (i < k < m)$$

となる P_k が存在するたびに、P_i が倍音 P_k を持つと判断し、確信度 $R(i)$ の値を次の計算で増加させていく。

$$R(i) \leftarrow R(i) + P_k$$

この計算の結果、すべての i $(0 \leqq i < m)$ に対して確信度 $R(i)$ が最も高くなった P_i を、そのフレームのピッチとする。

以上のようにして求めた、各音符の音長区間における各フレームのピッチのうちの最大値を、その音符の音高とする（予備実験から最大値の方が平均値よりも精度良くピッチを同定できた）。

- **相対音高差・相対音長比の送信**

同定した音高・音長の系列を、それぞれ相対音高差・相対音長比に変換し、サーバに送信する。相対音高差は半音の差が 100 となるように正規化し、相対音長比は前音の音長との比をパーセンテージで表す。

2.2 サーバ（検索）

サーバは曲 DB を保有しており、曲の「検索」処理を行なう。検索には DP マッチングを用い、クライアントからの入力の系列と DB 中の各曲中の系列との距離を求める。マッチングの結果、入力の系列と最も距離が近い系列をもつ曲から順に正答の候補とし、クライアントにそのタイトルのリスト（曲数は任意に設定できる）を送信する。

サーバの各処理を以下に示す。

- **曲 DB**

本システムでは曲 DB を歌声で作成し、歌声の入力と同様の方法で、各音符を相対音高差・相対音長比の系列として保存しておく。歌声で DB を作成する利点として、楽譜形式での手入力よりも DB 作成の労力が少ないこと、楽譜のない曲でも容易に DB に追加可能であることが挙げられる。

- **DP マッチング**

DP マッチングでは、入力と DB 中の各曲の旋律の系列間の距離を求める。マッチングには、相対音高差・相対音長比を、それぞれ閾値によって粗いカテゴリに分割して、粗い精度に変換した相対値を用い、比較する粗い相対値同士の差を DP マッチングにおけるペナルティとし、その合計値を系列間の距離とする。具体的には次のように行なう。

入力の歌声の旋律を粗い精度に変換した、長さ I の相対音高差・相対音長比の系列を QR_p・QR_s とする。また、DB 中のある曲が持つ旋律を粗い精度に変換した、長さ J の相対音高差・相対音長比の系列を SR_p・SR_s とする。入力の系列 QR_x と DB 中の曲の持つ系列 SR_x についての距離 $D_x(QR_x, SR_x)$ を DP マッチングによって求める（ここで、$x = p, s$ とする）。本システムでは SR_x のすべての箇所を開始点として QR_x の先頭からのマッチングを行なうことが可能となっている。

粗い精度の相対音高差・相対音長比が取り得る整数値の集合を、それぞれ C_p・C_s とし、相対値間のペナルティ d_x を 次のように定義する。

$$
\begin{aligned}
QR_x \quad & = q_0, q_1, q_2, \ldots, q_{I-1}\ , \quad q_i \in C_x \\
SR_x \quad & = s_0, s_1, s_2, \ldots, s_{J-1}\ , \quad s_j \in C_x \\
d_x(a,b) \quad & = |a-b| \\
& : a \text{ と } b \text{ の間のペナルティ}, \ a, b \in C_x \\
& (\text{ただし } x = p, s)
\end{aligned}
$$

DP マッチングの計算は 2 次元のマッチング行列 $m(i,j)$ を用いて行なう。各要素の計算は図 2 のように、はじめに 1 行目の計算（Step A）、次に 1 列目の計算を行なう（Step B）。それからすべての要素について計算を行ない（Step C）、最後に一番下の行の最小値を用いて距離 D_x を求める（Step D）。以上の処理を DB 中のすべての曲について行ない、入力の系列との距離 D_x を求める。

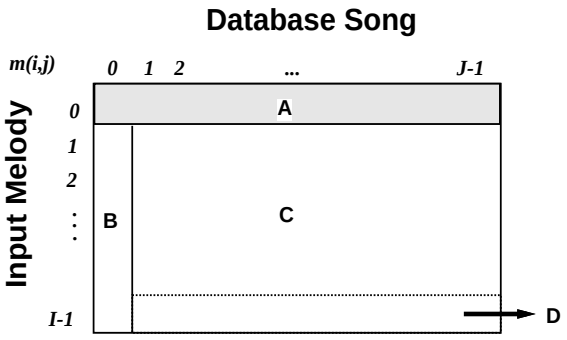

図 2　DP マッチング行列 $m(i,j)$
Fig. 2　DP matching matrix $m(i,j)$

以下に各 Step A～D の計算手順を示す。

– **Step A の計算**

$0 \leqq j < J$ において

$$m(0,j) \quad = \quad d_x(q_0, s_j)$$

– **Step B の計算**

$0 < i < I$ において

$$m(i,0) \quad = \quad \infty$$

– **Step C の計算**

$1 \leqq i < I,\ 1 \leqq j < J$ において

$$m(i,j) = d_x(q_i, s_j) + min(\begin{array}{l} m(i-1,j), \\ m(i-1,j-1), \\ m(i,j-1)) \end{array}$$

– **Step D の計算**

$$D_x(QR_x, SR_x) = \min\{m(I-1,j) \,|\, 1 \leqq j < J\}$$

3. 実現上の課題と解決法

本システム全体の検索精度に大きく影響を与えるのがサーバの DP マッチングに利用する情報の精度である。利用者の歌声の変動や誤差の影響を吸収するために、従来研究 [2]〜[6] では、いずれもアドホックな閾値を利用し、相対音高差・相対音長比を粗い精度の相対値に変換して、マッチングに用いる検索キーとしていた。例えば、文献 [6] では、音高において、前音より「上がった、同じ、下がった」ということを表す値を U,E,D のなどのような 3 つの記号で表し、「ドレミミレド」を「XUUEDD」などと変換したものを用いた (X は最初の音なので相対値がないことを示す)。しかし、検索に効果的な閾値を設定することは難しく、特に音長に対しては有効な検索キーを生成することができなかった。また、粗い精度の相対値だけでは、正答を絞り込むことが困難であった。

以下、**3.1, 3.2** では閾値の決定問題とその解決法を述べる。また、**3.3, 3.4** では正答の絞り込み問題とその解決法を述べる。

3.1 検索キー生成のための閾値決定問題

図 3、4 はそれぞれ、歌声によって作成した DB (**6.3** の実験で用いたものと同じもので、日本・西洋のポップス、童謡、民謡、演歌などのジャンルから選んだ 200 曲) の曲中に出現する、すべての音符の相対音高差・相対音長比のヒストグラムである。相対音高は半音の差が 100 となるように正規化しており、相対音長は比をパーセンテージで表現している。ここで、図の上部に従来研究 [2] で用いられた検索キー生成のための閾値を示す (図の下部の動的閾値 (dynamic thresholds) については **3.2** で述べる)。文献 [2] では、相対音高差に関しては、前の音符から半音の幅の差を閾値とし、相対音長比に関しては、50% と 200% を閾値としていた。 ここで、閾値によって変換される粗い精度の相対値を表す記号として、音高には「上がった (Up)、 同じ (Equal)、 下がった (Down)」を表すカテゴリ U,E,D を用いる。音長には「長くなった (Longer)、 同じ (Equal)、 短くなった (Shorter)」を表すカテゴリ L,E,S を用いる。これらの図を見ると、音高に関しては U,E,D それぞれのカテゴリにほぼ均等に変換されていた。このため、DB 中の曲の系列に出現する U,E,D は、各々ほぼ等しい確率で出現する。マッチング処理の際には、検索キーとして入力される系列中の各音符の相対音高差に対しても、同じ閾値を用いて U,E,D の 3 つカテゴリに変換することで、入力系列中の 1 つの音符ごとに、正答として絞られる DB 中の曲の候補数は、約 1/3 ずつになっていくことが期待され、望ましい。

一方、音長に関しては、多くのものが E に分類され、L,S に分類されるものは比較的少なかった。このため、DB 中の多くの曲の音長の系列は、E が L,S に比べ多く出現する系列に変換されていた。この結果、

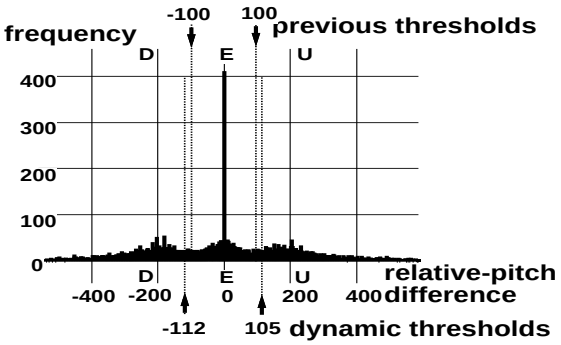

図 3 相対音高差の分布 (200 曲) と閾値
Fig. 3 Histogram of relative-pitch difference (in 200 songs) and thresholds.

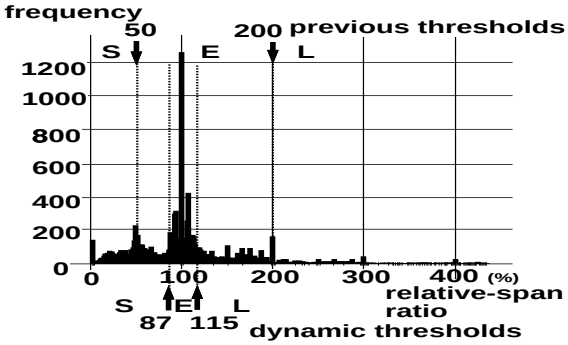

図 4 相対音長比の分布 (200 曲) と閾値
Fig. 4 Histogram of relative-span ratio (in 200 songs) and thresholds.

複数の曲で同様のパターンが出現する確率が高くなり、正答の曲を絞り込むことが難しかった。

以上のように、DB 中の曲の系列パターンに適した閾値をアドホックに設定することは難しく、特に音長は、精度の良い検索を行なうための閾値の設定がなされていなかった。また、曲 DB によっては、各曲の相対音高差・相対音長比の分布に偏りのある場合が考えられるため、本来は曲 DB の性質に応じた閾値を決定すべきであった。

3.2 閾値決定問題に対する解決法

本研究では、 DB 中の曲のすべての相対音高差・相対音長比の分布を考慮し、変換される値のカテゴリに含まれる相対値の合計度数が、カテゴリ間でできるだけ均等となるように、閾値を設定する手法を提案する。この閾値の決定手法は、相対音高差・相対音長比の分布から動的に定めているので、従来の静的な決定法に対して、**動的閾値決定法**と名付ける。具体的な動的閾値決定法については 4 で述べる。

図 3、4 の下部に本手法によって求められた閾値を示す。このように、DB 中の曲に含まれる相対音高差・相対音長比の系列を動的閾値によって変換すると、粗い相対値である各カテゴリ（U,E,D/L,E,S）の値の出現確率が等しくなる。検索時においても、歌声入力の相対音高差・相対音長比の系列を動的閾値によって変換することで、入力の 1 音符ごとに、正答の曲が効率良く均等に絞られていく。

また、後述する 3.4 にも関連するが、この閾値決定法によって、従来 3 段階にしか分類されなかった情報の粗さの精度を、自由に設定することが容易となる。例えば 5 段階に情報の粗さを設定する場合には、 5 つのカテゴリに分類される相対値の度数が均等に分類されるように、動的閾値を決定すれば良い。より細かい分類のカテゴリを用いることで、3 段階の粗さの精度では絞り込めなかったような曲 DB に対しても、正答の曲をより精度良く絞り込むことが可能となる。

3.3 正答の絞り込み問題

相対値を粗くカテゴリ分けして検索時のマッチングを行なうと、曲 DB 中の多くの曲で同様の系列パターンが出現する可能性が高くなり、正答の絞り込みは一般に難しくなる（図 5）。この問題を解決するために、3.2 の最後に述べたようなより細かい分類のカテゴリを用いることも考えられる。しかし、最初から細かい精度の相対値を用いると、利用者の歌声の変動や誤差による影響を吸収できず、逆に正答率が落ちてしまう。そこで、このトレードオフを勘案しながら、精度の高い検索を行なう手法が必要となる。

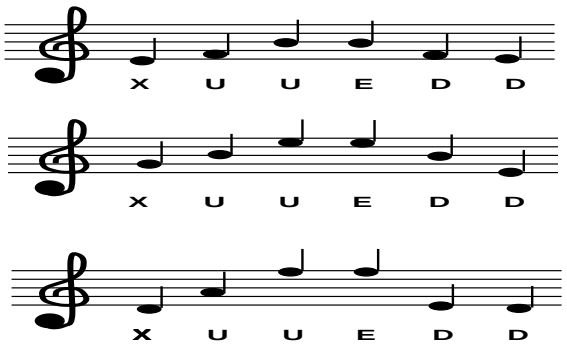

図 5　粗いカテゴリを用いたマッチングに見られる問題
Fig. 5　Problem in using approximate categories for matching.

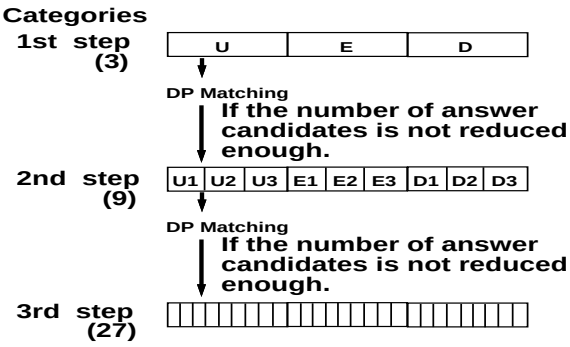

図 6　Coarse-to-Fine マッチング法の例
Fig. 6　Example for Coarse-to-Fine matching.

3.4 正答の絞り込み問題に対する解決法

本研究では、検索時に情報を粗い精度に変換ためのカテゴリ数を徐々に向上させながらマッチングを行なう手法を提案する。本手法では、まず、大まかなカテゴリ数（例えば 3 段階）の相対音高差・相対音長比を利用し、 DP マッチングを行なう。その結果、DB 中の曲で入力との距離が近いと解釈された曲の上位に着目する。ここで、その精度で絞り込みが十分かどうかの判定を行ない、絞り込めていなかった場合にのみ、より細かい精度（例えば 9 段階）の相対値を利用したマッチングを上位の曲に対して再度行なう。そして、カテゴリを細かくしながら絞り込みが十分にできるまでこれを繰り返す（図 6）。

たとえ粗い精度のマッチングで正答を絞り込めなくても、正答の曲は上位に含まれる性質があるため、このように上位の曲を対象に繰り返しより精密なマッチングを行うことで、正答の曲が適切に絞り込める。

このように精度を徐々に高めながらマッチングを行なう手法を、曲検索における **Coarse-to-Fine マッチング法** と呼ぶ。同様な概念の手法は画像のマッチング処理などでもよく用いられており、本研究ではそれを曲検索システムに応用した。具体的な Coarse-to-Fine マッチング法の手順については、**5** で述べる。

4. 動的閾値決定法

本節では、動的閾値決定法の手順と、得られた動的閾値を用いた検索キーの生成方法について述べる。

4.1 動的閾値決定法の手順

動的閾値 DT の決定手順を以下に示す。

（1） DB 内の全曲中の音高、音長の相対値で図 3、図 4のようなヒストグラムを作成する。

（2） ヒストグラムの総度数を Sum とし、カテゴリ数を n としたとき、1 つのカテゴリ内の合計度数の期待値を $EX = Sum/n$ とする。

（3） 各カテゴリの合計度数が均等に EX に近くなるようにヒストグラムを分割し、その各境界線を動的閾値 $DT(j)\quad (0 \leqq j \leqq n-2)$ とする。

このようにして決定した動的閾値を用いて、歌声の入力と DB 中の曲に出現するすべての音高・音長の粗い相対値を得る。以下では、その具体的な方法を述べる。

4.2 動的閾値を用いた検索キーの生成

歌声の相対音高差の系列 Q_p、相対音長比の系列 Q_s に対する動的閾値 $DT_p,\ DT_s$ を利用し、それぞれの粗い相対値情報の系列 $QR_p,\ QR_s$ を求める。

$DT_x(j)\ (0 \leqq j \leqq n-2,\ x = p, s)$ によって変換される n 段階の粗い相対値 $c_k\ (0 \leqq k < n,\ c_k < c_{k+1})$ の集合を C_x とする。長さ m の入力系列 Q_x を

$$Q_x = q_0\ q_1\ q_2\ \ldots\ q_{m-1}$$

で定義するとき、Q_x に出現する各音符の相対値 $q_i (0 \leqq i < m)$ を n 段階の粗い相対値情報 $r_i\ (\in C_x)$ に変換する際の条件は、以下のようになる。

$$r_i = \begin{cases} c_0 & (q_i < DT_x(0)) \\ c_k & (DT_x(k-1) \leqq q_i < DT_x(k)), \\ & 1 \leqq k < n-2 \\ c_{n-1} & (DT_x(n-2) \leqq q_i) \end{cases}$$

この変換により、Q_x は粗い相対値情報の系列

$$QR_x = r_0\ r_1\ r_2\ \ldots\ r_{m-1} \quad (r_i \in C_x)$$

に変換される。実際には、DP マッチングの前に、入力の旋律と DB 中の各曲の旋律に対して、共にこの変換を行う。

5. Coarse-to-Fine マッチング法

以下では、Coarse-to-Fine マッチング法の処理手順を述べる。

本システムでは、クライアントからサーバに送られる歌声入力の相対音高差・相対音長比の系列 Q_p, Q_s と DB 中のある曲の相対音高差・相対音長比の系列 S_p, S_s との間で DP マッチングを行なう前に、それぞれの系列を、動的閾値によって粗い相対値の系列に変換する。このとき、粗さのカテゴリ数を $n(0), n(1), \ldots, n(M-1)$ というように M 段階に変化させながら、粗い相対値の系列に変換して、意図する曲の候補を絞り込むことを考える。(ただし、$n(i) < n(i+1),\ i = 0, 1, \ldots, M-2$ とする)。

以下では、Q_x と $S_x (x = p,$ または $s)$ を、動的閾値 $DT_x(i, j)\ \ (0 \leqq j < n(i))$ によって、$n(i)$ 個の粗い精度の相対値で構成される系列 $QR_x(i)$ と $SR_x(i)$ に変換する手順を述べる。

（1） $i = 0$ とする。

（2） 動的閾値 $DT_x(i, j)$ によって Q_x と S_x を、それぞれ $n(i)$ 個の粗い相対値の系列 $QR_x(i)$、$SR_x(i)$ に変換する。

（3） DB 中のすべての曲の $SR_x(i)$ について、$QR_x(i)$ との系列間の距離 D_x を **2.2** の DP マッチングの計算により求め、D_x の小さい順に各曲にランク付けを行なう。

（4） D_x のランクの上位から $N(i)$ 番目までの曲に着目する（ただし、$N(i) > N(i+1)$ とする）。それらの D_x がすべて異なっている場合は絞り込みが完了しているとし、処理を終了する。

（5） 上位から $N(i)$ 番目までの曲の中で D_x が同じ値のものがある場合には、正答を絞り込めていない状態と判断し、$i+1$ を新たな i とみなして (2)〜(4) の処理を繰り返す。ただし、$i = M-1$ となった場合には、処理を終了する。

6. 実験と考察

本論文で提案した二つの手法の有効性を実験的に確

認するために、次の一連の評価実験をおこなった。

- 動的閾値決定法による動的閾値の情報量の評価 (6.2)
- 動的閾値決定法の性能評価 (6.3)
- Coarse-to-Fine マッチング法の性能評価 (6.4)
- Coarse-to-Fine マッチング法の歌声の誤差に対する耐久性の評価 (6.5)
- 両手法を共に用いた本システムの検索精度の総合評価 (6.6)
- WWW 上で運用する際の処理時間と転送データサイズの評価 (6.7)

以下、順番に実験結果と考察を述べる。

6.1 実験環境

サーバは JAVA (JDK1.1) で実装し、Just-In-Time compiler を用いて Sun Ultra Enterprise 3000 (CPU:UltraSPARC, 167MHz) 上で実行した。クライアントは WWW ブラウザとして Netscape Navigator を利用し、JAVA のアプレットとプラグインで実装したものを SGI 社のワークステーション Indigo2 Impact (CPU:R4400, 250MHz) 上で実行した。プラグインでは、8 kHz でサンプリングされた音声に対し、512 点での FFT を 1 フレーム 128 点 (16msec) でシフトしながら計算し、入力された歌声の音響信号を分析した。

6.2 動的閾値決定法による動的閾値の情報量の評価

動的閾値は、従来の閾値よりも理論的に情報量の大きい検索キーを生成するはずであるが、ここでは、それを実験的にも確認する評価実験を行なった。

6.2.1 実験

図 3,4 の動的閾値、従来の閾値のそれぞれを用いて、音高・音長情報を粗い情報（3 段階）に変換して比較する。曲 DB 中のすべての音符に対する各カテゴリ内の音符の出現確率 p_i から情報エントロピー $(-\Sigma p_i \log_2 p_i)$ を計算した結果を、表 1 に示す。

なお、表の一番右の欄の「上限値」とは、3 つのカテゴリに分類される情報が持つ最大の情報エントロピーの値 $(3 * (1/3) log_2 3 = 1.5850)$ を表す。

6.2.2 考察

従来の閾値による分類では音長情報は明らかに情報量が少ないが、動的閾値による分類では音高情報も音長情報もほとんど最大値に近付いている。この結果より、動的閾値が検索キーを生成するのに有効な手段であることが確かめられた。

表 1 閾値によって分割されるカテゴリの持つ情報エントロピー
Table 1 Information entropy of categories devided by thresholds.

	音高	音長	上限値
動的閾値	1.5850(bit)	1.5846(bit)	1.5850(bit)
従来の閾値	1.5836(bit)	1.2025(bit)	1.5850(bit)

6.3 動的閾値決定法の性能評価

実際の検索での、動的閾値決定法による正答率の向上を確認する評価実験を行なった。

6.3.1 実験

従来の閾値と動的閾値による検索精度の比較を行なった。様々なジャンル（日本・西洋のポップス、童謡、民謡、演歌など）からなる 200 曲の DB （総音符数 16718 個）に対し、12 人（男 8:女 4）の被験者の音声によって、音高のみ、音長のみ、および音高・音長の組合せによる検索を計 112 回行なった (音高・音長の組合せ検索では、**2.2** で述べた DP マッチングのマッチング行列 m の各要素の計算 (Step A,C) において、入力系列と DB 中の各曲の系列の各音符間の距離のペナルティ d を、各音符の音高間の距離 d_p と音長間の距離 d_s との和 $d = d_p + d_s$ で定義し、計算を行った。以下の実験でも同様)。閾値の条件以外は同条件で行ない、粗い旋律情報のカテゴリ数は音高、音長ともに 3 段階とし、Coarse-to-Fine マッチング法は用いていない。従来の閾値は、音高に関しては半音以内の音高差を「同じ」とみなし、音長に関しては 1/2 より大きく 2 倍未満の音長比を同音長とした [2]。入力を 8 秒間以内に限定したところ、入力音符数は平均で 18.5 となった。評価結果を表 2、3 に示す。「正答率」 は、意図する曲が厳密に 1 位に絞られていた場合のみを表し、他の複数の曲とともに入力旋律からの距離が等しく、1 位となった場合は含まない。また、「3 位以内」は意図する曲が上位の 3 曲以内に入っていた場合を表す。

6.3.2 考察

図 3 からも予想できるように、音高での検索はそれほど違いが生じなかったが、音長は明らかに正答率が向上したことが確認できた。音高・音長を組み合わせた検索も正答率が向上していることから、本手法が有効であることが確かめられた。

6.4 Coarse-to-Fine マッチング法の性能評価

Coarse-to-Fine マッチング法で $3 \rightarrow 9 \rightarrow 27$ 段階に音高・音長の精度を変化させた検索を行ない、その

表 2　従来の閾値を利用した正答率
Table 2　Matching accuracy with previous threshold-s.

	音高検索	音長検索	組合せ検索
正答率	47.3%	13.4%	90.2%
3 位以内	65.2%	25.0%	95.5%

200 曲に対して 112 回検索

表 3　動的閾値を利用した正答率
Table 3　Matching accuracy with dynamic threshold-s.

	音高検索	音長検索	組合せ検索
正答率	49.1%	35.1%	97.3%
3 位以内	67.0%	50.1%	99.1%

200 曲に対して 112 回検索

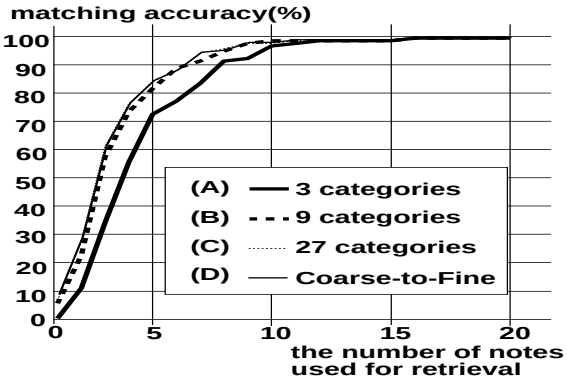

図 7　Coarse-to-Fine と各段階の粗さでのマッチング精度
Fig. 7　Matching accuracy of Coarse-to-Fine matching and its each phase.

検索の正答率を、各段階で用いた精度でのマッチングの結果（Coarse-to-Fine マッチング法を用いないもの）と比較し、Coarse-to-Fine マッチング法の性能について考察した。

6.4.1　実　　験

以下の 4 つの条件 **A** ～ **D** において検索に用いる音符数を 1 ～20 個の間で変化させ、正答率を比較した。Coarse-to-Fine マッチング法は **D** のみに用いている。Coarse-to-Fine マッチング法においては、マッチングに用いるカテゴリ数を 3 → 9 → 27 と向上させて行なった。このとき、カテゴリ数が 3 の検索の結果、入力との距離が小さいものから上位 30 曲のみに着目し、絞り込みが行なえていないときに、カテゴリ数が 9 の検索を行なった。さらに、絞り込みが行なえていないときに入力との距離が小さいものから上位 10 曲のみに着目し、カテゴリ数が 27 の検索を行なった。図 7に結果を示す。ここで、正答率は厳密に正答が 1 曲に絞り込まれているもので評価した。

- **条件 A**

動的閾値によって 3 段階の粗い相対値情報に変換した音高・音長を用いた検索。

- **条件 B**

動的閾値によって 9 段階の粗い相対値情報に変換した音高・音長を用いた検索。

- **条件 C**

動的閾値によって 27 段階の粗い相対値情報に変換した音高・音長を用いた検索。

- **条件 D**

Coarse-to-Fine を用い、音高・音長を動的閾値によって 3 → 9 → 27 段階に変化させながら検索。

6.4.2　考　　察

Coarse-to-Fine マッチング法での検索は、27 段階の粗い旋律情報を利用した検索と正答率がほとんど一致した。 これは、Coarse-to-Fine マッチング法の精度が最終的に 27 段階のものと同様になるということと、本実験の被験者の入力にはそれほど大きな誤差が含まれていなかったことが原因だと考えられる。Coarse-to-Fine マッチング法は誤差が含まれたときに効果的であるため、この実験より、ほとんど誤差の含まれない入力に対する評価では効果があまり表れないことが確認できた。

6.5　Coarse-to-Fine マッチング法による歌声の誤差に対する耐久性の評価

Coarse-to-Fine マッチング法が歌声の誤差に対して耐久性を持つことの評価を行なうため、人為的に誤差を投入した歌声の入力に対する正答率を、誤差を投入しない入力に対する正答率と比較する実験を行なった。

6.5.1　実　　験

27 段階の精度での検索 と Coarse-to-Fine マッチング法を用いた検索の 2 つの場合について、実験に用いた被験者の歌声の旋律に人為的な誤差を入れ、正答率の変化を観察した。Coarse-to-Fine マッチング法においては、**6.4** の D と同じ条件で行なった。誤差は、1 つの入力歌声の旋律の中から 2 つの音符をランダムに選択し、相対音高差に対しては、全音の幅ずらす処理（すなわち、相対値に +200 または-200 を施すこと）を行ない、相対音長比に対しては、相対値を 1/4 倍または 4 倍にする処理を行なった。誤差を含まない場合の条件 C,D それぞれの正答率を表 4 に、誤差を含む場合の正答率を表 5 に示す。なお、正答率は厳密

に正答が 1 曲に絞り込まれているもので評価した。

表 4　人為的な誤差を含まない場合の正答率
Table 4　Matching accuracy with no artificial errors.

	音高検索	音長検索	組合せ検索
27 段階	87.5%	82.1%	97.3%
Coarse-to-Fine	83.0%	79.5%	98.2%

200 曲に対して 112 回検索

表 5　人為的な誤差を含む場合の正答率
Table 5　Matching accuracy with artificial errors.

	音高検索	音長検索	組合せ検索
27 段階	75.9%	34.8%	90.1%
Coarse-to-Fine	80.3%	50.0%	94.6%

200 曲に対して 112 回検索

6.5.2 考　　察

人為的な誤差を含まない場合の音高検索、音長検索において、Coarse-to-Fine マッチング法を利用した検索は、27 段階の精度の検索よりも正答率が低くなった。これは、Coarse-to-Fine マッチング法が、カテゴリ数が粗い段階 (カテゴリ数 3 または 9 の段階) でマッチング処理を終了した際に、意図する曲が、マッチング結果の上位の曲中に含まれてはいながらも、マッチング精度が粗いために、正答にはならなかった場合があることが原因であった。一方、組合せ検索では Coarse-to-Fine マッチング法を用いた方が検索率がより向上していた。これは、Coarse-to-Fine マッチング法では、音高検索と音長検索のそれぞれにおいて、意図する曲がマッチングの結果の上位に含まれることが多かったため、音高と音長を組み合わせることで、効果的に曲を絞り込めていたからだと考えられる。これに対し、27 段階の精度の検索では、音高検索、音長検索のそれぞれにおいて、人為的でない誤差の影響があった場合、DP マッチングのペナルティが大きくなり、意図する曲が正答にならないばかりか、上位の曲中にも含まれないことが多かった。このため、音高と音長のいずれか一方にでも誤差の影響がある場合には、組合せ検索においても、その影響を受けており、検索精度が上がらなかったと考えられる。

人為的な誤差を含む検索では、いずれの検索においても Coarse-to-Fine マッチング法を用いた方が正答率が高かった。これは、Coarse-to-Fine マッチング法では、粗いマッチングの段階で正答の候補を絞り込むことで、精密なマッチングにおいて、誤差の影響を受けにくいためである。

6.6 両手法を共に用いた本システムの検索精度の総合評価

本研究で提案した動的閾値決定法と Coarse-to-Fine マッチング法を組み合わせた場合の、システムの検索精度の総合評価を行なった。

6.6.1 実　　験

本評価には、**6.4** の 条件 D における検索結果を用いた。音高のみ、音長のみ、音高・音長を組合せた検索について、それぞれ正答率を表 6 にまとめた。

表 6　動的閾値決定法と Coarse-to-Fine マッチング法を用いた検索における正答率
Table 6　Matching accuracy with dynamic threshold determination and Coarse-to-Fine matching.

	音高検索	音長検索	組合せ検索
正答率	83.0%	79.5%	98.2%
3 位以内	88.4%	88.4%	100.0%

200 曲に対して 112 回検索

6.6.2 考　　察

動的閾値決定法と Coarse-to-Fine マッチング法を組み合わせた検索では、歌声入力の誤差を考慮した上で、精度の良い情報を用いた検索を行なうことが可能となる。このため、表 6 の結果では音高検索、音長検索において、検索に用いるカテゴリ数を 3 とした表 3 の正答率よりも 30 〜40 % の向上が見られた。一方、音高・音長の組み合せ検索においては DB サイズの 200 曲がそれほど大きくない値であることが原因で、大きな向上は見られなかったが、100 % に近い正答率が得られた。

以上の結果より、動的閾値決定法と Coarse-to-Fine マッチング法を組み合わせた検索が、高い精度の検索を実現する上で非常に有効な手法であり、WWW 上の様々な利用者に対して歌声検索の処理を行なう上で、十分に活用できるものであることが確認できた。

6.7 WWW 上で運用する際の処理時間と転送データサイズの評価

WWW 上で検索システムの運用を行なう上で重要となる、検索に関する処理時間の評価を行なった。また、ネットワークにかかる負荷を評価するためにクライアントからサーバに転送されるデータのサイズを計測した。

6.7.1 実　　験

サーバの検索処理の評価のために動的閾値の生成時間、マッチング処理に要する時間の 2 つを計測した。

また、サーバ・クライアント間で転送されるデータサイズとして、**6.3** で用いた入力の旋律情報のデータサイズの計測を行なった。

サーバの検索処理における各処理時間は以下のようになった。

- **動的閾値の生成時間 (6.41 sec)**

サーバの実装環境で 200 曲（総音符数 16718 個）の DB に対する動的閾値の生成を 100 回行なった平均値である。閾値は音符の相対音高差・相対音長比をカテゴリ数 27 の粗い相対値に分割するものを生成した。

- **マッチング処理時間 (0.96 sec)**

サーバの実装環境で 200 曲（総音符数 16718 個）の DB に対する 112 回 (総音符数 3708 個) の検索を行なった平均値である。動的閾値と Coarse-to-Fine (3,9,27 段階) マッチング法を用いた検索を行なった結果である。

また、クライアントからサーバに転送されるデータのサイズの平均値・最大値は表 7のようになった（入力を 8 秒間以内に限定し、被験者 12 人が合計 112 回の検索を行なったときの平均値・最大値である)。

表 7　クライアントからサーバに転送されるデータサイズ
Table 7　Transmission data size from the client to the server.

平均値	最大値
274.5 (bytes)	420 (bytes)

6.7.2 考　　察

動的閾値の生成は、DB を更新した場合やサーバを起動した場合にのみに行えば、各閾値の情報を保持しておくことで、クライアントからの検索要求時に、毎回、値を計算しなくてもよい。この方式を採用することで、実験結果から得られた、動的閾値生成に要する時間は、検索システムの運用上、大きな問題にはならないことが確認できた。また、マッチング処理に関しても、実験で用いた 200 曲の DB に対する処理時間は、現在のインターネット利用者が、WWW ページを閲覧する際に、しばしば 1 秒以上の表示時間を要していることと比較すると、利用者が結果を待つことに、十分に耐え得る時間内であることが確認できた。

しかし、これら 2 つの処理時間は、DB 中の音符の総数に比例した計算時間を伴うため、今後 1 万曲規模の DB に適用するためには、処理を高速化するためのアルゴリズムの改良が必要となる。

一方、クライアントからサーバに対して転送されるデータに関しては、非常に小さく、入力時間を、仮に数倍長くしてもネットワークにかかる負荷や転送時間における問題は発生しないことが確認できた。

7. まとめと今後の課題

本論文では WWW 上で複数の利用者が活用できる歌声による曲検索システムを提案した。本研究では、検索キー生成のための最適な閾値を得るために動的閾値決定法を提案し、システムのマッチング精度を向上させることができた。特に、音長のみを検索キーとして用いた検索においては、200 曲の DB に対して、正答率を 20 % 以上、向上することができた。さらに、Coarse-to-Fine マッチング法を導入することで、正答の曲の候補を効果的に絞り込むシステムを実現し、音高・音長をともに用いて 100 % に近い正答率を実現することができた。

現在の実装では、予め用意された曲 DB に対する検索のみ可能であるが、WWW 上には MIDI などの様々な曲データが公開されている。そこで今後は、それらのデータを収集する音楽版 WWW Robot を開発することで、世界中の WWW 上の 曲 DB に対して検索を行うシステムを構築する予定である。

文　　献

[1] 貝塚 智憲, 後藤 真孝, 村岡 洋一: 歌声の旋律情報と歌詞情報をキーとした曲検索システム, 54 回情報処理学会全国大会, 1997.

[2] 蔭山哲也 , 高島洋典 : ハミング歌唱を手掛かりとするメロディ検索, 電子情報通信学会論文誌 D-II Vol.J77-D-II No.8, pp.1543-1551, 1994.

[3] T.Kageyama, K.Mochizuki, Y.Takashima : *Melody Retrieval with Humming*, ICMC Proc., pp.349-351, 1993.

[4] 蔭山哲也 : 音高・音長情報を利用したメロディ検索, 45 回情報処理学会全国大会, 1-357, 1992.

[5] 蔭山哲也 , 島津秀雄 , 高島洋典 : メロディ検索 - ハミングで音楽 *DB* を検索する , 43 回情報処理学会全国大会, 4-149, 1991.

[6] Asif Ghias, Jonathan Logan : *Query By Humming – Musical Information Retrieval in an Audio Database*, ACM Multimedia 95, Electronic Proc., 1995.

(平成年月日受付，月日再受付)

園田　智也

1998 早大・理工・情報卒．現在同大学大学院修士前期課程在学中．音楽検索，WWW サーチエンジンなど情報検索に関する研究に従事．情報処理学会第 55 回全国大会奨励賞受賞．

後藤　真孝　（正員）

1993 年早稲田大学理工学部電子通信学科卒業．1998 年同大大学院博士後期課程修了．同年，電子技術総合研究所に入所し，現在に至る．博士（工学）．音楽情報処理，マルチモーダルインタラクションなどに興味をもつ．1992 年 jus 設立 10 周年記念 UNIX 国際シンポジウム論文賞受賞．1993 年 NICOGRAPH'93 CG 教育シンポジウム最優秀賞受賞．1997 年情報処理学会山下記念研究賞受賞．情報処理学会，日本音楽知覚認知学会，ICMA 各会員．

村岡　洋一　（正員）

1965 年早稲田大学理工学部電気通信学科卒業．1971 年イリノイ大学電子計算機学科博士課程修了．Ph.D. この間，Illiac-IV プロジェクトで並列処理ソフトウェアの研究に従事．同学科助手ののち，日本電信電話公社（現 NTT）電気通信研究所に入所．1985 年より早稲田大学理工学部教授．現在同大学メディアネットワークセンター所長．並列処理，マンマシンインタフェースなどに興味をもつ．「コンピュータアーキテクチャ」（近代科学社）など著書多数．