Vol.2010-MUS-85 No.8
2010/5/28

複数F0候補を用いた音楽音響信号からのハミング楽曲検索

小杉 優[†1] 伊藤彰則[†1]
伊藤 仁[†1,*1] 牧野正三[†1,*2]

本稿では，音響信号しか存在しない音楽データに対するハミング楽曲検索手法を提案する．ほとんどのハミング楽曲検索方式では，楽曲のデータベースとして楽譜情報や MIDI のように記号化されたメロディ情報を想定しているが，実際には録音楽曲しか存在しない場合も多い．我々のシステムでは，多くの楽器音が含まれるモノラル音楽音響信号からデータベースを作成することを想定している．音響信号から，フレーム毎に基本周波数 (F0) を推定することでデータベースを作成する．F0 推定誤りによる検索精度の低下を抑えるため，複数の F0 候補を考慮する点が提案法の特徴である．実験結果から，単一の F0 候補を用いる方法と比較して，複数 F0 を考慮することで約 15 ポイントの検索性能向上が得られた．

A Query-by-Humming Music Information Retrieval from Audio Signals based on Multiple F0 Candidates

Yu Kosugi,[†1] Akinori Ito,[†1] Masashi Ito[†1,*1]
and Shozo Makino [†1,*2]

In this paper, we propose a query-by-humming (QbH) system that retrieves musical pieces given as audio signals. Most conventional QbH systems assume that the symbolic melody information is given a priori, which is not always true. In our system, the database for retrieval is generated from 1ch audio signal that contains many sounds. We generate the database by estimating fundamental frequencies (F0) of the audio signals frame by frame. To improve the retrieval accuracy, we exploit multiple F0 candidates to absorb the impact of F0 estimation errors. From the experiment, we obtained about 15 points of improvement by using multiple F0 candidates, compared with the QbH system with only one F0 candidate.

1. はじめに

近年，楽曲検索の必要性が高まっている．これまでの音楽情報検索は，テキスト情報の検索を目的として発展しているが，ユーザーの利便性を考え，メロディーのハミング入力による音楽検索に関する研究が進められている[1)]．それに伴い，楽曲データベースの構築をどのようにするのかという問題が発生する．ハミングを入力とした従来の楽曲検索システムでは，データベースとして，楽曲のメロディーライン等の基本周波数の時系列情報 (例えば，楽譜や MIDI 等) があらかじめ存在していると仮定している[2)]．しかし，実際に市販されている音楽，楽曲情報は，CD，MP3 などメディアやファイル形式に違いはあれど，音響信号として音楽家から提供されるのが一般的である．つまり，事前に楽譜などの音符情報が与えられずに，音響信号しか存在しない場合のほうが多いといえる．したがって，従来のハミング検索手法を実際の楽曲に適用する場合，音響信号のみしかない楽曲をデータベースに登録する場合，何らかの形でデータベースを新たに構築する必要がある．

自動でデータベース構築を行う場合，CD 等に含まれている音楽音響信号のような，複数の楽器音が混在する音声信号からのメロディーラインの音の高さ (以下，基本周波数，F0) 推定が必要となる．メロディーラインの F0 抽出の研究としては，後藤によって提案されている PreFEst[3)] や，亀岡らによる調波時間構造化クラスタリング (HTC)[5)] などがある．しかしながら，未知の混合音を対象とした場合，いずれの手法も完全な F0 抽出はできていない．

このように不完全な精度の F0 抽出結果を用いて高精度なハミング検索を行うためには，F0 抽出結果を一意に定めずに，複数の F0 候補を考慮して検索を行う方法が考えられる．Heo らは，ユーザのハミング音声に対する F0 抽出の誤りを補償するために，複数の F0 抽出結果を考慮した検索法を提案している[6)]．本研究では，この基本的なアイデアをデータベースの F0 抽出に応用し，音楽音響信号しか存在しない楽曲に対して，複数 F0 候補選択に基づくハミング楽曲検索システムの構築を目指す．

この目的の達成の為に，次のような課題に取り組む必要がある．

†1 東北大学
Tohoku University
*1 現在，東北工業大学
Presently with Tohoku Institute of Technology)
*2 現在，東北文化学園大学
Presently with Tohoku Bunka Gakuen University)

Vol.2010-MUS-85 No.8
2010/5/28

- **メロディーラインの候補が複数個存在するデータベースの自動構築**
  従来の楽曲データベースは楽譜等のメロディーラインの時系列情報が記載されたものが一般的である．前述の通りこのようなデータベースを自動生成するのは困難である．そこで，本研究では完全にメロディーラインの F0 を抽出するのではなく，メロディーラインの候補を複数個用意し，データベースとして構築する．
- **上述のデータベースを用いたハミング楽曲検索手法の実現**
  従来の楽曲検索とは異なるデータベースを用いているので，従来のハミング楽曲検索で提案されている手法をそのまま適応できない．そこで，複数 F0 候補を持つデータベースと歌唱入力に対しての，楽曲検索手法を構築する．

2. データベース構築

本研究におけるデータベース生成は以下のようにして行われる．

(1) 音響信号の周波数解析
(2) F0 の存在確率の推定
(3) F0 候補の選択
(4) データベースの作成

F0 推定には，後藤によって提案されている「PreFEst(Predominant F0 Estimation)」[3] の PreFEst-core 部を用いて，周波数 F に基本周波数 (F0) が存在する確率密度関数 $p_{F0}(F)$ を導出する．本研究では，この確率密度関数の値の高いピークをいくつか選択し，それを F0 の候補とする．なお，高い順に第 1 候補，第 2 候補のようにランク付けを行う．このようにして，複数個の F0 候補を楽曲のデータとし，データベースの構築が実現できる．

2.1 音響信号の周波数解析

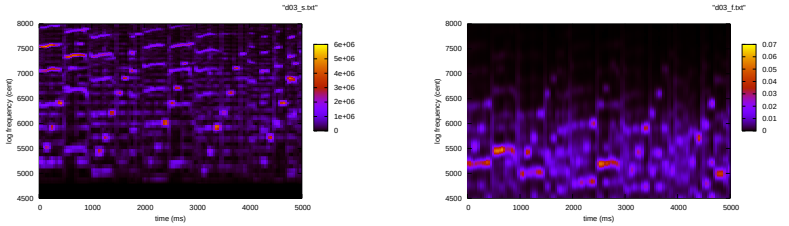
まずデータベース楽曲の音響信号を短時間フーリエ変換を行いスペクトログラムを生成する．このスペクトログラムより正しいメロディーラインを抽出することが必要となる．

2.2 基本周波数 (F0) の存在確率の推定

上述のスペクトログラムのデータでは，メロディーラインの高調波成分や，他の楽器の周波数成分が混在しており，このままではどの周波数に F0 が存在しているかわからない．このような例を**図 1**(a) に示す．

そこで，完全な F0 の存在する周波数を特定するのではなく，どの周波数にどの程度の確率で F0 が存在するのかを，確率密度分布で表すことを考える．

この F0 存在確率密度関数は，後藤によって提案されている「PreFEst (Predominant F0 Estimation)」[3] の PreFEst-core 部を用いることで算出する．この PreFEst-core では，モノラルの音楽音響信号に対し，その信号中のメロディーラインの推定を実現している．この技術では高調波構造をモデル化し，EM アルゴリズムにより学習を行うことで，ある時刻 t における周波数 F に基本周波数 (F0) が存在する確率密度関数 $p^t_{F0}(F)$ を推定している．この操作をすべての時刻 t に対し行うことにより，楽曲の確率密度関数が求まる．



(a) Spectrogram　　(b) Probability density of F0 ($p^t_{F0}(F)$)

図 1 スペクトログラムと F0 存在確率密度の例
Fig. 1 An example of spectrogram of music signal and its F0 probability density functions

PreFEst によって求めた確率密度関数の例を図 1(b) に示す．図 1(a) のように，スペクトログラム上では，メロディーラインを歌唱するボーカル音声の高調波成分だけでなく，他の伴奏の楽器音による基本周波数成分及び高調波成分が混在しており，メロディーラインの基本周波数が比較的目立ちにくくなっている．しかし，スペクトログラムから F0 存在確率密度関数を導くことで，余計な高調波成分が除外されスペクトログラムに比べすっきりしたデータとなる．

2.3 F0 候補の選択

前項で求められた確率密度関数 $p^t_{F0}(F)$ から，メロディーラインの基本周波数 $F0(t)$ の値を求めるためには，$p^t_{F0}(F)$ が最大となるような F の値を求めればよい．

$$F0(t) = \operatorname*{argmax}_F p^t_{F0}(F) \tag{1}$$

しかし，F0 の確率密度関数において，同時に鳴っている音の基本周波数に対応する複数のピークが拮抗した場合，それらのピークが確率密度関数として選ばれてしまうことがある．また，真の F0 の 2 倍音や 1/2 倍音のピークが最大となることもある．

Vol.2010-MUS-85 No.8
2010/5/28

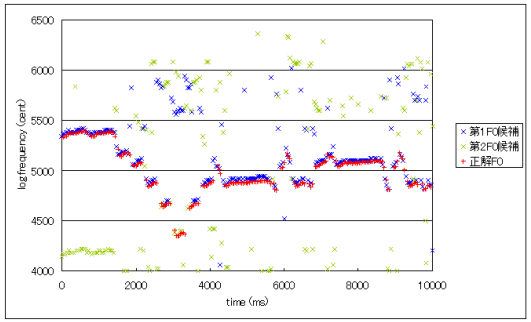


図 2 複数 F0 候補データベースの例
Fig. 2 An example of a multiple F0 database

このように，確率密度関数が最大となる周波数は必ずしも F0 の周波数に対応しているとは言えないが，F0 の周波数には (最大でなかったとしても) 比較的大きいピークが存在している可能性が高い．そのため，確率密度関数における複数個のピークのうち，どれかには正しいメロディーラインの F0 がある可能性は高いと考えられる．そこで本研究では，PreFEst-core により求まった確率密度関数から，フレーム毎に $p^t_{F0}(F)$ の値の高いピークをいくつか選択し，それを F0 の候補とすることによりデータベースを構築することを考える．

具体的には，まず確率密度関数でピークをとる f を検出し，f' とする．最も確率密度関数の値が高いときの周波数 f' を第 1F0 候補 $db_1(t)$ として選択する．

$$db_1 = \max_{f'} p^t_{F0}(f') \tag{2}$$

以降，i 番目 $(i = 2, \ldots, n)$ の確率密度関数のピークの周波数を第 iF0 候補周波数 $db_i(t)$ とする．

$$db_i = \max_{f'} p^t_{F0}(f') \quad (\text{但し，} db_1, \ldots, db_{i-1} \text{ は除く}) \tag{3}$$

n は最大で保持する F0 候補の個数をとした．これを楽曲すべての区間において実行し，楽曲データベース db を生成する．

抽出された第 2 位までの F0 候補と，この楽曲の正しいメロディーラインの F0 系列を重ねたものを**図 2** に示す．この通り，ほとんどのフレームにおいて第 1F0 候補が正しいメロディーラインを抽出できていることがわかる．また，図中において 3000 ms あたりでは，第 1F0 候補がメロディーラインに比べ 1200 cent ほど高い周波数帯を抽出してしまっている．これはオクターブ誤りで F0 抽出においてよく見受けられる現象であるが，第 2F0 候補では，正しいメロディーラインの抽出がなされている．

表 1 F0 抽出精度評価実験：実験条件
Table 1 Experimental conditions of the F0 extraction experiment

楽曲サンプリング周波数	モノラル 44.1kHz
入力音声周波数帯域	3000-12000 cent
F0 存在周波数帯域	3600-8400 cent
cent の刻み	20 cent ずつ
時間のシフト幅	40ms
使用楽曲	オリジナル楽曲 8 曲
楽曲総フレーム数	3657 フレーム
正解判定	50 cent 以内

2.4 データベース抽出精度評価実験

データベース中に何個の F0 候補を保持することで精度の高いデータベースが得られるかを調べるため，F0 の値が既知である楽曲に対し F0 候補を抽出し精度を調べた．今回使用した楽曲は，ボーカルの F0 周波数軌跡を入手しやすくするために，オリジナル楽曲 (ポップス) を 8 曲分 (但し，サビ前後の区間のみ) 用意した．このオリジナル楽曲は，一般的な歌唱ありのポピュラー音楽のサビの一部を想定して作成したものである．楽曲作成においては，Roland 社のデジタルオーディオワークステーション SONAR HOME STUDIO XL および付属のソフトウェア音源を使用した．ボーカル音声については，クリプトン・フューチャーメディア社の「VOCALOID2 初音ミク」による合成音声を使用した．

その他の実験条件は**表 1** に示す．

実験結果を**図 3** に示す．横軸は F0 候補の数，縦軸はその候補のいずれかが正解 F0 周波数に対して ±50cent 内にある割合である．F0 候補を 5 個保持することにより正しい F0 を 99.9%抽出できており，5 個程度の F0 候補を抽出することにより高い精度のデータベースが得られることが確認できた．

3. 楽曲検索手法

楽曲検索を行う際には，歌唱者によるテンポの違いと，歌唱者の歌唱キーの違い，楽曲の歌唱部分の違いが問題となるため，マッチングには工夫が必要である．

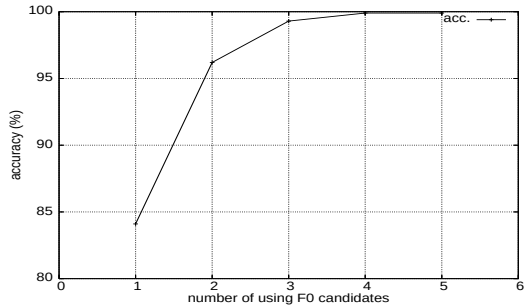


図 3 F0 抽出精度結果
Fig. 3 Results of the F0 extraction experiment

3.1 マッチング

データベース db とのマッチングについては，入力されたハミング歌唱クエリの時系列 $(in(t))$ を db 上でシフトすることにより，最もスコアの高い位置を見つけ検索することにより実現される．なお $in(t)$ は時刻 t における歌唱入力の F0 の値とする．

つまり，入力クエリとデータベース楽曲とを 1 フレームごとに比較し，スコアを算出し，入力クエリを時間軸，周波数軸上でシフトすることで，最もスコアの大きくなる箇所を見つけ，そのときのスコアを楽曲自体のスコアとする．以上の動作を全ての楽曲に行う．なお，周波数軸をシフトすることで，検索クエリの歌唱キーの個人差を吸収することができる．

入力音声の時刻 t における F0 の周波数を $in(t)$，データベースの時刻 t における i 番目の F0 候補の周波数を $db_i(t)$ とすると，楽曲のスコア S は以下のように表わされる．

$$S = \underset{0\leq\tau\leq t_{db}, 0\leq F\leq f_r - f_{in}}{\operatorname{argmax}} \left(\sum_{t=0}^{t_{in}} m\left(in(t)+F, db_1(t+\tau), ..., db_n(t+\tau)\right) \right) \quad (4)$$

ただし，t_{in},t_{db} はそれぞれ入力および楽曲の長さ，f_{in},f_r は $in(t)$,db の周波数帯域の幅を示す．τ，F は検索クエリ $in(t)$ の時間軸，及び周波数軸上のシフト量を表す．また，$m(in, db_1, ..., db_n)$ はマッチング関数であり，後述の節で定義する．

テンポの違いについては，今回は楽曲及び検索クエリのテンポが既知で一定であることを仮定し，正規化を行うことにより対処する．

3.2 スコアリング

マッチング時のスコアリングについては，単純には確率密度関数の値そのものを利用することが考えられる．しかし，F0 存在確率密度関数の性質として，楽器音の混合数が増えることにより，$p^t_{F0}(F)$ のピークが増加し．そして，1 つ 1 つのピークの最大値の値は $p^t_{F0}(F)$ が確率密度関数であるから減少してしまう傾向があることが挙げられる．つまり楽器音が少ないほど，$p^t_{F0}(F)$ の最大値が大きい値をとる傾向になってしまうため，単純に確率密度関数の値を使用することはできない．

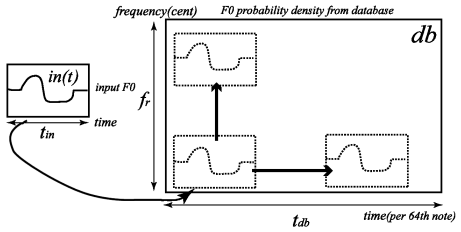


図 4 メロディ探索 (マッチング) の概念図
Fig. 4 Melody search.

そこで，複数 F0 候補間で候補順位の高い F0 候補から順に正解判定を行うような関数に基づき行う．このスコアリング関数では，複数個存在する F0 候補について，第 1F0 候補から順に第 $n(n = 1, 2, 3, 4, 5, \dots)$ 候補まで正解判定を行う．このとき，入力されたクエリの周波数と，F0 候補の周波数が，50 cent 以内であれば正解とする．ただし，各 F0 候補間には重みがつけられており，例えば使用する F0 候補の数が 3 個 $(n = 3)$ の場合，1 位候補が正解であれば 3pt，2 位ならば 2pt，3 位であれば 1pt のように，より上位の F0 候補が選ばれた場合，高いスコアがつくように傾斜配点をしている．

$$m(in, db_1, ..., db_n) = \max_{1\leq i\leq n} \delta_{50}(in(t), db_i(t))(n+1-i)$$

$$\delta_{50}(x, y) = \begin{cases} 1 & |x-y| < 50 \\ 0 & \text{otherwise} \end{cases}$$

このように順位のみでスコアリングすることにより，確率密度関数の値の絶対的な大きさの影響を正規化する．また，順位のみを用いることにより，データベース内に確率値を保存する必要がなく，データベースが小さくなるという利点もある．

3.3 検索クエリの作成

本研究では，ハミング歌唱による検索クエリを既知であるとしている．しかし，ユーザーが何も聴かずにテンポ一定で歌唱することは難しい．そこで，ユーザーはヘッドフォンから

表 2 ハミング楽曲検索精度実験条件
Table 2 Experimental conditions of the QbH experiment

データベース楽曲	108 曲 (**表 3** 参照)
楽曲サンプリング周波数	モノラル 44.1kHz
量子化ビット	16bit
F0 推定周波数帯域	3600-8400 cent(130-2090 Hz)
周波数シフト	20 cent (半音の 2/10)
入力クエリ	70 サンプル (表 3 参照)
被験者数	10 名 (男性 8 名，女性 2 名)A-J
歌唱範囲	4 小節程度

流れるテンポ一定のクリック音を聞き，それにあわせてハミング歌唱を行う．ハミング歌唱を行う場合，ユーザーが楽曲の休符の区間を飛ばして歌唱してしまう可能性があるが，このような手法により，ハミング入力音声のテンポも既知になり，ユーザーの歌唱入力の時間方向のゆらぎも減少すると考えられる．

4. ハミング楽曲検索精度実験

使用する F0 候補の個数が，単独の場合と複数候補を持つ場合とについて，実際にハミング歌唱入力を用いて，検索精度を比較し検証してみた．

データベース楽曲には，RWC 研究用音楽データベース[4] のポピュラー音楽 100 曲および，市販されているポピュラー音楽 8 曲を使用した．また入力する検索クエリとしては，同データベースの 26 曲を選択し，楽曲のサビの一部分のメロディーラインを 4 小節程度被験者がハミング歌唱し，その後テンポ正規化を行うことにより作成した．

実験結果を**図 5** に示す．この結果から，単独 F0 候補を用いた楽曲検索では 1 位検索率が 55.7% であるのに対し，5 個 F0 候補を用いた楽曲検索では 1 位検索率が 64.3%まで向上した．このことから，ハミング検索において F0 候補を複数個保持することが有効であることがわかる．

なお，検索精度が悪くなる原因の一つとして，F0 候補をすべてのフレームで一定個数選択していることが挙げられる．

例えば，**図 6**(a) の確率密度分布をとるフレームにおいては，第 1F0 候補の確率密度の値と，第 2 候補の値とでは拮抗しているため，どちらの F0 候補を使用すべきかは一概に判断できない．しかし，図 6(b) の分布の場合，第 1 候補の確率密度の値と第 2 候補の値とが大きく離れており，ほぼ第 1 候補を使用すべきであると考えられる．それにもかかわらず，第

表 3 使用楽曲
Table 3 The database used in the experiment

データベース	108 曲 RWC ポピュラー音楽データベース (RWC-MDB-P-2001) 100 曲 市販楽曲 8 曲
入力クエリ	70 サンプル 上記データベース内の 23 曲 (no.1-3,5-13,15-17, 19,20,28,32-33,40,47,48) 市販楽曲 3 曲
被験者数	10 名 (男性 8 名，女性 2 名)

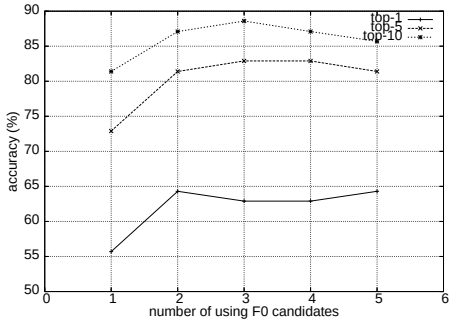


図 5 ハミング楽曲検索精度結果

2 候補だからといって検索に用いるとかえって検索精度に悪影響を及ぼしてしまう．

そこで，前述のように常に使用する F0 候補の個数を固定するのではなく，確率密度関数の値によって使用する F0 候補の個数を変化させる必要があると考えられる．

5. F0 確率密度の比に基づく F0 候補数選択

前項での考察から，第 1 候補の確率密度関数との比により F0 候補を選択するようなスコアリングを導入した．n 番目の F0 確率密度のピーク値を p_n とする．定数 $r(0 < r \leq 1)$ を定め，$p_k \geq rp_1$ となるピークを F0 候補として選ぶ．$r = 1.0$ のとき単独 F0 候補のみ使用する場合と同一条件となり，r の値を低くすることで使用する F0 候補の個数を増やすこととなる．

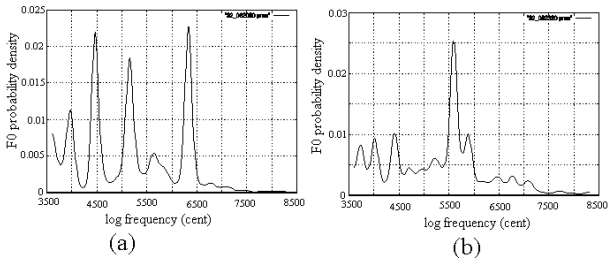


図 6 (a) 第 1 候補と第 2 候補が拮抗している場合 (b) 第 1 候補と第 2 候補が大きく離れている場合

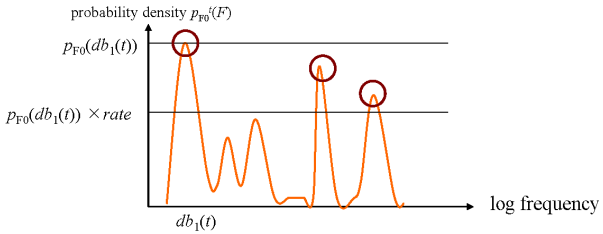


図 7 確率密度スコアリング

上記スコアリング関数を導入した場合の，結果を**図 8** に示す．1 位検索率については，$r = 0.6$ のとき最大となり，70.0%であった．．$r = 1.0$(F0 候補をひとつのみ使用することに相当) のときは 55.7%なので，10 ポイント以上の改善が実現された．また，前項で行った実験の使用する F0 候補の数を固定にした場合の結果 (F0 候補 5 個使用時：64.3%) よりも $r = 0.6$ のときの方が正解率が 5.7%向上している．

6. ま と め

本研究は，CD などの音楽音響信号しか存在しないような楽曲に対するハミング楽曲検索を行うためのシステムの実現を目指し，複数 F0 候補データベースを用いた楽曲検索システムについて検討した．F0 候補を複数個 (5 個) 利用することにより，1 位検索率を 55.7%から 64.3%まで改善することができた．また，スコアリング手法を修正することで，1 位検索率はさらに 70.0%まで改善した．

今後は，F0 候補抽出精度の向上を目指すとともに，さらなるスコアリング関数の検討も行う．また，検討する楽曲の追加も行っていく予定である．

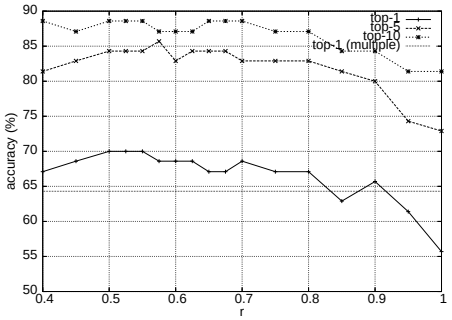


図 8 ハミング楽曲検索精度結果 (確率密度スコアリング)

参 考 文 献

1) 後藤真孝，齋藤毅，中野倫靖，藤原弘将，“歌声情報処理の最近の研究”，日本音響学会誌, Vol.64, No.10，pp.616-623, 2008.
2) M. Suzuki, T. Ichikawa, A. Ito and S. Makino, “Novel Tonal Feature and Statistical User Modeling for Query-by-Humming,” *J. of Information Processing*, vol. 17, pp.95-105, 2009.
3) M. Goto,“A Real-time Music Scene Description System: Predominant-F0 Estimation for Detecting Melody and Bass Lines in Real-world Audio Signals”, *Speech Communication*, Vol.43, No.4, 311-329，2004.
4) 後藤真孝, 橋口博樹, 西村拓一, 岡隆一,“RWC 研究用音楽データベース：研究目的で利用可能な著作権処理済み楽曲・楽器音データベース,” 情報処理学会論文誌, vol. 45, pp. 728–738, 2004.
5) 亀岡弘和, 西本卓也, 嵯峨山茂樹,“調波時間構造化クラスタリング (HTC) による音楽音響特徴量の同時推定”，情報処理学会研究報告，Vol.2005-MUS-61 pp71-78, 2005.
6) S.-P. Heo, M. Suzuki, A. Ito and S. Makino, “An Effective Music Information Retrieval Method Using Three-Dimensional Continuous DP,” IEEE Trans. Multimedia, vol.8, no.3, pp.633-639, 2006.