Vol.2010-MUS-85 No.9
2010/5/28

音程に注目した歌唱音声中の音符区間推定

鈴木基之[†1]　岡松竜徳[†1,*1]　任福継[†1]

本報告では，歌声を用いた音楽検索システムに用いるため，歌詞付きの歌唱音声から音符の区切り時刻を自動で検出する方法を提案する。従来から用いられてきたパワーの情報による区切り推定を行ったあと，すべてのフレーム間において音程を計算し，その変化する時刻を差分値のヒストグラムを用いて検出する。実際の歌唱音声に対して音符の区切り時刻の推定実験を行ったところ，従来のパワーによる方法での検出性能が F 値で 0.513 だったのに対し，提案する方法では 0.729 と，大幅に性能を向上させることができた。

On-set detection method of notes in singing voice based on tonal information

Motoyuki Suzuki,[†1] Tatsunori Okamatsu[†1,*1]
and Fuji Ren[†1]

It is desirable that an Music Information Retrieval (MIR) system accepts a singing voice with lyrics as a retrieval key. In order to use a front-end of a Query-by-Singing MIR system, a new on-set detection method has been proposed. After detecting on-set frames by using power information, tonal intervals are calculated for all combinations of frames, and on-set frames are detected by using partial differential coefficients and its histogram. Experimental results showed the proposed method gave higher performance than the power-based method. F-value of the proposed method was 0.729, and f-value of the power-based method was only 0.513.

†1 徳島大学大学院ソシオテクノサイエンス研究部
Institute of Technology and Science, The University of Tokushima
∗1 現在，郵便事業株式会社
Presently with Japan Post Service Co., Ltd.

1. はじめに

近年音楽圧縮技術が向上し，大量の音楽データを mp3 プレーヤーなどの小型デバイスに保存，再生することが可能となった。これらのデバイスは小型であるがゆえに，一般的にインターフェイスが貧弱である。その一方で保存可能な曲数は何千曲と膨大な数になっているため，保存してある曲の中から特定の 1 曲を指定して再生する，といった操作が非常にやり難い仕様となっている。

こうした問題を解決するため，歌声やハミングを用いて楽曲を検索するシステム（いわゆるコンテンツベースの検索システム）が開発されてきた。こうしたシステムにおいては，入力された歌唱音声から音符の情報（音の高さや長さ）を抽出する必要があるため，まずは音符ごとに区切る作業が必要となる。多くのシステム（例えば，1)–3) 等）では入力として「タタタ」等のハミングによる歌唱に限定することで，音符の区切り時刻をハミング音声のパワー情報から抽出する方式を採用している。

しかし一方で，こうした歌唱方法は自然ではないため，歌詞による歌唱音声を入力としたい，という要望も多い。そこでいくつかのシステムでは，歌詞による歌唱を入力可能とするよう，対処が行われている。歌唱音声からピッチを抽出し，その動き（変化する時刻）に注目して音符区切りを推定する方法[4)–6)] や，ピッチやその他の音響的特徴量を統計的にモデル化[7)–9)] する方法などが提案されているが，ピッチ推定自体が難しい問題であるため，それらを元にしても精度の高い性能は得られていない。また，音声認識システムと組みあわせることで，歌詞の認識結果と同時に音符の区切りを推定する方法[10),11)] も提案されているが，歌唱音声の認識は非常に難しく，その結果として音符区切りの推定結果もあまり精度が高いとはいえない。また，いくつかのシステム[12)–14)] では，「音符ごと」ではなく「フレーム」をベースとして検索を行うことで音符の区切り情報を必要としない検索方法を提案しているが，この方法では原理的に「音符の長さ」の情報を検索に用いることができない，という問題点がある。

本報告では，ピッチ情報に基づく音符の区切り時刻推定法を提案する。ピッチの抽出誤りに対して頑健に推定を行うため，ピッチの値そのものではなく，2 つのフレーム間のパワースペクトルの相互相関関数を求めることで，ピッチを抽出せずに直接音程を推定する方法[15)] を用いて，音程の変化から音符の区切り時刻を推定する。

Vol.2010-MUS-85 No.9
2010/5/28

2. 音程に注目した音符の区切り時刻の推定法

2.1 概　　要

ピッチ情報を用いた区切り時刻推定法では，ピッチの変化した時刻を区切り時刻として推定する。同様に，ある音符との音程を計算していけば，音程が変化する時刻を区切り時刻と見做すことができる。入力音声の中からある 1 つのフレームを基準フレームとして選択し，基準フレームと他のすべてのフレームとの間の音程を計算する。こうして得られた音程の時間波形から変化点を抽出し，音符の区切り時刻として出力する。

この方法では，最初に選択する基準フレームの性質によって推定性能が左右されることが容易に予測できる。すべてのフレームとの音程が正しく計算されれば高精度に区切り時刻を推定することが可能となるが，一方で無音や無声子音などに対応するフレームを基準フレームとしてしまった場合，正しく音程が計算できないために区切りの推定精度は劣化する。更に，慎重に基準フレームを選択したとしても，すべてのフレームとの音程が正しく推定可能である基準フレームが存在するとは限らず，あるフレームは前半のフレームとの音程が推定できず，別のフレームを選択すると，後半のフレームとの音程の推定精度が低下する，といった状況になることも考えられる。

そこで，すべてのフレームを基準フレームとして選択し，その他のフレームとの音程を計算する，ということを繰り返すことで頑健性を向上させる。この方法では，すべてのフレームの組み合わせの数（歌唱音声が N フレームからなるとすると，$\frac{N(N-1)}{2}$ 回）だけ音程推定を行うこととなるが，一部の組みあわせで音程推定精度が劣化したとしても，他の組み合わせによる推定結果から，頑健に音符の区切り時刻を推定することが可能となる。

図 1 に，すべてのフレーム間での音程推定結果のイメージ図を示す。この図において，x 軸，y 軸はともに時間を表し，ひとつの歌唱音声データを両方の軸に置く。$(x, y) = (i, j)$ である点は，i 番目のフレームと j 番目のフレームの間の音程を表し，色の濃さで表現している。直線 $y = x$ 上のすべての点については，同じフレーム間の音程であるので常に 0 [cent] であり，その他の音程についてもすべて正しく推定されたとすると，全体として市松模様となる。

このような音程推定結果から，音符の区切り時刻を推定する。図において区切り時刻は水平，または垂直の線となるため，以下のように計算する。

(1)　x 軸上の各フレームにおいて，それぞれ y 軸方向に差分フィルタをかけ，音程が変化する時刻を強調する

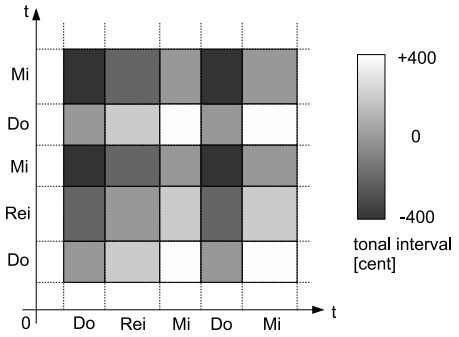

図 1　音程推定結果のイメージ図
Fig. 1　Virtual example of tonal intervals

(2)　y 軸上の各フレームにおいて，それぞれ x 軸方向に値を集計し，ヒストグラムを作成する。もしこのフレームが音符の中心付近であれば，音程に（y 軸方向へ）差分フィルタをかけた結果はほぼ 0 となるため，ヒストグラムの最頻値は 0 となる。一方，音符の区切り時刻付近のフレームにおいては，最頻値が 0 以外の（絶対値が大きな）値となる。そこで，こうした最頻値が 0 ではないフレームを音符の区切り時刻の候補フレームとして選択する。

この方法において，差分フィルタをかける時のフィルタ幅と，ヒストグラムをとる際の階級幅は，性能を見ながら最適なパラメータを設定する必要がある。

2.2 提案方法の詳細

本報告で提案する方法は，基本的に音程の変化情報から音符の区切り時刻を推定するものである。しかし，従来から用いられてきているパワーも区切りを推定する上で重要な情報を持つ。そこで，これらを組み合わせることで高精度な区切り時刻の推定を行う。

歌詞による歌唱を行った場合，ハミングによる歌唱とは異なり，すべての音符の開始時刻直前にパワーの小さい区間（ハミングにおける「タ」の破裂部）が存在するわけではない。しかし，歌詞による歌唱であっても，パワーが小さい区間（休符や息継ぎ，歌詞の中の子音の破裂部等）の直後は，音符の開始時刻に相当すると考えてよい。そこで，まずパワーの情報を用いて音符の区切り時刻を推定し，その後，パワーでは区切れなかった区間に対して音程を用いて推定を行う。

具体的なアルゴリズムは以下のとおりである。

(1) 入力された歌唱音声をフレームに分割し，各フレーム内の平均パワー $p(i)$ を計算する。ここで，i はフレーム番号を表す。

(2) しきい値 θ_p に対して，$p(k-1) < \theta_p$，$p(k) > \theta_p$ となるフレーム k を音符の開始時刻として出力する。
次以降のステップは，$p(i) > \theta_p$ であるフレームのみに対して行う。

(3) すべてのフレームの組み合わせに対して，パワースペクトルの相互相関関数に基づく方法[15] を用いて音程 $t(i,j)$ を計算する

(4) 各 i フレームに対して，差分フィルタを適用して差分値 $d(i,j)$ を式（1）で計算する。

$$d(i,j) = \frac{\sum_{k=-w_d}^{w_d} kt(i,j+k)}{\sum_{-w_d}^{w_d} k^2} \quad (1)$$

ここで，w_d は差分を経産する窓幅を制御するパラメータであり，実際の窓幅は $2\times w_d+1$ フレームとなる。

(5) 各 j フレームに対して，$d(0,j), d(1,j), \cdots d(N,j)$ のヒストグラムを作成する。階級幅は w_h とし，階級値が $0, \pm w_h, \pm 2w_h, \cdots$ となるように設定をする。この時の頻度を $h(j, c(m))$ とする。ここで $c(m)$ は m 番目の階級値である。

(6) j 番目のフレームに対応するヒストグラムの最頻値が 0 であれば 0，それ以外であれば 1 を出力する 2 値関数 $b(j)$ を作成する。

$$b(j) = \begin{cases} 0 & if \quad \underset{k}{\operatorname{argmax}}\, h(j,k) = 0 \\ 1 & otherwise \end{cases} \quad (2)$$

(7) 2 値関数 $b(j)$ を平滑化する。まず，前後のフレームが 0 である孤立した 1（$b(j-1) = b(j+1) = 0, b(j) = 1$）をすべて 0 へと変更し，その後，孤立した 0（$b(j-1) = b(j+1) = 1, b(j) = 0$）を 1 へと変更する。

(8) 2 値関数 $b(j)$ において，連続して 1 である区間の中心フレームを音符の区切り時刻として出力する。

表 1 実験条件
Table 1 Experimental conditions

データベース	使用した曲	日本の童謡 48 曲
	歌唱者	男性 19 名，女性 8 名
	データ数	202 データ
音響分析条件	サンプリング周波数	16 kHz
	フレーム幅	64 ms ハニング窓
	フレーム周期	50 ms
パラメータの設定	θ_p	-5 dB
	w_d	1, 2
	w_h	50, 100 ($w_d = 1$) 30, 60 ($w_d = 2$)
評価方法	指標	再現率，適合率，F 値
	正解の定義	前後 ±1 フレームまでのずれを許容

3. 音符区切り時刻の推定実験

3.1 実 験 条 件

実験条件を，表 1 に示す。本実験では，独自に収集した歌唱データベースの音声を用いた。男性 19 名，女性 8 名に日本の童謡 48 曲の中から数曲を歌唱してもらい，全部で 202 データを実験に用いた。各データに含まれる平均の音符数はおよそ 26 音符であった。

差分フィルタの計算に用いる窓幅として 3 フレームと 5 フレームを実験した。ヒストグラムの階級幅は，「半音」と「全音」に対応するように設定した。連続する 2 つの音符の音程が半音である場合，各フレームの音程系列は 0, 0, 0, 100, 100, 100 [cent] のようになる。差分フィルタの窓幅を 3 フレームとすると，差分値は 50 となる。窓幅が 5 フレームの時の差分値は 30 となる。そこで，階級幅を「半音」に設定する，とは，差分フィルタの窓幅が 3 の場合は 50，5 の場合は 30 に設定した，ということである。同様に「全音」の設定は，100 と 60 となる。

3.2 音符区切りの抽出精度

まず最初に，従来からのパワーによる音符区切りの抽出実験を行ったところ，再現率が 37.5%，適合率が 91.4%となった。これを見ると，再現率が非常に低く，パワーだけでは歌唱音声の音符区切りを高精度に推定できないことがわかる。一方，適合率は 90%を越えており，パワーで抽出した音符区切りの時刻はほぼ正解であることがわかる。つまり，2.2 節で説明したように，歌詞による歌唱音声であってもパワーの小さい区間の直後には音符の開始時刻があると考えられるため，まずはパワーの情報から区切り時刻を求め，その後，区切

Vol.2010-MUS-85 No.9
2010/5/28

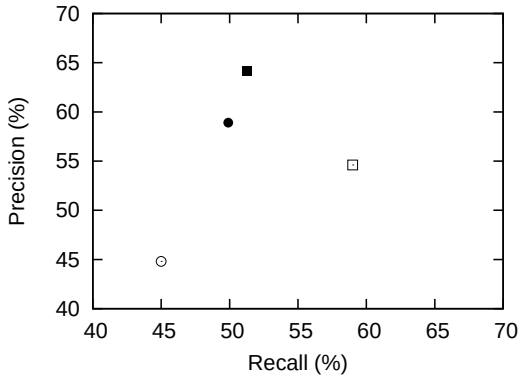

図 2 パワーが高いフレームに対する推定精度
Fig. 2 On-set detection performance for high power frames

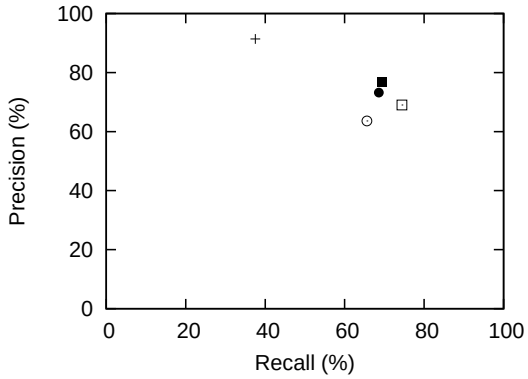

図 3 すべてのフレームに対する推定精度
Fig. 3 Total on-set detection performance

れなかった区間に対して音程の情報から区切り時刻を推定する，という方法が妥当であることが示された。

次に，パワーでは区切れなかった（しきい値以上のパワーを持つ）フレームに対する，提案方法による音符区切り時刻の推定精度を図 2 に示す。図中の 4 つの記号は，それぞれ差分フィルタの窓幅とヒストグラムの階級幅を変えた時の性能を示している。四角い記号は窓幅が 3 フレームの，丸い記号は 5 フレームの時の結果である。また，白い記号はヒストグラムの階級幅を「半音」に，黒く塗り潰してある記号は「全音」に設定をした時の結果である。

この図を見ると，差分フィルタの窓幅は 5 よりも 3 がよいことがわかる。今回の実験では，フレーム周期は 50 [ms] と設定したため，窓幅を 5 とすると 250 [ms] の区間から差分値を計算したことになる。これは短い音符の長さを越えてしまうため，性能が劣化したものと考えられる。もちろんフレーム周期を 50 [ms] より短くすれば，これらの結果も変化していくものと思われる。

差分フィルタの窓幅を 3 に設定した場合，ヒストグラムの階級幅は，どちらに設定しても同程度の結果となった。階級幅を「全音」とした場合は，原理的に半音の差しかない 2 つの音符を区切ることは不可能となる。しかし一方で「半音」と設定すると，差分値の小さなゆらぎに過剰に敏感になってしまう。そのため,「半音」に設定すると再現率は上がるが同時に適合率が下がってしまい，結果として両者に大きな差はみられない，という結果になったと思われる。

最後に，パワーが小さいフレームも含めた入力音声全体に対する推定精度を図 3 に示す。図中の十字記号は，パワーのみによる推定結果を示している。この結果から，従来のパワーのみによる方法と比較して，再現率が大幅に改善されていることがわかる。一方で適合率は多少の低下となってしまった。両者の F 値は 0.532，0.729 となり，提案方法の方が大幅に性能を向上させていることがわかった。

3.3 考　　察

3.3.1 短い音符に対する性能

前節で得られた結果を分析した結果，短い音符を正しく区切ることができていない例が多いことがわかった。図 4 に，短い音符が連続するデータに対する推定結果の例を示す。この図において，赤い折れ線グラフは提案方法が出力した 2 値関数 $b(j)$ の値を，緑の線は正解の音符区切り時刻を示している。これを見ると，図の右側においていくつかの音符があるが，それらの音価が短いために，提案方法では区切りの推定に失敗している。

この実験においてはフレーム周期を 50 [ms] としているため，差分フィルタは窓幅を 3 フレームとしても 150 [ms] から計算していることになる。今回用いたデータの中で，一番短い音符の持続時間は 125 [ms] しかなかったため，正しく区切ることができなかったと思われる。

そこで，同じデータに対してフレーム周期を 10 [ms] に変更して実験を行った。その時の結果（図 4 の右側のみ）を図 5 に示す。この図をみると，短い音符に対してもいくつかは正

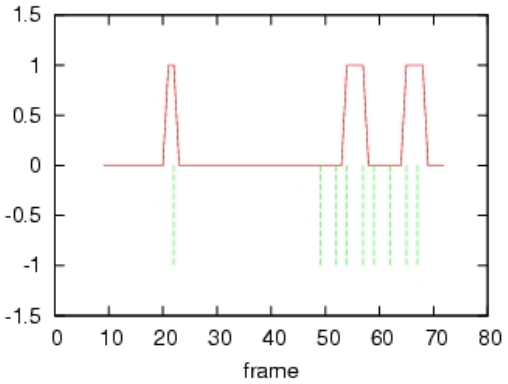

図 4 短い音符に対する推定結果の例（フレーム周期 50 ms）
Fig. 4 On-set detection results for short length notes (frame shift: 50 [ms])

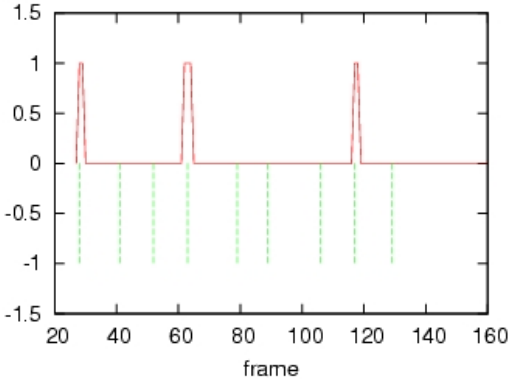

図 5 短い音符に対する推定結果の例（フレーム周期 10 ms）
Fig. 5 On-set detection results for short length notes (frame shift: 10 [ms])

しく区切りを推定することができたことがわかる。しかし一方で，フレーム周期を短くするとフレーム数が増えるため，音程計算の回数が 2 乗で増加してしまう，という問題がある。そのため実用化する際には，計算時間と短い音符に対する推定精度の双方を考慮した上でフレーム周期を決定する必要がある。

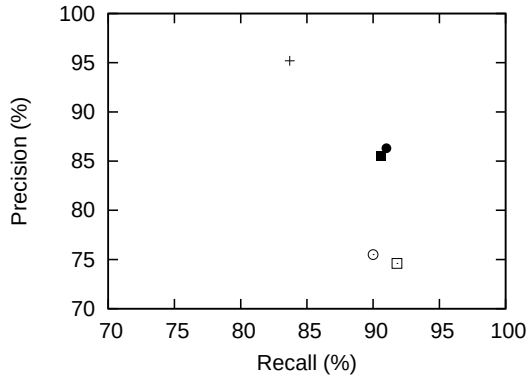

図 6 ハミングに対する音符区切り時刻の推定精度
Fig. 6 On-set detection performance for humming voice

3.3.2 ハミングに対する性能評価

本報告で提案した方法は歌詞による歌唱音声を扱うことが可能である。この方法に対して，従来のシステムへの入力として用いられてきたハミングを入力した場合について考察を行った。

提案方法では，まずパワーによる区切り推定を行う。ハミングが入力された場合，その区切りのほとんどはパワーを用いることで推定可能であるため，後段の音程による区切り時刻の推定ステップでは何も出力しない，というのが望まれる挙動である。しかし実際には誤認識によっていくつかの区切り時刻が出力されることが考えられ，パワーのみによる従来の方法に比べて性能が劣化することが予想される。そこで実際にハミング入力に対する性能評価を行った。

用いた入力音声は，前節の実験と同じ歌唱者（男性 19 名，女性 8 名）によるハミング音声 202 データである。その他の実験条件についても，前節の実験と同じとした。

図 6 にハミングに対する推定精度を示す。この図を見ると，パワーのみによる従来の方法は，ハミング入力に対しては十分な性能を示すことがわかる。一方提案方法は，再現率は従来法に比べて高いが，適合率が劣化してしまう，という結果となった。これは，音程による区切り推定によって数多くの（誤った）区切り時刻を出力したためと思われる。

この結果に対して F 値を計算してみる（表 2）と，パワーのみによる方法と提案方法で，ほぼ同等の性能を出していることがわかる。このことから，提案方法は歌詞による歌唱音声

表 2 ハミングに対する性能の F 値
Table 2 F-value for humming voice

方法	w_d	w_h	F 値
パワーによる方法	—		0.892
提案方法	1	半音	0.823
		全音	0.880
	2	半音	0.820
		全音	0.886

だけではなく，ハミングに対しても十分な精度を持って利用可能であることがわかった。

4. ま と め

歌詞による歌唱音声を入力とする楽曲検索システムに用いるため，音程の変化に注目した高精度な音符区切り時刻の推定方法を提案した。まず従来の方法と同様にパワーの情報を用いて音符区切りを推定し，その後区切れなかった各区間に対して音程情報を用いて区切り時刻を推定していく。音声をフレームに分割した後ですべてのフレームの組み合わせに対して音程を計算し，その変化時刻を差分フィルタとヒストグラムの最頻値を用いることで推定する。実際に歌唱音声に対して音符区切りの推定実験を行ったところ，従来のパワーによる方法が F 値で 0.513 であったのに対して 0.729 と大幅に性能が向上した。またハミング入力に対しても従来の方法と同等の性能を示し，どちらの入力に対しても利用可能であることがわかった。

本報告で提案した方法は音程の変化に注目した方法であるため，原理的に音程が 0 [cent] である 2 つの音符の区切りを推定することはできない。この問題を解決するためには，MFCC や歌詞の認識結果といった別の情報を併用する必要がある。今後はこうした情報も利用した方法を開発していく予定である。

参 考 文 献

1) Kosugi, N., Nishihara, Y., Sakata, T., Yamamoto, M. and Kushima, K.: A Practical Query-By-Humming System for a Large Music Database, *ACM Multimedia 2000*, pp.333–342 (2000).
2) Ghias, A., Logan, J., Chamberlin, D. and Smith, B.C.: Query By Humming: Musical Information Retrieval in An Audio Database, *Proc. ACM Multimedia*, pp. 231–236 (1995).
3) Liu, B., Wu, Y. and Li, Y.: A Linear Hidden Markov Model for Music Information Retrieval Based on Humming, *Proc. ICASSP 2003*, Vol.V, pp.533–536 (2003).
4) Pollastri, E.: Some Considerations About Processing Singing Voice for Music Retrieval, *Proc. ISMIR*, pp.285–286 (2002).
5) Klapuri, A.P., Eronen, A.J. and Astola, J.T.: Analysis of the Meter of Acoustic Musical Signals, *IEEE Trans. Speech, Audio, and Language Processing*, Vol.14, No.1, pp.342–355 (2006).
6) Birmingham, W., Pardo, B., Meek, C. and Shifrin, J.: The MusArt Music-Retrieval System: An Overview, *D-Lib Magazine*, Vol.8, No.2 (2002).
7) Meek, C.J. and Birmingham, W.P.: A Comprehensive Trainable Error Model for Sung Music Queries, *Journal of Artificial Intelligence Research*, Vol.22, pp.57–91 (2004).
8) Raphael, C.: A Graphical Model for Recognizing Sung Melodies, *Proc. ISMIR*, pp. 658–663 (2005).
9) Toh, C.C., Zhang, B. and Wang, Y.: MULTIPLE-FEATURE FUSION BASED ON ONSET DETECTION FOR SOLO SINGING VOICE, *Proc. ISMIR*, pp.515–520 (2008).
10) Suzuki, M., Hosoya, T., Ito, A. and Makino, S.: Music Information Retrieval from a Singing Voice Using Lyrics and Melody Information, *EURASIP Journal on Advances in Signal Processing*, Vol.2007, pp.Article ID 38727, 8 pages (2007). doi:10.1155/2007/38727.
11) Suzuki, M., Hosoya, T., Ito, A. and Makino, S.: Music Information Retrieval from a Singing Voice Based on Verification of Recognized Hypotheses, *Proc. ISMIR*, pp. 168–171 (2006).
12) Jang, J.R., Hsu, C. and Lee, H.: CONTINUOUS HMM AND ITS ENHANCEMENT FOR SINGING/HUMMING QUERY RETRIEVAL, *Proc. ISMIR*, pp.546–551 (2005).
13) Hu, N. and Dannenberg, R.B.: A Comparison of Melodic Database Retrieval Techniques Using Sung Queries, *Joint Conference on Digital Libraries*, Association for Computing Machinery, pp.301–307 (2002).
14) Mellody, M., Bartsch, M.A. and Wakefield, G.H.: Analysis of Vowels in Sung Queries for a Music Information Retrieval System, *Journal of Intelligent Information Systems*, Vol.21, No.1, pp.35–52 (2003).
15) Suzuki, M., Ichikawa, T., Ito, A. and Makino, S.: Novel Tonal Feature and Statistical User Modeling for Query-by-Humming, *IPSJ Journal*, Vol.50, No.3, pp. 1100–1110 (2009).