

2 値多重音響特徴ベクトルを用いた類似音楽探索とその高速化

永野 秀尚[†] 柏野 邦夫[†] 村瀬 洋^{†*}

Similar Music Retrieval Using Polyphonic Binary Feature Vectors
and Its Acceleration

Hidehisa NAGANO[†], Kunio KASHINO[†], and Hiroshi MURASE^{†*}

あらまし 本論文では、多重奏音響信号をクエリーとして多重奏音響信号から類似する音楽を探索する類似音楽探索法とその高速化手法を提案する。本類似音楽探索においては多重奏音響の類似性に基づいた探索を高速に行う必要がある。そこで、まず、多重奏において同時に発生する複数の音の有無に着目した 2 値多重音響特徴ベクトルを提案する。本特徴は多重奏音響信号の類似性に基づいた高速探索に適している。そして更に、本特徴を用いた探索における高速化手法を導入する。本手法は類似度行列のスパース化により探索空間を制限し、より高速な探索を行う。216 曲の実験用音楽データベースを用いた探索実験では、例えば、楽曲単位の探索において、クエリーの長さが平均 19 秒のとき、スペクトル特徴を用いた場合に 62.5%であった精度が、提案の 2 値多重音響特徴ベクトルを用いることにより 89.3%にまで向上した。また、本高速化手法においては、精度をほとんど損なわずに、DP 照合を用いたずらし照合法に比べ約 1.6 倍から 18 倍の高速化を達成した。

キーワード 音楽検索, 多重奏, 音響探索, 2 値多重音響特徴ベクトル

1. ま え が き

音や映像のメディア情報の増加、多様化により様々なメディア情報の探索技術が必要とされている。我々は特に、メディア探索、すなわち、長時間の音や映像の信号(蓄積信号)またはそのデータベース(DB)と、探したい音や映像の信号(参照信号)がクエリーとして与えられたとき、蓄積信号中の参照信号に類似する区間を探し出す探索技術について研究を進めてきた[1]~[4]。そして、本論文では音楽演奏の音響信号の探索において、多重奏音響信号を参照信号とし、同じく多重奏音響信号の中から、参照信号に類似する区間を見つけ出す手法とその高速化手法を提案する。このような多重奏音響信号による類似音楽探索は、例えば、通常多重奏音響信号である音楽 CD の断片などを参照信号とし、それに類似する音楽を音楽音響信号の DB から検索するために必要な技術である^(注1)。更に、

我々は参照信号と同一の音楽演奏の区間だけではなく、類似する音楽、例えば、再演奏されたもの、他の楽器を用いて演奏されたもの、テンポの異なる演奏、若しくは移調されたものなど、同一の音響信号ではないが同じ音楽である類似音楽も探索することを目的とする。

メディア探索の重要な応用例である音楽検索についてはこれまで多くの研究がなされてきたが、それらは、その目的により二つに大別できる。一つは音楽 CD などと信号レベルでほぼ同一の音楽の検索をねらう一致探索であり、この場合、音響信号から得られるスペクトル特徴及びそのヒストグラムを用いる手法[1],[3],[4]や音響特徴間の類似度をベクトル量子化により高速に計算する手法[5]などが提案されている。一方、先に述べたように、信号レベルでは必ずしも一致していないが、演奏者が異なる同一の曲や、主旋律が同じ曲や、アレンジの異なる曲などのように、何らかの意味で類似しているものの検索をねらう類似探索がある[6]~[11]。このような類似音楽探索においては、音響特徴が蓄積信号と参照信号で大きく異なる

[†] 日本電信電話株式会社 NTT コミュニケーション科学基礎研究所, 厚木市

NTT Communication Science Laboratories, NTT Corporation, 3-1 Morinosato Wakamiya, Atsugi-shi, 243-0198 Japan

* 現在, 名古屋大学大学院情報科学研究科

(注1): 本論文では DB や蓄積信号中から参照信号に類似する区間がどこにあるか探し出す技術を探索といい、DB などから類似する音楽を検索するタスクについて検索という。

るためや、音響信号以外に楽譜などで音楽が表現されるため、音響特徴以外の特徴を用いた探索が必要となる。特に Ghias らは単旋律の音高系列を“S(同じ)”, “U(上がる)”, “D(下がる)”の相対音高の文字列で表す melodic contour を用いた単旋律の照合手法を提案し、ハミングをクエリーとし、類似する単旋律を検索する類似音楽探索においてその有効性を示した [7]。melodic contour は旋律の表現としては楽譜などに比べ単純であるが、単旋律の識別、照合には有効であり、その他の研究においても単旋律の探索における特徴として用いられている。そして、これまでの類似音楽探索の多くは、ハミングなどで入力された単旋律の音高系列をクエリーとし、同じく単旋律の音高系列からなる DB からクエリーに類似するものを、単旋律同士の照合により探索するものであった。ここで、多くの手法において、DB の音高系列は、MIDI や楽譜情報、若しくはクエリーと同じくハミングによる主旋律の入力から得られる、既に主旋律などが抽出された単旋律の音高系列であった。音響信号からなる DB を対象とした類似音楽探索については、橋口らが、ハミングによる単旋律音高系列をクエリーとし、単旋律对多重奏の照合を前提とした手法を提案している [11]。しかし、本研究で対象とする多重奏音響信号をクエリーとした多重奏音響信号からの類似音楽探索の場合、単旋律のクエリーや DB を前提とした照合法を用いるには、多重奏からの主旋律抽出が必要となることや、若しくは多重奏と多重奏の間での主旋律の照合など困難な問題が多い [12], [13]。このため、これまで、多重奏音響信号同士の比較に基づく類似音楽探索の手法は提案されていなかった。

このように、本論文において対象とする多重奏音響信号における類似音楽探索においては、同一の曲であっても、演奏の変化、楽器の変更、演奏速度の違いなどにより、音響信号が激しく変化するため、スペクトル特徴のような音響信号を用いた探索はその探索精度の低下が予測される。また、従来の単旋律の照合を基本とした手法では、音響信号からの旋律の抽出や多重奏と多重奏の間での主旋律の照合及びそのための探索時間の増加などの困難な問題点があった。そこで、本論文ではこのような多重奏音楽の音響信号の照合のための 2 値多重音響特徴ベクトルを提案し、この特徴を用いた類似音楽探索法を提案する。この提案は、Ghias らの melodic contour と同様に、音楽演奏の時系列を単純な符号の 1 次元系列で表現し、この符号系列に対

し、文字列照合を行うことでロバストかつ高速な類似音楽の探索を狙うものである。2 値多重音響特徴ベクトルは、各成分を 2 値とする特徴ベクトルであり、この各成分は各々対応する音高の音の有無に大まかに対応する。この特徴は Ghias らの melodic contour が旋律の正確な記述を目的としたものではないと同様に、同時に発音している音の正確な記述を目的とするものではないことに注意したい。これを 2.2 で述べるように、多重奏音楽の音響信号から、同時に発生する複数の音に基づき、順次抽出し 2 値多重音響特徴ベクトルの時系列を得る。そして、蓄積信号と参照信号から得られた 2 値多重音響特徴ベクトルの系列について、2 値多重音響特徴ベクトルの類似度に基づいた文字列照合を行うことにより、類似音楽探索を行う。この 2 値多重音響特徴ベクトルにより、多重奏音響信号同士の照合が可能になり、本論文で目的とする多重奏音響信号間の類似音楽探索が可能になる。更に、本論文ではこの 2 値多重音響特徴ベクトルの類似度に基づいた文字列照合の高速化手法を提案し、その探索の高速化を行う。

以下、2. では 2 値多重音響特徴ベクトルを用いた類似音楽探索手法とその高速化手法について述べ、3. で提案手法の探索精度及び探索速度に関する実験について述べる。そして、4. でまとめる。

2. 類似音楽探索法

2.1 探索手続きの概要

図 1 に探索手続きの概要を示す。ここで、蓄積信号 (stored signal) と参照信号 (reference signal) は多重奏音楽の音響信号であり、探索においては、蓄積信号中の参照信号に類似するすべての区間を検出することを目的とする。まず、探索の準備として、各音響信号について周波数分析を行い特徴ベクトルを時系列順に抽出する。そして、抽出された特徴ベクトル

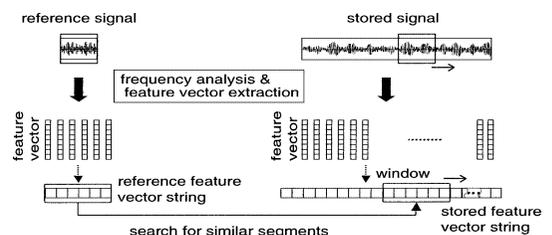


図 1 探索手続きの概要

Fig. 1 Overview of searching.

ルを順次符号化し符号系列を得る．ここで蓄積信号と参照信号から得られる符号系列を、各々、蓄積符号系列 (stored feature vector string)、参照符号系列 (reference feature vector string) と呼ぶ．そして、探索においては、蓄積符号系列中の参照符号系列に類似する長さ w のすべての部分系列を探索し、検出する．ここで w は探索窓 (window) の幅として与えられるものとする．そして、参照符号系列と長さ w の部分系列の類似度は符号間の類似度に基づいた DP 照合により求める．これは、演奏速度の違いなどによる時間軸方向の信号伸縮に対応するためである．なお、この DP 照合により得られる類似度を系列間類似度と呼び、照合を行う符号系列の長さ、すなわち参照符号系列の長さ w により正規化されているものとする．以上、本探索で行う処理についてその概要を述べたが、本探索は、形式的には、参照符号系列との系列間類似度が探索しきい値 s_t より大きい蓄積符号系列中の長さ w の部分系列をすべて検出することとする．

2.2 2 値多重音響特徴ベクトル

次に、特徴ベクトルとして用いる 2 値多重音響特徴ベクトルの抽出過程について述べるが、そのために SNAP (Simultaneous-Note-set Alteration Point; 同時単音集合変化時刻) の概念を提案し導入する．

多重奏音楽は、複数の単音の組合せから構成されているが、どれか一つの単音が立ち上がる時刻、あるいは同時に複数の単音が立ち上がる場合にはその時刻を SNAP と呼ぶ．すなわち、SNAP は多重奏音楽において、単音の立上りにより、同時に鳴っている単音の組合せの変化が起きた時刻を示す^(注2)．そして、隣り合う SNAP と SNAP の間の一定の組合せの音が鳴っている区間を snapshot と呼ぶ．そして、この snapshot ごとに一つの 2 値多重音響特徴ベクトルを抽出し、一つの 2 値多重音響特徴ベクトルを一つの符号に符号化する．この snapshot 単位の符号化には、分析窓に重なりをもたせながら抽出した多数の短時間スペクトル特徴の系列を順に一つひとつ符号化する場合に比べ、一般に、符号系列長を短くできるというメリットがある．また、snapshot 単位の符号化では、演奏速度の違いなどによる音の長さの伸縮を吸収することも期待できる．

SNAP に似た概念としては拍があり、拍位置の抽出手法もいくつか提案されている [14], [15]．このような拍位置を抽出し、拍単位で符号化することも考えられる．しかし、今回の音楽の演奏の分割の目的は、音楽

的な拍位置を厳密に抽出することではなく、探索のために音楽を符号系列として表現する際に 1 符号が対応する区間を得ることである．ここで、各符号は複数の音の有無を大まかに表現するものである．そのため、音の組合せの変化に着目した snapshot 単位で符号化することとした．

以下に、SNAP の概念を用いた 2 値多重音響特徴ベクトルの抽出過程について、順に述べる．

2.2.1 音響特徴抽出

まず、帯域通過フィルタを用い音響波形信号から各周波数におけるパワースペクトルを得る．帯域通過フィルタは 75 Hz から 9600 Hz まで 1 オクターブごとに 48 ずつ 7 オクターブ分、計 336 個のフィルタを周波数の対数軸上で等間隔に配置して用いた．そして、各フィルタで、44 ms の時間区間の分析を、11 ms ごとに行い、各時間、各周波数におけるパワーを抽出した．

2.2.2 SNAP 抽出

上記で得た時間-周波数-パワー空間のスペクトログラムを用い、SNAP を抽出する．本論文では、文献 [15] の拍位置抽出の考え方に基づき、周波数成分の立上りとなるスペクトログラム上の点が集中する時刻を SNAP として求めることとした．また、複数の単音が同時に発生する場合にも、厳密には発生時刻にばらつきがある．そこで、このようなばらつきは基本拍間隔 [15] に基づき吸収し、複数の単音が同時刻に発生したとする．そして、この時刻を一つの SNAP とすることとした．ただし、本論文では、基本拍間隔も周波数成分の立上りとなる点に着目して計算する．以下、本論文の SNAP 抽出手続きにおいて用いる用語を述べた後、その抽出手続きについて述べる．

本論文で周波数成分とは、スペクトログラムを時間-周波数-パワー空間における曲面としてみなしたとき、その曲面上のパワーが一定のしきい値を超える周波数軸方向のピークについて、ピーク近傍における曲面の尾根方向の接線ベクトルを接続したものとする．そして、周波数成分の立上りとなっている点としては以下の抽出開始端点と端点候補点を抽出し用いることとした．抽出開始端点は、周波数成分上の点で、スペクトログラム上で周波数軸方向の極大値であり、その周

(注2): ただし、ある単音の組合せの演奏のあとに続けて同じ単音の組合せが再度演奏された場合、あとの音の立上りにより二つの単音の組が分離している場合は、単音の組合せが変化せずとも、後者の演奏の立上りの時刻を SNAP とみなす．例えば、ピアノで {C, E, G} の和音を弾き、そのあと続けて再度同じ和音を弾いた場合には後者の和音の立上りは SNAP とみなす．

波数成分上でパワーに関する抽出しきい値を時間軸方向に見て初めて上回る点である．端点候補点は周波数成分の尾根上の点で，尾根上でくぼみとなった後，尾根を登り始める点である．以降，抽出開始端点と端点候補点をまとめて SNAP 形成候補点と呼ぶ．

次に，本論文で用いた SNAP の抽出手続きについて述べる．

(1) 最大パワーがしきい値以上で，時間方向の長さもしきい値以上の各周波数成分から SNAP 形成候補点を抽出する．

(2) 得られた各 SNAP 形成候補点 i について， i の時刻が SNAP になる確率 (SNAP 確率) P_i を次式のようにして計算する [15]．

$$P_i = 1 - \prod_{j \in \Gamma_i} \left(1 - p_j \exp \left(-\frac{(t_i - t_j)^2}{\Delta T_i^2} \right) \right). \quad (1)$$

ここで， p_j は SNAP 形成候補点 j の端点確率であり， i が周波数成分の立上りであることのゆい度を表す．端点確率は，各 SNAP 形成候補点について，その点周辺のスペクトログラムの形状等から求めることも考えられるが，本論文ではすべての SNAP 形成候補点で同じ定数を与えた．また， t_i は i の時刻であり， Γ_i は i からの時刻の差がしきい値 (本論文では 5 秒とした) 以下の SNAP 形成候補点の集合である．また， ΔT_i は i における基本拍間隔の $\frac{1}{8}$ の値である．そして， i からの時刻の差がしきい値 (上記しきい値と同様に 5 秒とした) 以下の SNAP 形成候補点の集合を Ψ_i とし， Ψ_i における二つの SNAP 形成候補点の時刻の差のうち最頻のものを基本拍間隔とする．本論文では， Ψ_i における SNAP 形成候補点の時刻の差を 0.2 秒ごとに分類し，最頻となる時刻の差を基本拍間隔とした．

(3) 各 SNAP 形成候補点について，その SNAP 形成候補点及びその SNAP 形成候補点とほぼ同時刻 (本論文では時刻の差が 0.5 秒以内とした) の SNAP 形成候補点の SNAP 確率の総和 (SNAP 近傍総確率) を計算する．

(4) SNAP 近傍総確率とその近傍 (時刻が近い SNAP 形成候補点，本論文では時刻の差が 0.5 秒以内とした) で最大の点で，その近傍内の SNAP 形成候補点の SNAP 確率の最大値がしきい値以上の点の時刻を SNAP 候補時刻とし，その集合を T とする．

(5) T において， T の各 SNAP 候補時刻での基本拍間隔を求める．本論文では， T の SNAP 候補時刻 t_i の基本拍間隔は， T の SNAP 候補時刻のうち t_i

との時刻の差がしきい値 (本論文では 5 秒とした) 以下の SNAP 候補時刻からなる T の部分集合を Φ_{t_i} とし， Φ_{t_i} における SNAP 候補時刻の差のうち最頻のものを基本拍間隔とする．本論文では， Φ_{t_i} における SNAP 候補時刻の差を 0.2 秒ごとに分類し，最頻となる時刻の差を基本拍間隔とした．

(6) T から SNAP 近傍総確率が最大の時刻 t を選ぶ．

(7) T の SNAP 候補時刻のうち， t との時間差が t の基本拍間隔の $\frac{1}{8}$ 以内の SNAP 候補時刻は SNAP でないとし， T から取り除く．

(8) T のうち，まだ選ばれていない時刻を，SNAP 近傍総確率の大きい順に t として選び (7) を繰り返す．これを T の選ばれていない時刻がなくなるまで繰り返す．

上記 (1)~(8) の操作で得られる T の時刻を SNAP とする．このようにして得られた各 SNAP を用いて，時間-周波数-パワー空間のスペクトルを “snapshot”-周波数-パワーのスペクトルにする．ここでは，各 snapshot で，各周波数における snapshot 内でのパワーの最大値をその周波数でのパワーとした．

2.2.3 倍音除去

次に，上記で得られた各 snapshot において，倍音除去を行い，各音 (半音単位の音高で表現) の有無を調べる．この倍音除去は，同時に発音している音の基本周波数に大まかに対応した表現を得るために行う．

まず，ある周波数について，対数軸上で最も周波数の近い半音単位の音高をその周波数の対応する基本音と呼ぶこととする．そして，各 snapshot で周波数軸方向のパワーのローカルピークについて，その対応する基本音を求め，その音が存在したとする．このようにして，この snapshot でローカルピークであった音を a_1, a_2, \dots, a_N とし，そして，それらに対応する基本音を A_1, A_2, \dots, A_J とする．なお，ここでは，異なるピーク a_i, a_j が同じ基本音 A_k に対応し得る．そして， $h_{i,1}, h_{i,2}, \dots, h_{i,\delta}$ を a_i の 1 倍音から δ 倍音とし，このうち，この snapshot 内で存在する a_i の倍音の個数を c_i とする．そして， $c_i \geq \varepsilon$ のとき， a_i の対応する基本音 A_j が存在したとする．ここで ε は倍音除去のしきい値である．すなわち， a_i の δ 倍音のうち ε 個以上が存在すれば，この snapshot 内で a_i の対応する基本音が存在したとする．本論文では，

$$c_i = \sum_{k=1}^{\delta} e(h_{i,k}) \quad (2)$$

とし, $e(h_{i,k})$ は, $h_{i,k}$ の対応する基本音と同じ基本音に対応する音が存在した^(注3)場合に 1, そうでないとき 0 とした.

2.2.4 2 値多重音響特徴ベクトルへの変換

上記で得られた基本音の有無に着目し, 各 snapshot について 12 ビットの 2 値多重音響特徴ベクトルを構成する.

特徴ベクトルの各ビットは下位から順に A, A#, B, ..., G# の各音に対応しており, この snapshot 内で同じ音名となる基本音が存在したとき, 対応するビットを 1 に, そうでないとき 0 にする. 例えば, A (220 Hz), E (330 Hz), A (440 Hz), C (523 Hz), A (880 Hz) が snapshot 内に存在した場合, この snapshot の特徴ベクトルは 2 進表現で “000010001001” とする. 2 値多重音響特徴ベクトルは, 多重音響信号において同時に発生する音の有無に大まかに対応していることになる.

2.3 2 値多重音響特徴ベクトルの類似度

2 値多重音響特徴ベクトル $\mathbf{x} = (x_1, x_2, \dots, x_{12})$ と $\mathbf{y} = (y_1, y_2, \dots, y_{12})$ の間の類似度は

$$\frac{\sum_{k=1}^{12} u_k}{\sum_{j=1}^{12} (x_j + y_j)} \quad (3)$$

と定義する. ここで u_k は, $x_k = y_k = 1$ のとき $u_k = 2$ で, そうでないとき $u_k = 0$ と定義される. なお, $\mathbf{x} = \mathbf{y} = \mathbf{0}$ の場合, \mathbf{x} と \mathbf{y} の類似度は 1 とする. これは無音の snapshot 同士は同じ和音として取り扱うことを意味している. なお, 類似度としてはハミング距離を 0 から 1 の範囲に正規化したもの等も考えられるが, 予備実験の結果, 比較的良好な探索結果が得られた式 (3) を用いた.

2.4 2 値多重音響特徴ベクトルの符号化

2 値多重音響特徴ベクトルの符号化は 2.2.4 で述べた 2 値多重音響特徴ベクトルを単に 12 ビットの 2 進数とみなすことで行う. この符号化により, 2 値多重音響特徴ベクトルを簡潔に符号化できると同時に, 探索時に参照符号系列の符号をシフトしてから探索することにより, 移調された音楽の探索も可能となる.

なお, 符号間の類似度 (符号間類似度) も対応する 2 値多重音響特徴ベクトル間の類似度 (式 (3)) により定義するものとする.

2.5 探索の高速化

2.5.1 DP 照合

提案探索法の説明に先立ち, 本探索で用いた DP 照合について説明する. $\mathbf{T} = [c_T(1), c_T(2), \dots, c_T(n)]$ を蓄積符号系列, $\mathbf{P} = [c_P(1), c_P(2), \dots, c_P(m)]$ を参照符号系列とする. そして, 探索窓が蓄積符号系列中のある位置に置かれたとき, その中の部分符号系列 (以下, 単に部分系列と呼ぶ) を $\mathbf{W} = [c_W(1), c_W(2), \dots, c_W(w)]$ と表す. \mathbf{P} と \mathbf{W} の DP 照合において, $c_P(x)$ と $c_W(y)$ の累積類似度 $S(x, y)$ を,

$$S(x, y) = \max \begin{bmatrix} S(x, y-1) \\ + s(c_P(x), c_W(y)), \\ S(x-1, y-1) \\ + 2s(c_P(x), c_W(y)), \\ S(x-1, y) \\ + s(c_P(x), c_W(y)), \end{bmatrix} \quad (4)$$

ただし,

$$S(x, 1) = S(x-1, 1) + s(c_P(x), c_W(1)), \quad (5)$$

$$S(1, y) = S(1, y-1) + s(c_P(1), c_W(y)), \quad (6)$$

$$S(1, 1) = 2s(c_P(1), c_W(1)) \quad (7)$$

とする. ここで, $s(c_P(x), c_W(y))$ は $c_P(x)$ と $c_W(y)$ の符号間類似度であり 0 から 1 の値をとるものとする. なお, 極端な符号の対応付けを防ぐため, 適当な整合窓を用いることとし, 整合窓外の格子点 (x, y) を通るパスは無視し, 式 (4), (5), (6), (7) では, そのようなパスを選択しないこととした. そして, \mathbf{P} と \mathbf{W} の系列間類似度は $S(m, w)$ を

$$\frac{S(m, w)}{m + w} \quad (8)$$

として正規化したものとする.

2.5.2 提案探索法

次に, 2.1 で述べた探索の高速化手法について述べる. 2.1 で述べた探索は単に DP 照合を用いたずらし照合法 (以下, 単にずらし照合法と呼ぶ), すなわ

(注3): ここでは, snapshot 内で各周波数のパワーを正規化し, ある周波数について, そのパワーがきい値より大きい場合, その周波数の音が存在するとした.

ち、蓄積符号系列上で探索窓を順に1符号ずつずらしながら探索窓内の部分系列と参照符号系列をDP照合で照合し、蓄積符号系列のすべての部分系列と照合を行う手法でも行うことができる。しかし、より高速な探索のため、ここで説明する探索法を提案する。この手法の高速化の要点は、参照符号系列と蓄積符号中の部分系列の照合を類似し得る部分系列との照合に限定することと、この照合にかかる計算量自体を軽減することにある。なお、符号間の類似度が同じであれば提案探索法とずらし照合法では同一の探索結果が得られる。しかし、符号間の類似度が“スパース”であるとき、提案探索法ではより高速に探索結果が得られる。なお、符号間の類似度がスパースとは符号間類似度が0より大きい符号と符号の対が少ないことを意味している。

ここで、符号間の類似度を表す参照テーブルを導入する。本論文ではこれを類似度行列と呼ぶ。類似度行列の (u, v) 成分は符号 u と v の符号間類似度を表す。この類似度行列はずらし照合法においても符号間類似度の参照に用いられる。しかし、この類似度行列をスパースにすること、すなわち、ほとんどの成分の値が0となるような類似度定義を用いることで、提案探索法ではより高速な探索が行える。

図2に提案探索法の概要を示す。本手法では、まず各符号について蓄積符号系列中での出現位置を調べておく。そして、探索窓の位置は、図2(a)に示すように、 P の一符号と類似する符号を含む位置に合わせる。図2(a)では、 P の i 番目の符号が探索窓内の j

番目の符号に類似している。ここで符号が類似するとは符号間類似度が0より大きいということである。そして、蓄積符号系列中の探索窓内の部分系列と参照符号系列を図2(b)のようにして照合する。図2(b)のグラフはDP照合での符号の対応付けを行うグラフを表しており、水平方向の軸(x 軸)は P の各符号に、垂直方向の軸(y 軸)は探索窓内の符号系列の各符号に対応する。なお、図中の2重線は整合窓の境界を表す。そして、通常のDP照合では整合窓内の各格子点について累積類似度を計算するが、提案手法では、後述するように、白点については累積類似度は計算せずとも0となる。このようにして、累積類似度を計算する点の数を減らすことにより、DPマッチングの計算量を削減しながら、参照符号系列と類似する符号をもつ部分系列とのみ照合を行うことで探索を高速に行う。

以下に、提案手法を詳細に説明する。

(ステップ1) 各符号について T での出現位置を調べる。そして、符号の種類ごとに、その符号の出現位置をリストとして保持する。このリストは(ステップ4)の探索窓の位置合せに用いる。

(ステップ2) T の q 番目を先頭とする探索窓で未照合か照合済かを表すフラグを $f_q(1 \leq q \leq n-w+1)$ とし、すべての q について f_q を未照合に初期化。

(ステップ3) 照合においては、ある d について、 $x+y < d$ となる格子点 (x, y) では累積類似度は0となるように、直線 $x+y = d$ 上の格子点 (x, y) から以下の(ステップ4)の手順に従い位置合せを行い、照合する。具体的には、 d の値を2から z まで1ずつ増やししながら(ステップ4)を繰り返す。 z は $(1-s_t)(m+w)+2$ より小さい最大整数である。

(ステップ4) 1から $\min(m, d-1)$ までのすべての整数 i について、1から順に以下を行う。 j を $d-i$ とし、格子点 (i, j) が整合窓内の点ならば、 $c_P(i)$ と類似する符号を類似度行列から抜き出し、その符号の出現位置を(ステップ1)のリストから順に選ぶ。選ばれた出現位置を k としたとき、 $1 \leq k-j+1 \leq n-w+1$ かつ f_{k-j+1} が未照合であるなら、探索窓の先頭位置を T の $k-j+1$ 番目に合わせ(図2(a))、参照符号系列と探索窓内の部分系列を図2(b)のようにして照合し、 f_{k-j+1} を照合済とする。なお、 $k-j+1 < 1$ または $k-j+1 > n-w+1$ の場合や、 f_{k-j+1} がもとより照合済の場合は照合しない。そして、照合の結果、系列間類似度が s_t より大きい場合、現在の探索窓の位置に類似区間が見つかったとする。上記の照合

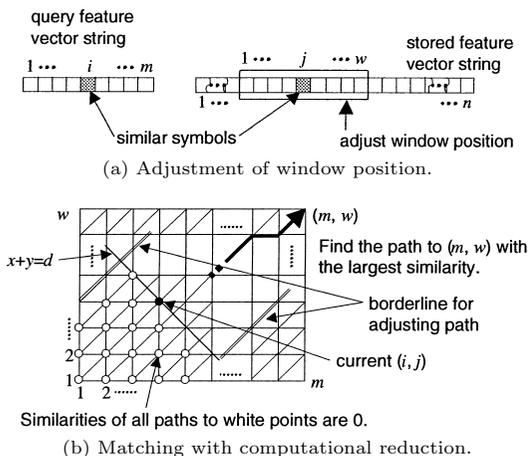


図2 提案探索法の概要
Fig. 2 Overview of proposed search method.

を, $c_P(i)$ に類似するすべての符号について (ステップ 1) のリストから得られる T 上でのすべての出現位置を用いて行う (ステップ 4) では, 以上をすべての i について順に行う. □

図 2 (b) の照合においては, 白点の格子点の累積類似度は 0 とし, 整合窓内の白点以外の点についてのみ累積類似度を計算する. なお, 直線 $x + y = d$ 上の格子点 (i, j) 及び格子点 (i, j) より右下の格子点については, 整合窓内であれば累積類似度の計算が必要である.

提案手法では以下の (1), (2) 及び更に (3) により, 累積類似度 $S(x, y)$ の計算回数を減らし, 高速化を図っている.

(1) (ステップ 3) で d の値の上限を z とし, 系列間類似度の上限値が s_t 以下になった場合, 探索を打ち切る.

(2) (ステップ 4) では, 探索窓が未照合の位置に合わされた場合に照合が行われる. ゆえに, この照合においては, 図 2 (b) で白点で表される格子点 (x, y) , すなわち, x と y が既に i と j の値として用いられたような x と y については $c_P(1), c_P(2), \dots, c_P(x)$ と $c_W(1), c_W(2), \dots, c_W(y)$ の間に類似する符号はなく, $S(x, y)$ は 0 となり, $S(x, y)$ の計算を行わない.

(3) 図 2 (b) の照合では白点以外の格子点 (x, y) について, 式 (4), (5), (6), (7) に従い, x と y を 1 から, 各々, m と w まで増やしていき $S(x, y)$ を順次求め, 各格子点への生き残りパスを求めていく. ここで, 格子点 (x, y) を通る格子点 (m, w) へのパスの累積類似度の上限値 $S_u(x, y)$ は

$$S_u(x, y) = S(x, y) + (m - x) + (w - y) \quad (9)$$

となる. そして,

$$\frac{S_u(x, y)}{m + w} \leq s_t \quad (10)$$

の場合, (x, y) を通るパスでは s_t を超えない. そこで (ステップ 4) の図 2 (b) の照合においては, $S(x, y)$ を求めていく過程で, $\frac{S_u(x, y)}{m + w} \leq s_t$ となった格子点 (x, y) を通るパスの探索は打ち切る. そして, すべてのパスの探索が打ち切られた場合, その探索窓の位置での照合を打ち切る.

2.5.3 計算量の比較

ここでは提案探索法とずらし照合法の計算量について述べる. 両探索法においては $S(x, y)$ の計算が主である. そして, この計算は式 (4), (5), (6), (7) により行われ, 計算量は定数オーダーである. そこで $S(x, y)$

の計算回数 ($S(x, y)$ を計算する格子点の数) で両手法の計算量を見積もる. 以降, 簡単のため, 式 (4), (5), (6), (7) による 1 回の $S(x, y)$ の計算を符号照合と呼ぶ.

探索窓の幅を w , 蓄積符号系列の長さを n とし, このとき, 一度の DP 照合において整合窓内にある格子点の数を c とすると, ずらし照合法における符号照合回数は合計で

$$c(n - w + 1) \quad (11)$$

となる. 提案探索法については, 最悪の場合はずらし照合法と同じ符号照合回数が必要である. しかし, 符号と符号が類似しない確率を平均 p_d とすると, 平均的には符号照合回数は, 2.5.2 の (1) と (3) による削減を無視しても,

$$\left(c - \sum_{l=1}^c p_d^l \right) (n - w + 1) \quad (12)$$

となる. 式 (11) と式 (12) より, 類似度行列がスパースであればあるほど, 提案探索法の符号照合回数はずらし照合法に比べて少なくなることがわかる. また, ずらし照合法に対する提案探索法の符号照合回数の割合は

$$\frac{\left(c - \sum_{l=1}^c p_d^l \right) (n - w + 1)}{c(n - w + 1)} = 1 - \frac{1}{c} \sum_{l=1}^c p_d^l \quad (13)$$

であり, c が小さいほど, その割合は小さくなり, 符号照合回数の削減の効果が大きいことがわかる.

3. 実験

2 値多重音響特徴ベクトルを用いた探索の探索精度の評価及び, 2 値多重音響特徴ベクトルを用い類似度行列をスパース化した場合の探索精度及び提案探索法による探索速度の評価のための実験を行った.

実験は実験用の音楽 DB を用い, この中から実際に類似音楽を探索することで行った. 音楽 DB には, ポップス, クラシック, インストルメンタルなどの様々なジャンルの多重奏の楽曲 216 曲の音響信号を用いた. なお, ここでは, これらの 216 曲の録音の一つひとつを楽曲と呼ぶ. この DB の音響信号の長さは合計で約 480 分であった. なお, 音響信号は 48 kHz サンプリング, 16 bit 量子化, ステレオ録音のものをモノラル

化して用いた．実験用の参照信号については，参照信号の長さにより，探索精度，探索速度及び速度向上率が変化するため，各実験で示す長さごとに15個ずつ用意し探索実験を行った．各参照信号は複数の三重奏アンサンブル音楽の音響信号から切り出した楽曲の一部である．なお，これらのアンサンブル音楽については，それぞれ，参照信号を切り出した楽曲（原曲）を含め，計8曲の類似楽曲がDBに含まれている．これらは，原曲と，原曲に移調，楽器変換，テンポ変換のアレンジを行った各編曲を実際に演奏家に2回ずつ演奏してもらい録音したものである．実験ではこれらの八つの類似楽曲（目的楽曲）中の参照信号と同じ演奏区間（目的区間）の探索及び類似楽曲の検索を行うこととした．すなわち，各参照信号に対し，8個の目的区間及び目的楽曲があることになる．なお，これらのアンサンブル音楽の演奏にはピアノ，フルート，オーボエ，バスーン，バイオリン，チェロを用いた．そして，楽器変換の際には一つまたは二つのパートを異なる楽器で演奏した．

探索精度については区間単位，楽曲単位で評価することとした．区間単位の探索精度については，各探索において，検出された区間のうちの正答の割合と，目的区間のうち検出されたものの割合が等しくなる探索しきい値でのこれらの割合の値とした．なお，両割合が等しくなることが複数回ある場合は，両割合が等しくなったときの値のうちの最大値とした．ある探索結果を正答と判断する基準については，その探索結果の区間がある目的区間と重複率[11]が0.5以上のとき，この探索結果を正答とするとした．なお，区間 a と区間 b の重複率は

$$\text{重複率} = \frac{(a \cap b) \text{ の区間長}}{(a \cup b) \text{ の区間長}} \quad (14)$$

である．また，目的区間が検出されたかの判断も探索により検出されたある類似区間との重複率が0.5以上のとき，検出されたとする．また，検出された区間のうち，重複率が0.5以上で重なり合う区間は一つの検出結果とみなした．

楽曲単位の探索精度については，楽曲中に類似区間が一つでも検出された場合，その楽曲は検出されたとする．そして，楽曲単位の場合の探索精度は，各探索において，検出楽曲中の目的楽曲の割合と，目的楽曲のうちの検出された割合が等しくなる探索しきい値でのこれらの割合の値とした．なお，両割合が等しくなることが複数回ある場合は，両割合が等しくなったと

きの値のうちの最大値とした．

3.1 探索精度の評価実験

探索精度の比較評価のため (a) スペクトル特徴を用いた場合 (b) 2値多重音響特徴ベクトルを用いた場合 (c) 2値多重音響特徴ベクトルを用いたスパースな類似度行列を用いた場合の3通りの場合について探索精度を調べた．

(a) のスペクトル特徴は，2.2.1の帯域通過フィルタを用い，探索時の極端な速度低下を避けるため，分析区間を400msとし100msごとに抽出した．そして，符号化はLBGアルゴリズムを用いて学習した符号帳に基づくベクトル量子化(VQ)とし，符号間類似度はVQコードが同じ場合に1，それ以外は0とした．なお，ベクトル量子化の際の符号帳の大きさは，2値多重音響特徴ベクトルとの比較のため $4096 (= 2^{12})$ とした．

そして (b) の場合は式(3)に従った符号間類似度により定義される類似度行列を用い (c) の場合においては (b) の類似度行列の0.85より小さい成分の値を0として得られるスパースな類似度行列を用いた．なお，2値多重音響特徴ベクトルを用いた場合の符号長はDB全体で94744であった．

実験では3通りいずれの場合も，各特徴及び類似度に基づき2.1の探索手続きにより探索した．ただし，参照符号系列の符号長 m に対し，探索窓の符号長 w については， $w = m$ とし，DP照合における整合窓については， $|x - y| < \max(2, [0.06m])$ となる格子点 (x, y) は整合窓内の点であるとして定義した． w は m に対し可変であるが，本探索実験においては参照信号と同じ演奏区間を探索することを目的としているため $w = m$ とした．

なお (b) と (c) の場合は参照符号系列の2値多重音響特徴ベクトルを1ビットずつ12通りにシフトしたのものについても探索を行った．

表1，表2に各特徴における区間単位の探索精度と楽曲単位の探索精度を示す．参照信号の長さは2値多重音響特徴ベクトルで抽出した際の符号長，または，実際の演奏における音符による長さを表している．上述のように，各長さについて15個の参照信号を用いて評価しており，表の探索精度はその平均値である．なお，実際の参照信号の平均の長さは符号長10，30，50，70の場合で，各々，約3，8，14，19秒であった．このとき，標準偏差は，各々，約0.2，0.5，0.9，0.9であり，いずれの場合も，平均値からの偏差は平均値

表 1 探索精度の比較 (区間単位)
Table 1 Search accuracy (by location).

参照信号の長さ	(a) スペクトル特徴	(b) 2 値多重音響特徴ベクトル	(c) 2 値多重音響特徴ベクトル (スパース化)
符号長 10	28.1 %	44.4 %	39.5 %
符号長 30	46.9 %	70.1 %	63.8 %
符号長 50	53.9 %	71.6 %	71.2 %
符号長 70	55.0 %	82.1 %	78.7 %
四分音符 24 音符分	53.3 %	72.9 %	70.9 %

表 2 探索精度の比較 (楽曲単位)
Table 2 Search accuracy (by title).

参照信号の長さ	(a) スペクトル特徴	(b) 2 値多重音響特徴ベクトル	(c) 2 値多重音響特徴ベクトル (スパース化)
符号長 10	45.0 %	57.9 %	54.7 %
符号長 30	55.7 %	81.8 %	77.2 %
符号長 50	59.8 %	83.8 %	81.3 %
符号長 70	62.5 %	89.3 %	86.0 %
四分音符 24 音符分	64.6 %	80.6 %	82.0 %

の 20%以下であり、平均値からの変動はほとんどなかった。四分音符 24 音符分の場合、参照信号の長さは平均約 21 秒であり、符号長の平均は約 74 であったが、符号長の最小値は 21、最大値は 114、標準偏差は 34.7 とばらつきがあった。

表 1、表 2 のいずれにおいても、実用上重要と思われる符号長が 30 から 70、または、四分音符 24 音符分の場合において 2 値多重音響特徴ベクトルの探索精度はスペクトル特徴を大きく上回っている。また、2 値多重音響特徴ベクトルで類似度行列をスパース化しても探索精度はあまり低下しないことがわかる。なお、四分音符 24 音符分の場合、表 1 においては (b) の場合の探索精度は (c) の場合の探索精度より上であるが、表 2 においては (c) の場合の探索精度の方が高くなっている。(c) の場合の類似度行列のスパース化は、異なる曲の識別にはより有効である場合があるが、例えば同じ曲内の繰返しのように、似ているが異なる区間を識別する場合には、その識別ができなくなる場合がある。今回の四分音符 24 音符分の場合、繰返しにより類似する区間が同一曲内に複数回含まれるようなクエリーがあり、そして、目的区間以外は不正解とみなされることから、(c) の場合に区間単位の探索精度が楽曲単位の探索精度に比べ低くなってしまったものと思われる。

3.2 探索速度の評価実験

次に提案探索法の探索速度についての評価実験結果を述べる。ずらし照合法においては 3.1 の (b) の場合について探索を行い、提案探索法においては (c) の

表 3 符号照合回数の比較 (区間単位)
Table 3 Average number of symbol matchings required in the location search.

参照信号の長さ	ずらし照合法	提案探索法	照合回数比
符号長 10	31,180,800	728,282	1/43 (2.3 %)
符号長 30	93,434,880	7,895,351	1/12 (8.5 %)
符号長 50	149,468,160	22,677,253	1/6.6 (15.2 %)
符号長 70	457,962,240	114,775,998	1/4.0 (25.1 %)
四分音符 24 音符分	577,487,245	208,073,344	1/2.8 (36.0 %)

表 4 符号照合回数の比較 (楽曲単位)
Table 4 Average number of symbol matchings required in the title search.

参照信号の長さ	ずらし照合法	提案探索法	照合回数比
符号長 10	31,180,800	777,273	1/40 (2.5 %)
符号長 30	93,434,880	8,765,645	1/11 (9.4 %)
符号長 50	149,468,160	24,099,514	1/6.2 (16.1 %)
符号長 70	457,962,240	119,386,645	1/3.8 (26.1 %)
四分音符 24 音符分	577,487,245	212,282,596	1/2.7 (36.8 %)

場合について探索を行った。なお、探索窓の幅や整合窓については 3.1 と同じ条件であり、両探索手法とも 12 通りにシフトした参照符号系列についても探索を行っている。探索精度はずらし照合法においては 3.1 の (b) の場合、提案探索法においては (c) の場合と同じになる。以下は、これらの条件下での探索速度の評価となる。

表 3、表 4 に表 1、表 2 の各探索精度における両手法の 1 回の探索における符号照合回数の平均値を、表 5、表 6 に 1 回の探索にかかった CPU 時間の平均を示す。表 3、表 4 の照合回数比はずらし照合法の符

表 5 探索時間の比較 (区間単位)

Table 5 Average search time required in the location search.

参照信号の長さ	ずらし照合法	提案探索法	速度比
符号長 10	3.07 s	0.17 s	18.1
符号長 30	10.38 s	1.62 s	6.4
符号長 50	17.81 s	4.57 s	3.9
符号長 70	49.60 s	22.01 s	2.3
四分音符 24 音符分	66.90 s	40.14 s	1.7

表 6 探索時間の比較 (楽曲単位)

Table 6 Average search time required in the title search.

参照信号の長さ	ずらし照合法	提案探索法	速度比
符号長 10	3.07 s	0.17 s	18.1
符号長 30	10.38 s	1.72 s	6.0
符号長 50	17.81 s	4.75 s	3.7
符号長 70	49.60 s	22.58 s	2.2
四分音符 24 音符分	66.90 s	40.79 s	1.6

号照合回数に対する提案探索法の符号照合回数の割合を示している。そして、表 5、表 6 の速度比はずらし照合法による探索速度を 1 としたときの提案探索法の探索速度を示している。なお、CPU 時間の計測には PC (CPU: Pentium 4 2 GHz, L2 キャッシュ 256 KB, OS: Red Hat Linux 7.2) を用いた。区間単位、楽曲単位の探索精度のいずれの場合にも提案探索法の符号照合回数はずらし照合法の数%から 40%程度であり、CPU 時間にして 18 倍から 1.6 倍の高速化を達成している。符号照合回数の削減率よりも CPU 時間における高速化の割合が小さいのは、式 (9) と式 (10) の計算のオーバーヘッドによるものと考えられる。なお、表 5、表 6 の提案探索法の CPU 時間には (ステップ 1) の計算にかかる CPU 時間は含まれていない。これは、探索前に、各蓄積符号系列ごとに各符号の蓄積符号系列中での出現位置を求めておくことも可能だからである。なお (ステップ 1) の計算にかかる CPU 時間は DB 全体に対し、0.01 秒とわずかであり、探索ごとに行っても探索時間はほとんど変わらない。

4. む す び

本論文では多重奏音響信号による類似音楽探索のための 2 値多重音響特徴ベクトルを提案し、それによる類似音楽探索法を提案した。そして、本探索における高速化手法について述べた。216 曲の実験用音楽 DB を用いた探索実験では、スペクトル特徴に比べて 2 値多重音響特徴ベクトルの方が高精度であることが示された。例えば、楽曲単位の探索において、クエリーの

符号長が 70 (平均 19 秒) のとき、スペクトル特徴を用いた場合に 62.5% であった精度が、提案の 2 値多重音響特徴ベクトルを用いることにより 89.3% にまで向上した。また、本高速化手法においては精度をほとんど損なわずに、DP 照合を用いたずらし照合法と比べて約 1.6 倍から 18 倍の高速化を達成した。

なお、本論文の実験では、原曲と、原曲に移調、楽器変換、テンポ変換のアレンジを加えた編曲について、2 度の異なる演奏を録音したものを類似楽曲として探索した。しかし、これらの類似楽曲においては和音の構成音は変更されていない。和音の変更に対する探索の頑健性については、今後、評価が必要である。また、今後は、提案探索法による探索の更なる高速化、高精度化のための類似度行列の構成法、及び連続 DP への高速化手法の適用などの検討も行う予定である。

謝辞 日ごろ御指導を頂く NTT コミュニケーション科学基礎研究所の管村昇所長、片桐滋主幹研究員、牧野昭二部長に感謝する。そして、日ごろ御協力を頂く同研究所メディア認識研究グループの諸氏に感謝する。また、本研究の遂行において議論、御協力頂いたブライアントスに感謝する。

文 献

- [1] 柏野邦夫, ガビンズミス, 村瀬 洋, “ヒストグラム特徴を用いた音響信号の高速探索法—時系列アクティブ探索法” 信学論 (D-II), vol. J82-D-II, no.9, pp.1365–1373, Sept. 1999.
- [2] 黒住隆行, 柏野邦夫, 村瀬 洋, “時系列アクティブ探索法のための特徴わずみに頑健な確率ディザポートニング” 信学論 (D-II), vol. J84-D-II, no.8, pp.1817–1825, Aug. 2001.
- [3] 木村昭悟, 柏野邦夫, 黒住隆行, 村瀬 洋, “グローバルな枝刈りを導入した音や映像の高速探索” 信学論 (D-II), vol. J85-D-II, no.10, pp.1552–1562, Oct. 2002.
- [4] 永野秀尚, 柏野邦夫, 村瀬 洋, “多数の小区間信号の探索に基づく背景音楽の探索” 信学技報, PRMU2002-86, WIT2002-29, Sept. 2002.
- [5] 成田智也, 杉山雅英, “楽曲の高速検索手法の検討” 信学技報, SP2000-16, June 2000.
- [6] 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” 信学論 (D-II), vol. J77-D-II, no.8, pp.1543–1551, Aug. 1994.
- [7] A. Ghias, J. Logan, D. Chamberlin, and B.C. Smith, “Query by humming: Musical information retrieval in an audio database,” Proc. ACM Multimedia '95, pp.231–236, San Francisco, USA, Nov. 1995.
- [8] L.A. Smith, R.J. McNab, and I.H. Witten, “Sequence-based melodic comparison: A dynamic programming approach,” in Melodic Similarity: Concepts, Procedures, and Applications, vol.11 of Com-

- puting in Musicology, eds. W.B. Hewlett and E. Selfridge-Field, pp.101-117, The MIT Press, England, 1998.
- [9] 園田智也, 後藤真孝, 村岡洋一, “WWW 上での歌声による曲検索システム” 信学論 (D-II), vol.J82-D-II, no.4, pp.721-731, April 1999.
- [10] N. Kosugi, Y. Nishihara, S. Kon'ya, M. Yamamuro, and K. Kushima, “Music retrieval by humming—Using similarity retrieval over high dimensional feature vector space,” Proc. IEEE PACRIM '99, pp.404-407, Victoria, Canada, Aug. 1999.
- [11] 橋口博樹, 西村拓一, 張 建新, 滝田順子, 岡 隆一, “モデル依存傾斜制限型の連続 DP を用いた鼻歌入力による楽曲信号のスポットティング検索” 信学論 (D-II), vol.J84-D-II, no.12, pp.2479-2488, Dec. 2001.
- [12] A.L. Uitdenbogerd and J. Zobel, “Manipulation of music for melody matching,” Proc. ACM Multimedia '98, pp.235-240, Bristol, UK, Sept. 1998.
- [13] T. Crawford, C.S. Iliopoulos, and R. Raman, “String-matching techniques for musical similarity and melodic recognition,” in Melodic Similarity: Concepts, Procedures, and Applications, vol.11 of Computing in Musicology, eds. W.B. Hewlett and E. Selfridge-Field, pp.73-100, The MIT Press, England, 1998.
- [14] 後藤真孝, 村岡洋一, “音響信号を対象としたリアルタイムビートトラッキングシステム—コード変化検出による打楽器音を含まない音楽への対応” 信学論 (D-II), vol.J81-D-II, no.2, pp.227-237, Feb. 1998.
- [15] 柏野邦夫, 音楽音響信号を対象とする聴覚的情景分析に関する研究, 博士論文, 東京大学, March 1995.
(平成 14 年 4 月 30 日受付, 15 年 3 月 19 日再受付)



永野 秀尚 (正員)

平 6 阪大・基礎工・情報卒。平 8 同大大学院修士課程了。同年 NTT に入社。現在, NTT コミュニケーション科学基礎研究所に所属。マルチメディア情報のための信号処理, パターン認識, 情報検索に興味をもつ。情報処理学会, IEEE 各会員。



柏野 邦夫 (正員)

平 2 東大・工・電子卒。平 7 同大大学院電気工学専攻博士課程了。工博。同年 NTT に入社。現在, NTT コミュニケーション科学基礎研究所主任研究員。音響信号の認識・分離・探索, 及び情報統合の研究に従事。メディア情報を対象とする信号処理及び知識処理に興味をもつ。平 5 情報処理学会奨励賞, 平 9 人工知能学会全国大会優秀論文賞, 平 11 日本音響学会栗屋潔学術奨励賞, 平 13 本会ソサイエティ論文賞, 平 14 本会業績賞各受賞。情報処理学会, 日本音響学会, 人工知能学会, IEEE, 各会員。



村瀬 洋 (正員)

昭 53 名大・工・電子卒。昭 55 同大大学院修士課程了。同年日本電信電話公社 (現 NTT) に入社。以来, 文字・図形認識, コンピュータビジョン, マルチメディア認識の研究に従事。平 4 から 1 年間米国コンピュータ大客員研究員。平 13 NTT コミュニケーション科学基礎研究所メディア情報研究部部長。平 15 名古屋大学大学院情報科学研究科教授, 現在に至る。工博。昭 60 本会学術奨励賞, 平 4 電気通信普及財団テレコムシステム技術賞, 平 6 IEEE-CVPR 国際会議最優秀論文賞, 平 7 情報処理学会山下記念研究賞, 平 8 IEEE-ICRA 国際会議最優秀ビデオ賞, 平 13 高柳記念電子科学技術振興財団高柳記念奨励賞, 平 13 本会ソサイエティ論文賞, 平 14 本会業績賞, 平 15 文部科学大臣賞各受賞。情報処理学会, IEEE 各会員。