論　文

打楽器音を対象にした音源分離システム

准　員　後藤　真孝†　　正　員　村岡　洋一†

A Sound Source Separation System for Percussion Instruments

Masataka GOTO †, *Associate Member and* Yoichi MURAOKA †, *Member*

あらまし　本論文では，複数の打楽器のみで演奏された音楽から各打楽器音の発音時刻と強度を認識する音源分離システムについて述べる．音源分離とは，複数の音源の音が混在している音響信号からそれぞれの音を分離して認識する技術であり，曲の音響信号からその楽譜を作り出す自動採譜において，中心となる重要な処理である．従来，楽音を対象にした音源分離システムは研究されてきたが，それらの手法は打楽器音に対して適用することができない．そこで本論文では，打楽器音の音源分離を実現する認識手法を提案する．本手法では，事前に登録してある打楽器音のテンプレートパターンと入力パターンとの距離を，改良したテンプレートマッチングにより求めてしきい値処理する．我々は，音量補正法，音源分離を実現する距離尺度，選択的注意の機構の三つの点でテンプレートマッチングを改良した．これにより，複数の音が混在したり音量が変化した場合にも各打楽器音を認識できる．本システムをワークステーション上に実装し，打楽器音の音源として電子楽器を用いて実験した結果，8 Beat のドラムパターンの演奏音を音源分離することができた．

キーワード　音源分離，テンプレートマッチング，自動採譜，打楽器，音認識

1. まえがき

近年計算機の発達に伴い，音楽と計算機を相互に結び付けた研究が多くなされてきた．これらは音楽情報処理という研究領域を確立し，現在も活発な研究がなされている．その1分野として，音源分離や自動採譜に関する研究が数多く行われてきた．音源分離は，複数の音源の音が混在している音響信号から，それぞれの音を分離して認識する技術であり，複数の楽器で演奏された音楽を計算機が扱う際に重要な役割を演じる．同時に，音源分離は人間の聴覚機能のうちの音の知覚過程を計算機で実現する技術であり，人間の音楽認識過程の初期段階に相当する．

従来の音源分離・自動採譜の研究は，単音の演奏から和音の演奏へ，単一楽器の演奏から複数楽器の演奏へと，対象にする音楽の制約を徐々に減らす方向で進められてきた[(1)~(9)]．しかし，これらの研究はピアノなどの楽音*を発する楽器のみを対象にしたものであり，打楽器などの噪音**を発する楽器は対象にしていなかった．

本研究では，打楽器音と楽音がともに含まれている音楽を音源分離・自動採譜するための第1段階として，打楽器のみで演奏された音楽を対象にした音源分離システムを実現した．ポップスやロックのような打楽器音が含まれている音楽では，楽音だけでなく打楽器音が重要な役割を演じている．しかし，打楽器音の多くが楽音とは性質の異なる噪音であるために，従来の楽音を対象にした音源分離の手法では打楽器音を扱うことができない．更に従来の手法をこれらの音楽に適用すると，打楽器音がノイズとなって楽音の認識を妨げ，打楽器音だけでなく楽音の認識すら困難になる．従って，これらの音楽を音源分離するためには打楽器音に対応した音源分離を実現することが重要である．

本システムは，複数の打楽器音が混合したモノラルの音響信号を入力とし，打楽器の種類とその発音時刻・強度を表すシンボルを標準 MIDI ファイルの形式で出力する．但し，個々の打楽器音は事前にシステムに登

† 早稲田大学大学院理工学研究科，東京都
Graduate School of Science and Engineering, Waseda University, Tokyo, 169 Japan

* 基音とその整数倍の倍音をもち，音の高さが明確で音階や協和音を構成できる音．調音とも呼ばれる．

** 周波数成分の関係が整数比例関係でなく，音の高さが不明確で音階や協和音を作りがたい音．非楽音とも呼ばれる．

録しておくものとする．

打楽器音の音源分離を実現するには次のような問題点がある．打楽器音は楽音とは性質が大きく異なり，複雑なスペクトル構造をもつために基音と倍音が明確に分けられず，音階上に周波数同定できない．そのため，従来の基音を中心に認識を進めていく楽音を対象にした手法では，打楽器音を音源分離することができない．また，単独で鳴っている打楽器音を認識するのは比較的容易だが，複数の打楽器音が同時に鳴っている場合には，各音の周波数成分がお互いに重なり合うために認識が困難になる．

以上の問題を解決するために，本論文では音源分離を実現する新たな認識手法を提案する．本手法は，音量補正法，音源分離を実現する距離尺度，選択的注意の機構の三つの点でテンプレートマッチングを改良した手法である．これにより，各音の周波数成分が重なり合ったり音量が変化した場合にも各打楽器音を認識できる．

以下，**2.**においてシステムの仕様と処理手順について述べる．そして**3.**で周波数解析の方法について，**4.**で本システムの処理の中心となる楽器同定・音源分離の方法について述べる．**5.**では実装した本システムによる実験結果を示し，その考察を行う．最後に**6.**で結論と今後の課題を述べる．

2. 打楽器音を対象にした音源分離システム

2.1 システムの仕様

本システムの入力信号は，ポップスやロックのような音楽で多く使用されるドラムスの演奏音とし，楽音は含まれていないものとする．典型的なドラムスの配置を図1に示す．このドラムスは，Bass Drum (BD)，Snare Drum (SD)，Low Tom (LT)，Middle Tom (MT)，High Tom (HT)，Hihat Close (HC)，Hihat Open (HO)，RIde cymbal (RI)，CRash cymbal (CR) の9種類の打楽器で構成され，個々の音は事前に計算機に登録しておく．このうちHCとHOは同一の打楽器であるが，演奏音が異なるために分けてある．またHOはその性質上，鳴り始めと鳴り終わりの両方を検出するのが望ましいので，消音時刻も認識する．

以上9種類の打楽器は，Erich von Hornbostel と Curt Sachs の体系的楽器分類によると[10]，膜鳴楽器†（BD，SD，LT，MT，HT）と体鳴楽器††（HC，HO，RI，CR）の二つに分けられる．前者の五つの音は1kHz以下の中低域の周波数成分が多く，後者の四つの音はすべてシンバル系の音なので，4kHz以上の高域の周波数成分が多い．但し，SDは高域の周波数成分も多く含んでいる．

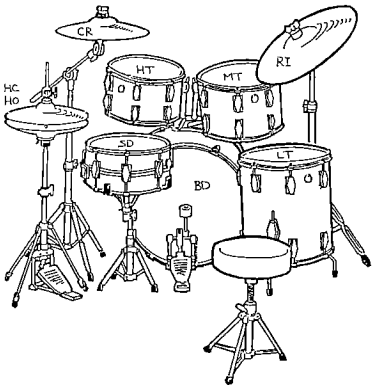


図1 典型的なドラムスの配置
Fig. 1 Normal formation of drums.

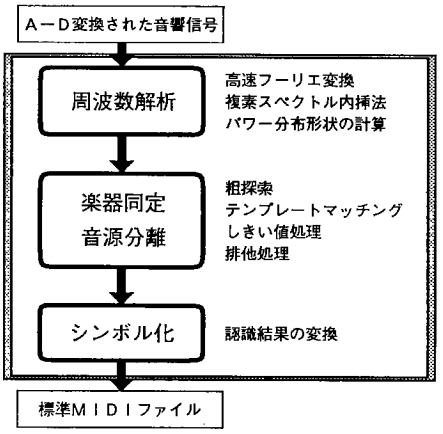


図2 処理手順
Fig. 2 The process flow chart.

音源分離の結果として出力するシンボルには標準MIDIファイル(SMF)を用いる．SMFは現在広く使用されている演奏データのフォーマットである．そこで本システムが出力したSMFを用いて，打楽器音を出力する電子楽器を自動演奏することにより，音源分離の結果を音で確認することができる．

2.2 処理手順

本システムの処理手順を図2に示す．各処理について概説する．

† 膜状のものの振動が基本になっている楽器．大部分が太鼓であり，打面に張られた皮の振動によって音がつくり出される．
†† 一定の容積をもった固体自身の振動が基本になっている楽器．

（1） 周波数解析 (3.参照)

A-D 変換された音響信号に対し，各時刻において存在する周波数成分の周波数とその強度を求める．この結果から，膜鳴楽器と体鳴楽器のそれぞれの認識に用いる2種類のパワー分布形状を得る．パワー分布形状については **3.2** で説明する．

（2） 楽器同定・音源分離 (4.参照)

打楽器音の音源分離を実現する認識手法により，打楽器の種類とその発音時刻・強度を認識する (楽器同定)．また複数の打楽器音が同時に鳴っている場合には，各打楽器音を分離して認識する (音源分離)．

（3） シンボル化

楽器同定・音源分離の結果を SMF の形式に変換する．

3. 周波数解析

周波数解析では

（1） 高速フーリエ変換 (FFT)

（2） 複素スペクトル内挿法 (ハニング窓対応補正法)

（3） パワー分布形状の計算

の順で処理を行う．

まず，サンプリングされた入力信号に対し，観測区間を時間軸方向にずらしながら高速フーリエ変換 (FFT) を行う．FFT で求めた周波数スペクトルは複素数であるため，複素スペクトルと呼ぶ．

FFT には Sande-Tukey の周波数間引き型 FFT を用い，FFT の窓関数にはハニング窓を使用する．ハニング窓は観測区間の始めと終わりの接続を滑らかにし，スペクトルの漏えい (広がり) を減少させる効果をもつ．また方形窓やハミング窓と比較して，ハニング窓は位相の保存性が良い[11]．

3.1 複素スペクトル内挿法

FFT だけで周波数分解能を上げるためには観測区間を長くしなければならず，音符の速い変化についていけなくなる．そこで，観測区間を長くせずに周波数分解能を上げるための方法として，複素スペクトルの位相特性を利用して内挿する複素スペクトル内挿法を用いる†．

文献(12)で提案されている従来の複素スペクトル内挿法は，方形窓を使用した FFT の結果を対象とするためハニング窓には対応していない．そのため，ハニング窓を使用した FFT による複素スペクトルに対して，従来の複素スペクトル内挿法をそのまま適用すると，求めた周波数，振幅に誤差を生じる．そこでこの誤差をなくすために，従来の複素スペクトル内挿法で求めた結果をハニング窓に対応するように補正する方法を提案する．

3.1.1 従来の複素スペクトル内挿法

従来の複素スペクトル内挿法は，区間周波数†† f の成分が存在するピークの前後にある複素スペクトル z_m, z_{m+1} ($m=[f]$†††) から，区間周波数 f，振幅 a を推定する方法である．但し，複素スペクトルは方形窓を使用した FFT により求めたものとする．複素スペクトルの各ピーク††††に対して本方法を適用することにより，周波数成分を精度良く求めることができる．

単位ベクトル

$$u=\frac{(z_{m+1}-z_m)}{|z_{m+1}-z_m|} \tag{1}$$

を定義し，u と z_m，u と z_{m+1} の内積 $(\cdot,\cdot)$ を用いて次式により区間周波数 f，振幅 a を推定する．

$$f=m+\frac{(u,z_{m+1})}{(u,z_{m+1})-(u,z_m)} \tag{2}$$

$$a=\frac{\pi(f-m)(u,z_m)}{\sin(\pi f)} \tag{3}$$

3.1.2 ハニング窓対応補正法

ハニング窓を使用した FFT による複素スペクトルに対して **3.1.1** の方法を適用すると，得られた区間周波数 f，振幅 a に誤差を生じる．これは，式(2)，(3)が方形窓を使用した FFT を前提に導かれたものだからである．

そこで次式により，従来の複素スペクトル内挿法の結果をハニング窓に対応するように補正する．

$$z=3(f-m)-1 \tag{4}$$

$$F=m+z \tag{5}$$

$$A=6z(z-1)\frac{\sin\pi(f-m)}{\sin\pi z}a \tag{6}$$

これは，ハニング窓を使用した FFT の結果から従来の複素スペクトル内挿法で求めた区間周波数を f，振幅を a，$m=[f]$ としたときに，補正後の区間周波数 F と振幅 A を求める式である．これらの補正式の導出過程を付録に示す．

---

† 周波数分解能の向上は時間領域のオーバサンプリングによっても実現できるが，ここではオーバサンプリングに比べ処理量が抑えられる複素スペクトル内挿法を用いた．

†† 観測区間に含まれる波数のことで，波長が観測区間の長さである正弦波は区間周波数1となる．

††† $[\cdot]$はガウス記号を表す．

†††† 複素スペクトルのピークは，パワーの極大値の中で位相が前後で反転しているものを選択して求めた．

3.2 パワー分布形状

パワー分布形状とは，複素スペクトル内挿法で得られた周波数成分のパワーが，時間周波数平面上にどのように分布しているかを表すものである．つまり，周波数成分の周波数軸方向のパワー分布が時間変化する形状のことである．このパワー分布形状を使用して，4.で述べるテンプレートマッチングを行う．

本システムは，膜鳴楽器と体鳴楽器のそれぞれの認識のために，周波数軸のとり方が異なる2種類のパワー分布形状を使用する．一方のパワー分布形状[cent]は，中低域の分布をよく表すように周波数軸を対数尺度のcent†で示し，もう一方のパワー分布形状[Hz]は，高域の分布をよく表すように周波数軸を通常のHzで示す．膜鳴楽器の音は中低域の周波数成分が中心なので，その認識にはパワー分布形状[cent]を用い，体鳴楽器の音は高域の周波数成分が中心なので，パワー分布形状[Hz]を用いる．

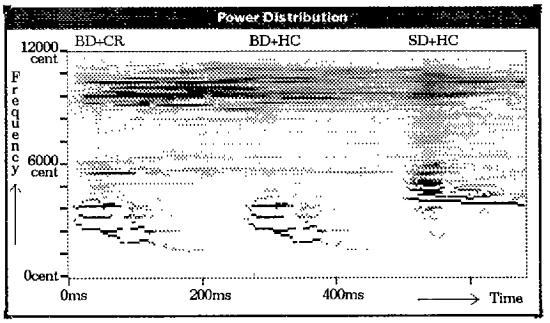
パワー分布形状は，複素スペクトル内挿法で得られた各時刻の周波数成分に対し，以下の処理を行うことにより求める．

（1） 一定の周波数 f_c ごとに周波数軸を区切る．各打楽器音の音響的性質をパワー分布形状が表すことができるように f_c を設定する．但し，パワー分布形状[cent]の場合には f_c をcentで表し，パワー分布形状[Hz]の場合には f_c をHzで表す．

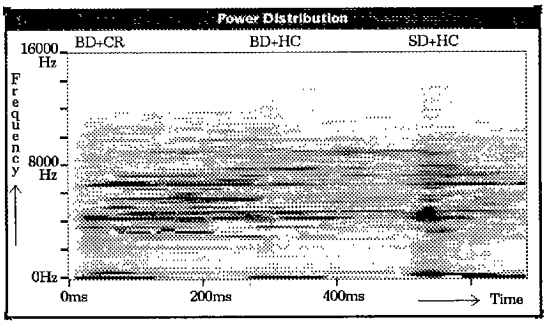
（2） 区切られた各区間ごとに，その区間に含まれる周波数成分のパワーの最大値を求める．この最大値を各区間の代表値とし，代表値の分布を周波数軸方向のパワー分布とする．

（3） 以上の処理を各時刻に対して行うことにより，周波数成分のパワー分布の時間変化を求める．

同一の音響信号から得られた2種類のパワー分布形状の例を図3に示す．



(a) Power distribution [cent]



(b) Power distribution [Hz]

図3 パワー分布形状の例
Fig. 3 Examples of power distribution.

4. 楽器同定・音源分離

打楽器音の音源分離を実現する認識手法を提案する．まず必要とされる機能について検討した後，それらの機能を実現する認識手法を提案する．本手法では，まず粗探索により打楽器音の発音時刻の候補を求める．次に，テンプレートパターンと入力パターンとの距離を，改良したテンプレートマッチングにより求めてしきい値処理する．そして，HCとHOの認識結果に対して同一楽器制約による排他処理を行う．

4.1 打楽器音の認識手法に必要とされる機能

打楽器音を対象にした音源分離システムを実現するためには，認識手法は以下のような機能をもつ必要がある．これらの機能の具体的な実現方法は，（ ）内に示した節で述べる．

（1） 各打楽器音の発音時刻を認識

打楽器の種類と共にその発音時刻を認識する．HOの場合には消音時刻も認識する（**4.2**,**4.3**,**4.5**）．

（2） 複数の打楽器音が同時に鳴った場合に対応

複数の打楽器音の周波数成分がお互いに重なり合った場合でも，それぞれの打楽器の種類とその発音時刻・強度を認識する（**4.3**）．

（3） さまざまな音量の打楽器音に対応

事前に登録した打楽器音の音量とは異なる音量の入力に対しても，打楽器の種類とその発音強度を認識する（**4.3.1**）．

† 民族音楽学者のエリスによって提唱された音程の表示法．同一周波数を表す f Hzと c centの関係は，基準周波数を f_0 とすると $c=1200\log_2 f/f_0$ となる．本論文では，$f_0=440\,\mathrm{Hz}\times 2^{3/12-5}\fallingdotseq 16.35\,\mathrm{Hz}$ とする．

（4） 周波数成分の小さな変動を吸収

事前に登録した打楽器音の周波数成分と入力の周波数成分がわずかに(式(12)，(14)の Ψ の範囲内で)異なる場合にも，打楽器の種類を認識する(**4.3.2**)．

（5） 同一楽器制約による排他処理

同一打楽器の異なる演奏音であるHCとHOに対して，同時に鳴っていると認識しないように排他処理をする(**4.5**)．

4.2 粗探索

粗探索により低い音と高い音の各発音時刻の候補を求めてから，その時刻に鳴り始めた音に対してテンプレートマッチングを行う．これは，すべての時刻の音に対してテンプレートマッチングを行う詳細探索よりも計算効率が良い．粗探索を行わない場合には，本来発音していない時刻の音に対してもテンプレートマッチングを行うために，発音していると誤認識する可能性がある．粗探索によりこの誤認識を回避できる．

粗探索の実現方法を述べる．発音時刻候補の決定は，パワー分布形状の時間軸方向の1次微分が大きい値をとる時間周波数平面上の座標を求めることにより行う．この座標が低域に密集している時刻を低い音の発音時刻候補とし，高域に密集している時刻を高い音の発音時刻候補とする．低い音の発音時刻候補はパワー分布形状[cent]から求め，高い音の発音時刻候補はパワー分布形状[Hz]から求める．

具体的な処理手順を以下に示す．但し，パワー分布形状[k]の時刻 t，周波数 f におけるパワーを $P_k(t,f)$ とする．

（1） 立上りの度合 $Q_k(t,f)$ を求める．$t=l-1$，l，$l+1$ の各時刻において連続して

$$\frac{\partial P_k(t,f)}{\partial t}>0 \qquad (7)$$

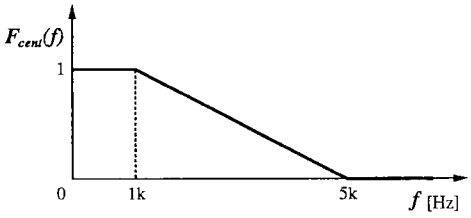
を満たすとき，$t=l$ における $\partial P_k(t,f)/\partial t$ の値を $Q_k(l,f)$ とする．満たさないときは $Q_k(l,f)=0$ とする．

（2） 各時刻 t ごとに $Q_k(t,f)$ の重み付き合計値 $S_k(t)$ を式(8)により求める．

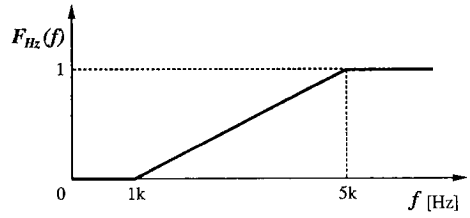
$$S_k(t)=\sum_f F_k(f)Q_k(t,f) \qquad (8)$$

但し，$F_{cent}(f)$ は図4(a)のように低域フィルタの特性に定め，$F_{Hz}(f)$ は図4(b)のように高域フィルタの特性に定める．

（3） $S_k(t)$ に対し，SavitzkyとGolayの2次多項式適合による平滑化微分を用いたピーク検出[13]を行い，極大値(ピーク点)を与える時刻を検出する．こうして，



(a) $F_{cent}(f)$ (Low pass filter)



(b) $F_{Hz}(f)$ (High pass filter)

図4 $F_k(f)$ の定義
Fig. 4 Definition of function $F_k(f)$.

$S_{cent}(t)$ から求めた時刻を低い音の発音時刻候補とし，$S_{Hz}(t)$ から求めた時刻を高い音の発音時刻候補とする．

4.3 テンプレートマッチング

テンプレートパターンと入力パターンとの距離を，テンプレートマッチングにより求める．テンプレートパターンは，事前に登録した打楽器音のパワー分布形状とする．一方入力パターンは，粗探索で求めた各発音時刻候補から始まる入力音のパワー分布形状の一部分とし，その時間長がテンプレートパターンと同じになるようにする．膜鳴楽器を認識する際には，テンプレートパターンと入力パターンの両者をパワー分布形状[cent]で表し，体鳴楽器を認識する際には，両者をパワー分布形状[Hz]で表して距離を求める．

この両者の距離を求めるのに，ユークリッド距離を距離尺度とする単純なテンプレートマッチングを用いると，次の三つの問題が生じる．

（1） テンプレートパターンと入力パターンの音量が異なると，両者が同じ音色でも距離が大きくなる．

（2） 複数の打楽器音が同時に鳴ると距離が大きくなる．

（3） 認識するのに関係のない周波数成分が異なっても距離が大きくなる．

そこで以上の問題点を解決するために，テンプレートマッチングを以下のように改良する．

4.3.1 音量補正法

距離を求める前処理として，正しい距離が得られる

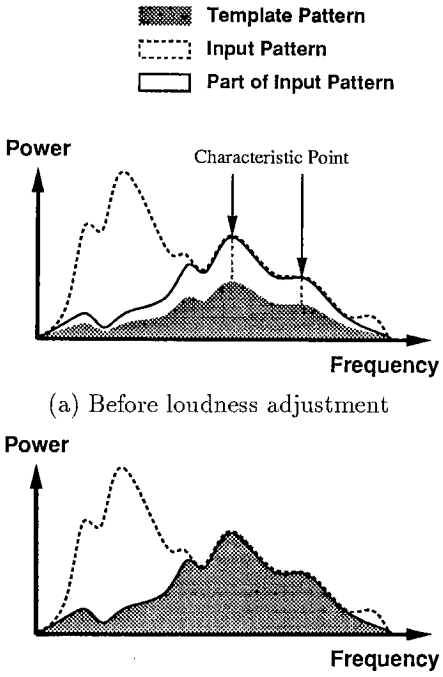


図5　音量補正法
Fig. 5　An example of loudness adjustment.

ように両者の音量を合わせる．これは単に両者のパワーの最大値を同じにするのではない．図5(a)の点線のように入力パターン中に他の打楽器音の成分がある場合には，その部分入力パターンとテンプレートパターンの音量を合わせる．部分入力パターンとは，入力パターン中にテンプレートパターンが含まれていると仮定したときに，入力パターン中のテンプレートパターンに相当する部分のことであり，図5(a)では実線で示してある．

音色が異なればすべての周波数成分が一致することはないので，図5(a)の矢印のような各楽器固有の特徴点から音量補正値を求め，図5(b)のようにテンプレートパターンの音量を変化させる．この各楽器固有の特徴点は，後述する重み関数により与えられる．また，打楽器の発音強度はこの音量補正値から求める．

音量補正値の具体的な求め方を述べる．

(1)　各時刻 t における音量補正値 $\delta_n(t)$ [dB]を求める．打楽器の種類を n，時刻 t，周波数 f における重み関数の値を $w_n(t,f)$，入力パターンのパワーを $I(t,f)$ [dB]，補正前のテンプレートパターンのパワーを $T_n(t,f)$ [dB]とする．各時刻で大きい値をとる $w_n(t,f)$ の上位 N_w 点の周波数を求め，その周波数における両者の差

$$\eta_n(t,f)=I(t,f)-T_n(t,f) \tag{9}$$

を求める．そして，求めた N_w 個の $\eta_n(t,f)$ の最小値を $\delta_n(t)$ とし，このときの $w_n(t,f)$ の値を $\sigma_n(t)$ とする．テンプレートパターン全体の時間に対して，$\delta_n(t)>\Theta_\delta$ を満たさない時間の割合が R_δ 以上の場合，入力パターン中にテンプレートパターンは含まれていないと判断する．この場合，これ以後のテンプレートマッチングの処理を行わずに両者の距離は最大値とする．

(2)　全体の音量補正値 Δ_n [dB]を求める．$\delta_n(t)>\Theta_\delta$ を満たす時刻における $\sigma_n(t)$ を重みとして，音量補正値 $\delta_n(t)$ の重み付き時間平均を Δ_n とする．

$$\Delta_n=\frac{\sum_{\{t|\delta_n(t)>\Theta_\delta\}}\delta_n(t)\sigma_n(t)}{\sum_{\{t|\delta_n(t)>\Theta_\delta\}}\sigma_n(t)} \tag{10}$$

こうして求めた音量補正値 Δ_n を用いて，補正後のテンプレートパターンのパワー $T'_n(t,f)$ を

$$T'_n(t,f)=T_n(t,f)+\Delta_n \tag{11}$$

により求める．

4.3.2　音源分離を実現する距離尺度

複数の打楽器音が同時に鳴ると，図5(b)からわかるように，認識対象の音が鳴っているにもかかわらずその他の周波数成分によって距離が大きくなる．そこで，ユークリッド距離に代わる新たな距離尺度を提案する．この距離尺度は，膜鳴楽器を認識する場合と体鳴楽器を認識する場合で差の定義が異なる．

(1)　膜鳴楽器を認識するための距離尺度

複数の打楽器音が同時に鳴った場合に対応するには，入力パターン中にテンプレートパターンが含まれているかどうかを判断する距離尺度を用いればよい．そこで，両者の大小関係をもとに距離を求める．

まず，各時刻 t，周波数 f における $I(t,f)$ [dB]と $T'_n(t,f)$ [dB]との差 $\gamma_n(t,f)$ を

$$\gamma_n(t,f)=\begin{cases}0 & (I(t,f)-T'_n(t,f)\geqq-\Psi)\\ 1 & (I(t,f)-T'_n(t,f)<-\Psi)\end{cases} \tag{12}$$

と定義する．但し，Ψ[dB]は正の定数とする．0でなく Ψ を用いたのは，周波数成分のわずかな変動を吸収するためである．そして，この差 $\gamma_n(t,f)$ に後述する重み関数 $w_n(t,f)$ を掛けて時間周波数平面上で積分することにより，距離 Γ_n を求める．

$$\Gamma_n=\sum_t\sum_f w_n(t,f)\gamma_n(t,f) \tag{13}$$

式(12)では，$I(t,f)$ がほぼ $T'_n(t,f)$ 以上であれば，

$I(t,f)$の中に$T_n'(t,f)$が含まれていると判断して差を0としている．これにより，同時に鳴っている他の打楽器音の成分があっても，入力パターン中にテンプレートパターンが含まれていれば小さい距離を得ることができる．逆に，入力パターン中にテンプレートパターンが含まれていなければ，式(12)により距離は大きくなる．

（2） 体鳴楽器を認識するための距離尺度

膜鳴楽器の距離尺度との違いは，式(14)の$\gamma_n(t,f)$の定義だけである．

$$\gamma_n(t,f)=\begin{cases}0 & (|I(t,f)-T_n'(t,f)|\leqq\Psi)\\ 1 & (|I(t,f)-T_n'(t,f)|>\Psi)\end{cases} \quad (14)$$

つまり，$I(t,f)-T_n'(t,f)>\Psi$のときに$\gamma_n(t,f)=1$となる点が膜鳴楽器の場合と異なる．体鳴楽器の音色はお互いに似ているので，周波数成分の異なる部分を強調するために式(14)のようにする．

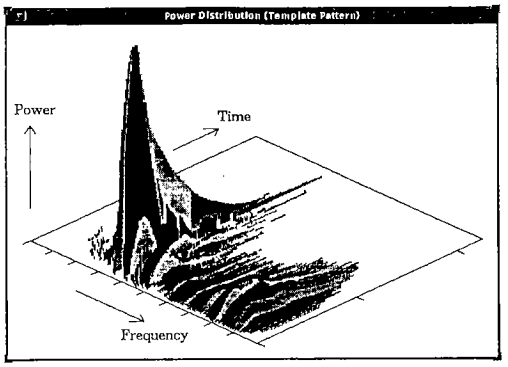
4.3.3 選択的注意の機構

距離を求める際にすべての周波数成分を等しく評価すると，認識するのに関係のない周波数成分が異なっても距離が大きくなる．そこで各楽器ごとに重み関数$w_n(t,f)$を用意し，式(13)のように時間周波数平面上の各楽器固有の領域に重み付けする．重み関数$w_n(t,f)$は，テンプレートパターン上の各楽器固有の特徴で大きな値をとるように，あらかじめ設定しておく．これにより，認識するのに関係のない周波数成分の差$\gamma_n(t,f)$は，距離Γ_nに影響しなくなる．

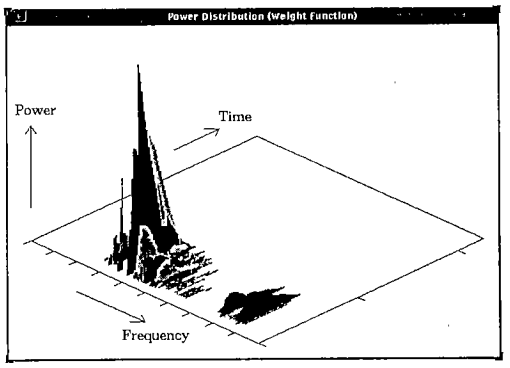
また音量補正法において，各楽器固有の特徴点を得るのにこの重み関数を用いる．つまり，選択的に注意を向ける周波数成分をもとに音量補正する．

重み関数の設定法は以下の通りである．多くの場合，テンプレートパターン上のパワーの大きい周波数成分がある領域で，重み関数が大きな値をとるように設定する．しかし各楽器固有の特徴は，必ずしもパワーが大きい周波数成分であるとは限らない．パワーが小さい周波数成分でも他の打楽器音と異なる特徴であれば，重み関数に大きな値を設定する．そして，できるだけ各楽器ごとに異なる領域を重み付けするようにする．この設定は，試行錯誤を繰り返して行った．具体的な作業手順を以下に示す．

（1） テンプレートパターンに対して時間周波数特性をもつ変形関数を乗算することにより，重み関数の初期値を得る．変形関数は，時間軸方向に設定した折れ線関数$x(t)$と周波数軸方向に設定した折れ線関数$y(f)$との積$x(t)y(f)$で表される．折れ線関数は，パワーの小さい周波数成分をカットし，時間が経つと減衰するように設定する．



(a) Template pattern



(b) Weight function

図6 テンプレートパターンと重み関数の例
Fig. 6 An example of template pattern and corresponding weight function.

（2） 主に以下のように，重み関数の一部の値を変更する．

・他の打楽器音にはない周波数成分がある領域で，重み関数が大きな値をとるように変更．

・発音直後の周波数成分を強調するように変更．

・いくつかの打楽器音に共通に含まれる周波数成分がある領域では，重み関数が小さな値をとるように変更．

（3） 複数の打楽器音が混合した音響信号に対して認識実験を行う．認識実験の結果，誤認識が減れば変更したままとし，増えれば(2)で変更する前の値に戻す．誤認識がなくなった時点で重み関数の設定を終える．

（4） (2)以降の作業を繰り返す．但し，重み関数の値をいくら変更しても誤認識が減らない場合には，**4.4**で述べるしきい値を変更して重み関数を設定し直す．

以上の手順で設定した重み関数の例(SD)を図6に示

す．

4.4 しきい値処理

4.3で求めた距離をしきい値処理し，認識結果とする．しきい値処理は，距離 Γ_n と発音強度（音量補正値 Δ_n）のそれぞれに対して行う．各打楽器ごとにこれらのしきい値を設定し，しきい値以下の距離としきい値以上の発音強度をもつものだけを認識結果とする．発音強度に対してしきい値処理を行うのは，非常に小さい音量で鳴ったと誤認識するのを防ぐためである．

4.5 排他処理

同一楽器制約によるHCとHOの排他処理を行うために，HOの場合には発音時刻を求めるだけでなく，消音時刻も求める．そこで，HOのテンプレートパターン中の持続音部分から時間幅 l で一部を切り出し，消音時刻認識用のテンプレートパターンとする．入力音のパワー分布形状の全時刻に対して，これを用いて再度テンプレートマッチングを行う．こうして得られた距離に対してしきい値処理し，しきい値以下のすべての時間をHOが発音中と判断する．消音時刻は，この発音中の時間の終了時刻から求める．

HOの発音中は，HCが発音したと誤認識することが多い．これは両者が同一楽器による演奏音のため，音色が非常に似ているからである．そこでHOの発音時刻から消音時刻の間は，HCが発音したと認識しないように排他処理を行う．

5. 実験と考察

本システムが有効に機能することを確認するために，あらかじめわかっている楽譜をもとに演奏した打楽器音を入力し，本システムによる認識結果をもとの楽譜と比較する実験を行った．

5.1 実験条件

入力する演奏音には，市販のサンプラー（Roland S-550）をドラムスの音源として用いた．サンプラーの出力音をテープ（Digital Audio Tape）に録音し，その再生音をA-D変換の入力とした†．実験に用いた各打楽器音の持続時間は100～1800 msとさまざまなものがあった．膜鳴楽器の音の多くは40～650 Hzに複数の主要な周波数成分をもち，打撃時のノイズ成分を2～8 kHzに含んでいた．一方，体鳴楽器の音の多くは3～12 kHzに主要な周波数成分をもっていた．

A-D変換は16 bit，44.1 kHzで行い，4,096点の観測区間を100点ずつシフトさせてFFTを行った．複素スペクトル内挿法の結果からパワー分布形状を求めるために，パワー分布形状[cent]では f_c=120 centごとに周波数軸を区切り，パワー分布形状[Hz]では f_c=160 Hzごとに周波数軸を区切った．但し，区切られた区間の総数はともに100個とした．各打楽器音のテンプレートパターンは，サンプラーによって単独で鳴らした打楽器音のパワー分布形状とした．またテンプレートパターンに対する重み関数は，**4.3.3**で述べた作業を行い設定した．重み関数の設定のための打楽器音には，二つの打楽器あるいは三つの膜鳴楽器が同時に発音する全組合せを用いた．テンプレートマッチング等における各パラメータの値を表1に示す．

表1 パラメータの値

Parameter	Value
N_w	10
Θ_δ	−0.34 [dB]
R_δ	0.2
Ψ	0.1 [dB]
l	158 [ms]

5.2 実験結果

まず，ポップスやロックなどで使用される8 Beatのドラムパターンの演奏音を音源分離したときの実験結果の例を示す．図7(a)，図8(a)のドラム譜の演奏音を音源分離した結果（SMF）をそれぞれ図7(b)，図8(b)に示す．比較のために，入力したドラム譜に相当するグラフィックス表示を図7(c)，図8(c)に，出力したSMFに相当するグラフィックス表示を図7(d)，図8(d)に示す．図中の三角形は発音時刻を表し，その横幅が発音強度を表す．また，HOの短い縦線は消音時刻を表す．但し，入力はドラム譜をT=120 ††で演奏した音である．

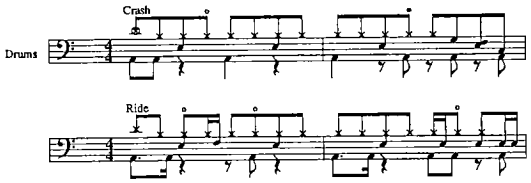
次に，複数の打楽器音が同時に発音する組合せに対する実験結果を述べる．

（1） 二つの打楽器が同時に発音する全組合せを入力して実験した結果，SD+RI，HO+CR，RI+CRの3組を誤認識した．SD+RIでは，本来鳴っていないはずのHCを鳴っていると誤認識した．HO+CRではHOが認識されず，RI+CRではCRが認識されなかった．

（2） 三つの膜鳴楽器が同時に発音する全組合せを入力して実験した結果，誤認識はなかった．

† サンプラーと計算機が異なる場所にあるため，一度テープ（DAT）を経由している．人間が採譜する際も多くの場合テープなどに録音された演奏音に対して行う．そのため我々は自動採譜の環境として，テープの再生音に対して非リアルタイムに採譜ができれば問題ないと考える．

†† 4分音符が1分間に120回鳴るテンポの速さ．

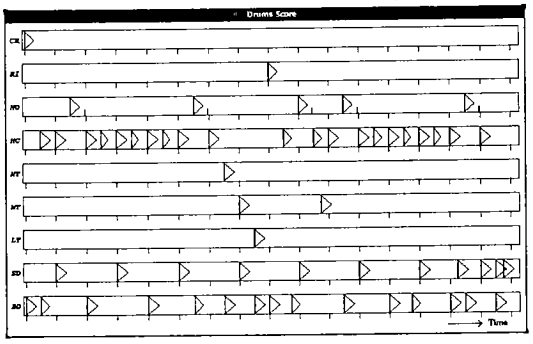


(a) An original score played

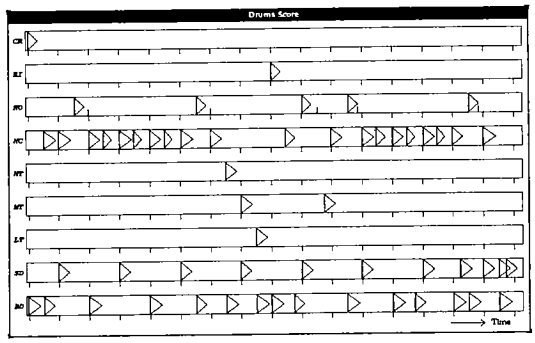
```
HEADER : Format=0  Ntrks=1  Division=48
Tempo = 500000us per MIDI quarter-note
Time Signature = 4/4
MIDI clocks in a metronome click  = 24
32nd-notes in a MIDI quarter-note = 8

[Time] [NoteNum] [GateTime] [Velocity]
    0:  36(C2 )         1          89
    0:  49(C#3)         1          84
   24:  36(C2 )         1          77
   24:  42(F#2)         1          89
   48:  38(D2 )         1          78
   48:  42(F#2)         1          92
   74:  46(A#2)        21          81
   96:  36(C2 )         1          90
   96:  42(F#2)         1          81
  119:  42(F#2)         1          59
  143:  38(D2 )         1          81
  144:  42(F#2)         1          89
  167:  42(F#2)         1          58
  192:  36(C2 )         1          90
  192:  42(F#2)         1          79
  215:  42(F#2)         1          57
  240:  38(D2 )         1          82
  241:  42(F#2)         1          89
  265:  36(C2 )         1          75
  266:  46(A#2)        23          83
  288:  42(F#2)         1          83
  312:  36(C2 )         1          90
  312:  48(C3 )         1          84
  336:  38(D2 )         1          76
  336:  45(A2 )         1          84
  350:  36(C2 )         1          88
  360:  41(F2 )         1          79
  384:  36(C2 )         1          90
  384:  51(D#3)         1          79
  407:  42(F#2)         1          71
  421:  36(C2 )         1          71
  432:  38(D2 )         1          81
  433:  46(A#2)        24          76
  468:  45(A2 )         1          84
  479:  42(F#2)         1          83
  504:  36(C2 )         1          88
  508:  46(A#2)        16          83
  528:  38(D2 )         1          81
  528:  42(F#2)         1          87
  561:  42(F#2)         1          68
  576:  36(C2 )         1          90
  576:  42(F#2)         1          80
  599:  42(F#2)         1          58
  611:  36(C2 )         1          77
  624:  38(D2 )         1          82
  625:  42(F#2)         1          84
  647:  42(F#2)         1          58
  671:  42(F#2)         1          75
  672:  36(C2 )         1          88
  683:  38(D2 )         1          82
  696:  36(C2 )         1          89
  698:  46(A#2)        14          82
  719:  38(D2 )         1          82
  720:  42(F#2)         1          91
  744:  36(C2 )         1          92
  744:  38(D2 )         1          80
  755:  38(D2 )         1          78
```

(b) The output in the form of standard MIDI file

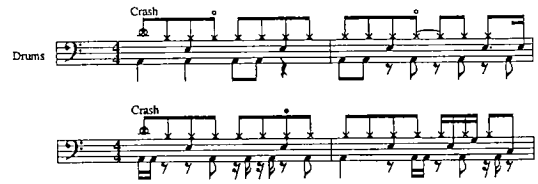


(c) The correct answer



(d) The result of sound source separation

図 7　実験結果 1
Fig. 7　Experimental results 1.

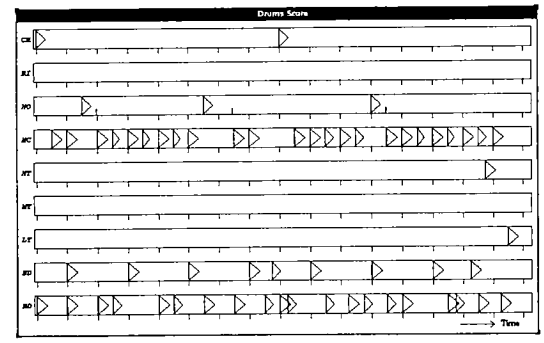


(a) An original score played

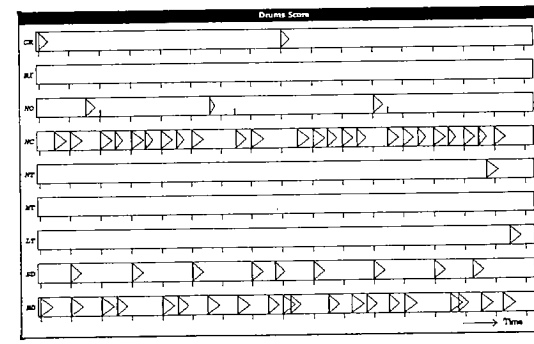
```
HEADER : Format=0  Ntrks=1  Division=48
Tempo = 500000us per MIDI quarter-note
Time Signature = 4/4
MIDI clocks in a metronome click  = 24
32nd-notes in a MIDI quarter-note = 8

[Time] [NoteNum] [GateTime] [Velocity]
    0:  36(C2 )         1          90
    0:  49(C#3)         1          74
   24:  42(F#2)         1          87
   48:  36(C2 )         1          88
   48:  38(D2 )         1          76
   49:  42(F#2)         1          88
   74:  46(A#2)        22          78
   96:  36(C2 )         1          89
   96:  42(F#2)         1          82
  119:  42(F#2)         1          52
  120:  36(C2 )         1          75
  144:  38(D2 )         1          79
  145:  42(F#2)         1          86
  167:  42(F#2)         1          57
  192:  36(C2 )         1          87
  192:  42(F#2)         1          79
  215:  42(F#2)         1          55
  216:  36(C2 )         1          76
  240:  38(D2 )         1          80
  241:  42(F#2)         1          87
  263:  36(C2 )         1          90
  270:  46(A#2)        40          45
  312:  36(C2 )         1          88
  312:  42(F#2)         1          75
  335:  38(D2 )         1          80
  336:  42(F#2)         1          88
  360:  36(C2 )         1          76
  372:  38(D2 )         1          68
  384:  36(C2 )         1          89
  384:  49(C#3)         1          72
  396:  36(C2 )         1          74
  408:  42(F#2)         1          87
  432:  38(D2 )         1          78
  433:  42(F#2)         1          87
  455:  42(F#2)         1          67
  456:  36(C2 )         1          76
  479:  42(F#2)         1          81
  492:  36(C2 )         1          90
  503:  42(F#2)         1          65
  516:  36(C2 )         1          75
  528:  38(D2 )         1          80
  530:  46(A#2)        22          75
  551:  42(F#2)         1          79
  552:  36(C2 )         1          73
  576:  36(C2 )         1          89
  576:  42(F#2)         1          75
  599:  42(F#2)         1          57
  623:  38(D2 )         1          79
  624:  42(F#2)         1          86
  647:  36(C2 )         1          88
  647:  42(F#2)         1          52
  660:  36(C2 )         1          75
  671:  42(F#2)         1          78
  683:  38(D2 )         1          81
  694:  42(F#2)         1          58
  696:  36(C2 )         1          87
  707:  48(C3 )         1          86
  719:  42(F#2)         1          80
  731:  36(C2 )         1          88
  743:  41(F2 )         1          79
```

(b) The output in the form of standard MIDI file



(c) The correct answer



(d) The result of sound source separation

図 8　実験結果 2
Fig. 8　Experimental results 2.

(3) BD, BD以外の膜鳴楽器一つ，体鳴楽器一つの計三つの打楽器が同時に発音する全組合せを入力して実験した結果†，BD+SD+RIで本来鳴っていないはずのHCを鳴っていると誤認識した．

5.3 考　　察

ポップスやロックなどで使用されるドラムパターンに対する実験結果では，実験結果1において3小節目の最初のHOが鳴り終わった直後に発音したHCを認識できなかったほかは，すべての打楽器を正しく認識できた．これは，HOとHCの音色が似ているためにHCの消音時刻をHOの消音時刻と誤認識し，排他処理によってHCの発音が抑制されたからである．認識した発音時刻のずれはほとんどが10 ms以下であり，長くても20数msであった．また，発音強度もほぼ同じものが得られた．本システムが出力したSMFを用いて自動演奏したところ，注意深く聞かなければ入力した演奏と違いのわからない演奏が得られた．**5.2**で示した例と同程度の難易度のいくつかのドラムパターンで実験した結果，16分音符の連打では誤認識することが多く，HOが鳴り終わった直後に発音したHCは実験結果1同様に認識できないことがあった．

二つの打楽器が同時に発音するすべての組合せの実験結果では3組を誤認識し，BDと膜鳴楽器一つと体鳴楽器一つの組合せでは1組を誤認識した．SD+RI，BD+SD+RIではHCが鳴っていると誤認識したが，これは実際に人間が聞いてもHCが鳴っているかどうかの判断がつかないため，誤認識してもやむを得ないと考える．HO+CR，RI+CRのときには，どちらか一方が認識されなかったが，このような体鳴楽器同士の組合せは実際の演奏ではあまりないので，特に問題にはならない．

以上から，本論文で提案した音源分離を実現する認識手法が，

- 9種類の打楽器で構成されるドラムスを対象
- 個々の打楽器音を事前にシステムに登録
- ドラムスの音源として電子楽器を使用
- ポップスやロックなどで使用される8 Beatのドラムパターンの演奏音を入力

という制約のもとで有効に機能することが確認できた．

6. むすび

本論文では，複数の打楽器で演奏された音楽を音源分離するシステムを提案し，その実現方法を述べた．本システムをワークステーション上に実装し，実験とその結果に対する考察を行った．その結果，事前に登録した複数の打楽器音のみによる演奏という制約のもとで，本システムが音源分離できることを示した．

ポップスやロックのように楽音と打楽器音が混合した音楽の自動採譜を実現するためには，楽音と打楽器音の両者を音源分離することが必要不可欠である．従来の音源分離は楽音しか対象にしていなかったのに対し，本研究ではテンプレートマッチングの改良を行うことにより打楽器音の音源分離を実現した．これは，楽音と打楽器音が混合した音楽の音源分離・自動採譜を実現するための第1段階として有効である．

今後の課題を述べる．周波数解析で複素スペクトル内挿法を使用したが，パワー分布形状を求める際にf_cの幅で周波数成分をまとめているため，実際にはBD,LTなどに多く含まれる150 Hz以下の成分に対してしか有効になっていない．今後の拡張で楽音の認識との組合せを行いやすいように今回は使用したが，使用せずにシステムを実現できる可能性がある．これについては今後検討する．実際の打楽器音は，SD一つとってもさまざまな音色があり，その打撃強度によって音色も変化する．今回の手法でそのような音色のバリエーションに対応するためには，大量のテンプレートパターンを用意する必要がある．しかしテンプレートパターンの数が多くなると，現在のように人間がすべての重み関数を設定するのは困難になる．そこで，遺伝的アルゴリズムを用いて計算機に重み関数を自動獲得させることを検討している．また16分音符の連打や，実際に人間が演奏したドラムスの演奏音にも対応していく予定である．最終的には，楽音と打楽器音がともに含まれた音楽を音源分離できるシステムの実現を目指していく．

文　　献

(1) 新原高水，今井正和，井口征士："歌唱の自動採譜"，計測自動制御学会論文集，**20**, 10, pp. 940-945 (1984-10).
(2) 阿久津達也，小池汎平，山内　宗，吉田　実，田中英彦："ICOTone on PSI-Ack II"，第35回情処全大，5Ff-7 (1987).
(3) 戸谷　実，武石浩幸，袖山忠一："異種楽器音の分離に関する研究"，1989信学春季全大，A-180.
(4) 片寄晴弘，井口征士："知的採譜システム"，人工知能学会誌，**5**, 1, pp. 59-66 (1990-01).
(5) 柏野邦夫，田中英彦："音源分離同定システムについての考察"，第43回情処全大，7C-1 (1991).

† 右足でBDを演奏し，片手が膜鳴楽器，反対側の手が体鳴楽器を演奏する場合を想定した実験である．

(6) 柏野邦夫，田中英彦："音源分離要因に関する一検討—共通FMと高調波関係"，1991信学秋季全大，A-107.
(7) 柏野邦夫，田中英彦："音源分離要因における時間的統合—Old-Plus-New Heuristicの導入"，第45回情処全大，7B-3 (1992).
(8) 長束哲郎，才脇直樹，井口征士："異種楽器を対象とした採譜システム"，信学'92春大，D-499.
(9) 柏野邦夫，田中英彦："音源分離要因に関する一検討：II—周波数成分の立ち上がりの時間差および傾きの効果"，信学'92秋大，A-138.
(10) ダイヤグラムグループ編，皆川達夫監修："楽器"，株式会社マール社 (1992).
(11) 原祐一郎，井口征士："フーリエ変換における窓関数の位相特性"，計測自動制御学会論文集，**19**, 7, pp. 551-556 (1983-07).
(12) 原祐一郎，井口征士："複素スペクトルを用いた周波数同定"，計測自動制御学会論文集，**19**, 9, pp. 718-723 (1983-09).
(13) Savitzky A. and Golay M. J. E. : "Smoothing and Differentiation of Data by Simplified Least Squares Procedures", Anal. Chem., **36**, 8, pp. 1627-1639 (July 1964).

付　　録

式(5)，(6)の導出

観測区間の長さを T，虚数単位を j とし，区間周波数 F，振幅 A，位相 ϕ なる単一周波数成分を

$$g(t)=Ae^{j(2\pi Ft/T-\pi F+\phi)} \qquad (A\cdot1)$$

とすると，その複素スペクトルは次式のようになる．

$$z_m=\frac{1}{T}\int_0^T g(t)e^{-j2\pi mt/T}dt$$

$$=\frac{A\sin(\pi F)}{\pi(F-m)}e^{j\phi} \qquad (A\cdot2)$$

ハニング窓 $\nu(t)$ は

$$\nu(t)=\frac{1}{2}\left(1-\cos\frac{2\pi t}{T}\right)$$

$$=\frac{1}{2}-\frac{1}{4}[e^{j(2\pi t/T)}+e^{-j(2\pi t/T)}] \qquad (A\cdot3)$$

であるから，

$$\nu(t)g(t)=\frac{A}{2}e^{j(2\pi Ft/T-\pi F+\phi)}$$

$$+\frac{A}{4}e^{j(2\pi(F+1)t/T-\pi(F+1)+\phi)}$$

$$+\frac{A}{4}e^{j(2\pi(F-1)t/T-\pi(F-1)+\phi)} \qquad (A\cdot4)$$

となる．よってハニング窓を使用したFFTによる複素スペクトル z_m は，式(A・1)，(A・2)から

$$z'_m=\frac{A\sin(\pi F)}{2\pi(F-m)}e^{j\phi}+\frac{A\sin(\pi(F+1))}{4\pi(F+1-m)}e^{j\phi}$$

$$+\frac{A\sin(\pi(F-1))}{4\pi(F-1-m)}e^{j\phi}$$

$$=\frac{A\sin(\pi F)}{2\pi}e^{j\phi}$$

$$\left\{\frac{-1}{(F-m)(F+1-m)(F-1-m)}\right\} \qquad (A\cdot5)$$

と求まる．

ハニング窓を使用したFFTによる複素スペクトルに対して**3.1.1**の方法を適用すると，式(2)，(3)は実際には次式のようになる．

$$f=m+\frac{(u,z'_{m+1})}{(u,z'_{m+1})-(u,z'_m)} \qquad (A\cdot6)$$

$$a=\frac{\pi(f-m)(u,z'_m)}{\sin(\pi f)} \qquad (A\cdot7)$$

これに式(A・5)を代入すると，

$$f=m+\frac{F-m+1}{3} \qquad (A\cdot8)$$

$$a=A\frac{\sin(\pi F)}{\sin(\pi f)}\frac{f-m}{2(F-m)(F+1-m)(F-1-m)} \qquad (A\cdot9)$$

となる．これから，z を式(4)のようにおくと，実際に含まれている成分の区間周波数 F，振幅 A は

$$F=m+z \qquad (A\cdot10)$$

$$A=6z(z-1)\frac{\sin\pi(f-m)}{\sin\pi z}a \qquad (A\cdot11)$$

となり，式(5)，(6)が求まる．

(平成5年7月28日受付，12月20日再受付)

後藤　真孝

平5早大・理工・電子通信卒．現在同大大学院修士課程在学中．音楽情報処理，分散協調処理，並列処理システム，マルチメディアシステムなどに興味をもつ．平4 jus設立10周年記念UNIX国際シンポジウム論文賞受賞．情報処理学会，日本神経回路学会各会員．

村岡　洋一

昭40早大・理工・電気通信卒．昭46イリノイ大大学院博士課程了．同年日本電信電話公社電気通信研究所入所．昭60早稲田大学理工学部教授．Ph. D．並列処理アーキテクチャ，電子化図書，ニューロネットワークなどに興味をもつ．