尚美学園大学芸術情報学部紀要　第20号　抜刷

オーディオーMIDI符号化ツール「オート符」における表情付け解析機能の実装

Implementation of Musical Expression Analysis Functions for Audio-MIDI Encoder Tool “AUTO-F”

2011年11月

茂出木 敏雄

オーディオ－MIDI符号化ツール「オート符」における表情付け解析機能の実装

茂出木 敏雄

Implementation of Musical Expression Analysis Functions for Audio-MIDI Encoder Tool “AUTO-F”

MODEGI Toshio

Abstract

Our previously developed audio to MIDI code converter tool “Auto-F” has a feature of high-precision harmonic tone analysis functions based on the Generalized Harmonic Analysis algorithm. Applying this tool, from given vocal acoustic signals we can create MIDI data, which enable to playback voice-like signals with a standard MIDI synthesizer. However, for general MIDI editing purposes, encoded harmonic tone note-events should be identified and removed. It has been difficult to distinguish harmonic tone note-events from fundamental ones, because a harmonic tone may be based on multiple fundamental tones and some harmonic tone may be also a fundamental one in general music works. Moreover, in some cases microtonal expression control such as a pitch-bend between semi-tone based note events should be analyzed and added. In this paper, we propose an improved frequency analysis algorithm, which can decrease signal analysis processing loads by improving an analysis frame positioning. The improved algorithm can increase a temporal analysis precision, which can distinguish harmonic tone components from fundamental ones, analyze micro-tone notes more precisely.

Key Word

MIDI encoder, vocal, musical expression analysis,
harmonic tone removal, microtonal expression, Auto-F

［抄録］

既開発の音響信号からMIDI符号に自動変換するツール「オート符」は、一般化調和解析に基づく周波数解析を採用することにより倍音を含む和音を高精度に解析できるという特徴があり、音声信号を与えると、標準的なMIDI音源で近似的に音声を再現可能なMIDIデータを生成できる。しかし、本ツールを楽曲のMIDI打ち込み支援の用途に活用する場合、生成されるMIDIデータより倍音に対応する音符だけを高精度に除去する機能が要望される。一般的な楽曲では倍音は複数の基音に由来して発生することが多く、倍音と基音を兼ねる音符も少なくないため識別は容易ではなかった。また、楽曲によっては、半音未満の精度で微分音解析を行い、ピッチベンドなどの表情制御コードを自動的に付加されることが要求される。本稿では、周波数解析における時間分解能と処理速度を向上させることにより、倍音と基音の識別精度および半音間の微分音の解析精度を大幅に向上させることが実現できたので、その結果を報告する。

［キーワード］

MIDI符号化、ボーカル、表情付け解析、倍音除去、微分音表現、オート符

1．はじめに

先報告[1]において、「あらゆる音を音符に変換できる“オート符”」というツールについて言及したが、本稿ではこのツールにフォーカスし、具体的な符号化アルゴリズムの詳細と、最近実施している改良開発項目について述べる。

オーディオの統合編集ツールDAWでは波形オーディオトラックとMIDIデータトラックの混在編集が可能であり、MIDIデータトラックのコンテンツを波形オーディオトラックに変換転送することは可能である。しかし、逆に波形オーディオトラックのコンテンツをMIDIデータトラックに変換転送することは困難であった。そこで、筆者らは、波形オーディオトラックとMIDIデータトラックの相互変換が可能な統合オーディオ編集ツールを提案した[2]。そして、この構想を実現するため、与えられた音響信号に対して一般化調和解析[3]を用いて平均律音階のスケールで高精度な周波数解析を行い、MIDIデータ形式に自動変換する技術の開発を進めてきた[3][4][5]。本技術は「オート符®SA」という名称で汎用的な音響解析ツールとしてまとめ、2001年より財団法人デジタルコンテンツ協会のホームページより無償配布を進めており、主として採譜業務の支援等に活用いただいている[9]。

本解析ツールは、特に和音解析精度が高く、音声信号に適用すると解析されたフォルマント成分がMIDI形式に和音近似され、一般的なMIDI音源を用いてボーカルが再現できるという特徴をもつ。そこで、より明瞭なボーカルが再現できるよう周波数解析機能に改良を加え、SMFファイルに著作権情報などをMIDI形式に符号化したボーカルデータを可聴または非可

聴の形態で埋込んだり[6]、楽器音を用いた新規な音声合成システムへの応用を試みている[7]。また、電子透かし埋め込みなどの音響信号処理を施した前後の各音響信号に対して一般的なMIDI音源で再生可能なMIDIデータに符号化変換を行い、更に2つのMIDIデータの時間軸方向の排他的論理和に対応する差分MIDIデータを作成することにより、音響信号処理による信号劣化成分をピアノロール等により可視化したり、標準MIDI音源により可聴化する手法も提案した[8]。

しかし、本ツールを楽曲のMIDI打ち込み支援の用途に活用する場合、高精度に解析される倍音は返って邪魔であり、生成されるMIDIデータより倍音に対応する音符だけを高精度に除去する機能が要望される。倍音除去機能については、既開発のツール[9]にも実装していたが、一般的な楽曲では倍音は複数の基音に由来して発生することが多く、倍音と基音を兼ねる音符も少なくないため識別は容易ではなかった。そのため、基音を過剰に削除することが多く、実用に供しない面があった。また、楽曲によっては、半音未満の精度で微分音解析を行い、ピッチベンドなどの表情制御コードを自動的に付加されることが要求される。微分音解析機能についても、同様に既開発のツール[8]にも実装していたが、ピッチベンドの制御イベントに割り当てるチャンネルが不適切であったため、通常のMIDI音源で所望の再生音を得ることができず、同様に実用に供しない面があった。本稿では、解析フレームの配置方法を見直し、周波数解析における時間分解能と処理速度を向上させることにより、倍音と基音の識別精度と微分音の解析を大幅に向上させ、倍音を高精度に除去することが実現でき、標準MIDI音源で所望の微分音を再現することができたので、その結果を報告する。

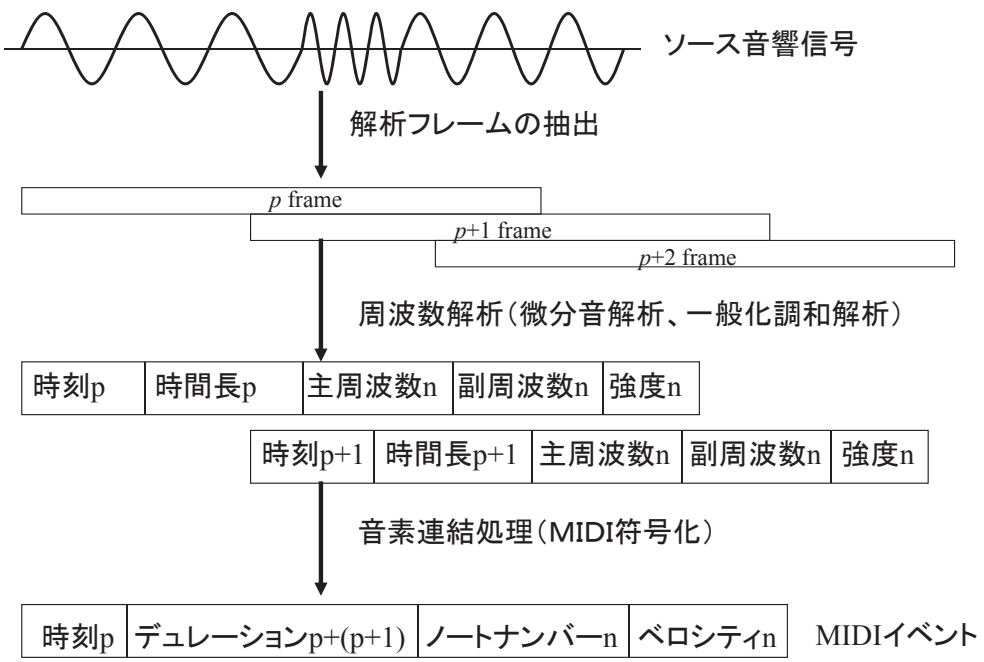


図１　既提案のMIDI符号化処理の概要構成

２．提案する音響信号のMIDI符号化ツールの概要と改良手法

図1は筆者らが提案するMIDI符号化処理の主要構成を示す[5)]。はじめに、与えられたソース音響信号より周波数解析対象のフレームを抽出するが、後続フレームへのシフト幅はソース音響信号の周波数変動を大まかに検出しながら適応的に可変設定するようにしている。即ち、周波数変化が大きい個所ではシフト幅を狭くし、周波数変化が小さい個所ではシフト幅を広くする。続いて、平均律音階の半音（ノートナンバー）単位に非線形な周波数次元で周波数解析を行うが、周波数が高くなるにつれ、半音間隔が粗くなるため、一般的な短時間離散フーリエ変換法DFTでは2和音以上が混在する音響成分を正確に分離解析できない。そこで、一般化調和解析手法[3)]を採用し、併せて周波数ごとに半音間を微分音（副周波数）に分割して解析を行うようにしている。最後に、時間的に隣接する同一主周波数の解析成分（音素）を連結し音符としてまとめ、MIDIイベント形式で符号化する。図1例では、単一のMIDIイベントで記載しているが、実際にはデュレーション情報がなく時刻が異なる2つのノートオンとノートオフ・イベントで符号化される。また、微分音解析結果を基に、ノートオンとノートオフ・イベントの間にピッチベンドやエクスプレッション・イベントなどの表情制御コードを符号化して挿入することもできる。

図2に、筆者らが先に開発したMIDI符号化処理[7)]を基本に改良した具体的な処理構成を示す。本稿では、処理（B-1）から処理（B-3）に示される高時間分解能の改良型周波数解析手法を提案

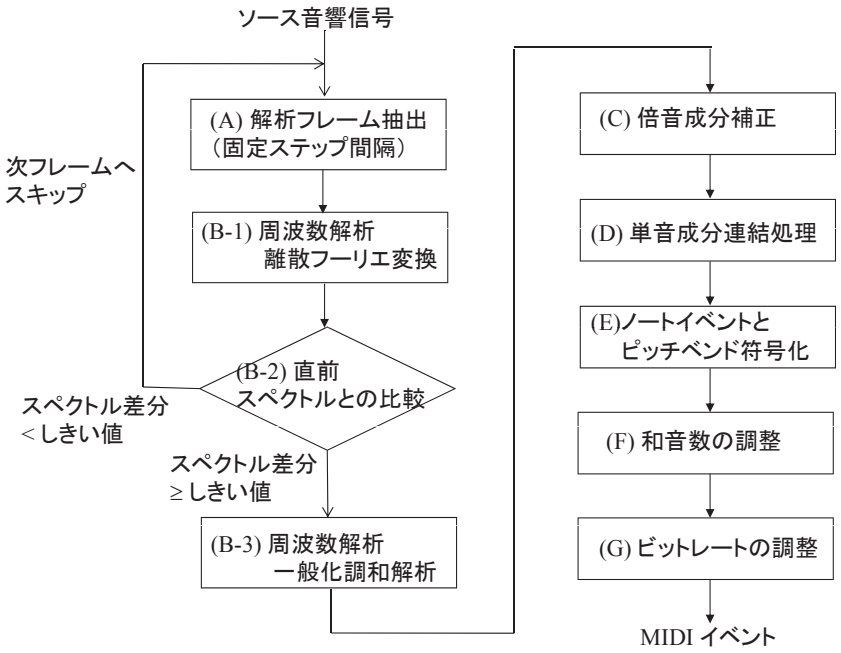


図２　本稿で提案するMIDI符号化処理の具体的な処理構成

し、文献5）で提案した時間軸方向の拡大や延長を不要にして、処理速度の大幅な向上をはかっている。更に、処理（C）を追加することにより高精度な倍音除去を実現し、処理（E）のノートイベントの符号化処理にピッチベンド符号化機能を追加している。以下、既提案の処理を含む処理（A）から処理（G）の各々に対して、７つの節に分けて説明する。

２．１．解析フレーム抽出（固定ステップ間隔）（A）

音響解析では解析フレームを時間軸方向に移動させながら、信号全体の解析を行うが、この際のフレーム長とシフト幅の設定方法について以下述べる。

周波数分解能はフレーム長により変化し、経験上ソース音響信号のサンプリング周波数が44.1[kHz]の場合、低域部まで忠実に解析するためには4096サンプル以上必要である。解析時の解析フレーム長は解析周波数ごとに可変に設定するが、抽出するフレーム長は上限値として、例えば$T=4096/N$を与える。ここで、Nは文献7）で提案した時間軸拡大倍率を示す自然数で、解析対象の音響信号がボーカルなど高時間分解能の処理が要求される場合、例えば$N=4$のように1以上の値を設定する。

一方フレームシフト幅Wは、小さくするほど時間分解能が向上するが計算時間も増大する。そして、解析対象信号が単調である箇所に対して、必要以上にフレームシフトを細かくすると、後述する単音成分連結処理で支障をきたす。そこで、効率的な計算および高精度な単音成分連結処理のためにも、フレームシフト幅Wは解析対象フレームごとに変化させ、最適な値を

設定する方法が望ましい。先提案ではゼロ交差解析を行って、最適なフレームシフト幅を設定していたが[7]、本稿では後述する周波数解析時において離散フーリエ変換を行う段階で決定するようにしたため、ここでは最小値（例えば、$T=4096/N$に対して$W=64/N$サンプルを与える。Nは文献5）で提案した時間軸拡大倍率を示す自然数で）を固定で与えればよい。

２．２．周波数解析（B）

２．２．１．周波数解析・離散フーリエ変換（B-1）

前節で述べた方法により、サンプリング周波数f_sの原音響信号よりp番目に抽出された解析フレームのサンプル配列を$x(p, i)$（$0 \le i \le T-1$）とする。本周波数解析は、n（$0 \le n \le 127$）をMIDIのノートナンバーとして128種の解析周波数$f(n)=440 \cdot 2^{(n-69)/12}$の調和関数を基本にした離散フーリエ変換により行う。ただし、前節で時間軸拡大倍率Nに1以上の値を設定した場合は、$a=12 \cdot \log_2 N$（例えば、$N=2$の場合$a=24$）として、解析するノートナンバーの上限は$127-a$となる。周波数が高くなるにつれ、ノートナンバー間の周波数間隔が広くなるため、特に$n>60$では解析精度が低下してしまう。また、楽曲によっては、半音未満の精度で微分音解析を行い、ピッチベンドなどの表情制御コードを自動的に付加されることが要求される。そこで、ノートナンバー間を以下のようにM個の微分音に分割した$128M$種の調和関数を用いて解析を行う[7]。

$$f(n,m) = 440 \bullet 2^{(n-69+\frac{m}{M})/12} \qquad (1)$$

p番目の解析フレームに対して、ノートナンバー分の相関配列$E(p, n)$（$-a \le n \le 127-a$）と副周波数配列$S(p, n)$を定義し、$-a \le n \le 127-a$および$0 \le m \le M-1$に対して以下式で相関計算を行う。式(2)において$T(n, m)$は解析フレーム長で、調和関数の1周期がフレーム長T以下の場合、フレーム長Tを超えない範囲で調和関数の周期の最大の整数倍になるよう設定し、kを1以上の適当な整数値として、$T(n, m)=k/f(n, m)$で与える。調和関数の1周期がフレーム長Tより大きい場合は本解析処理を行わない。

$$A(p,n,m) = \frac{1}{T(n,m)} \sum_{i=0}^{T(n,m)-1} \left(x(p,i) \sin\left(\frac{2\pi f(n,m)(i+pW)}{f_s} \right) \right)$$

$$B(p,n,m) = \frac{1}{T(n,m)} \sum_{i=0}^{T(n,m)-1} \left(x(p,i) \cos\left(\frac{2\pi f(n,m)(i+pW)}{f_s} \right) \right)$$

$$E(p,n,m) = A(p,n,m)^2 + B(p,n,m)^2 \qquad (2)$$

ここで、p>0で$A(p\text{-}1,n,m)$と$B(p\text{-}1,n,m)$の値が既知の場合、(2)式の$A(p,n,m)$と$B(p,n,m)$を算出する式は以下のように変形でき、直前解析フレームp-1における相関計算結果を用いて、計算範囲を縮小でき高速に算出できる。

$$A(p,n,m)=A(p-1,n,m)-\frac{1}{T(n,m)}\sum_{i=0}^{W-1}\left(x(p-1,i)\sin\left(\frac{2\pi f(n,m)(i+(p-1)W)}{f_s}\right)\right)$$

$$+\frac{1}{T(n,m)}\sum_{i=T(n,m)-W}^{T(n,m)-1}\left(x(p,i)\sin\left(\frac{2\pi f(n,m)(i+pW)}{f_s}\right)\right)$$

$$B(p,n,m)=B(p-1,n,m)-\frac{1}{T(n,m)}\sum_{i=0}^{W-1}\left(x(p-1,i)\cos\left(\frac{2\pi f(n,m)(i+(p-1)W)}{f_s}\right)\right)$$

$$+\frac{1}{T(n,m)}\sum_{i=T(n,m)-W}^{T(n,m)-1}\left(x(p,i)\cos\left(\frac{2\pi f(n,m)(i+pW)}{f_s}\right)\right)$$

$$E(p,n,m)=A(p,n,m)^2+B(p,n,m)^2 \qquad (2')$$

続いて、ノートナンバーnごとに、$0\leq m\leq M-1$の範囲で$E(p,n,m)$を最大にする$E(p,n,m_{max})$を求め、$E(p,n)=E(p,n,m_{max})$、$S(p,n)=m_{max}$と定義する。

2.2.2. 直前スペクトルとの比較 (B-2)

(2)式または(2′)式で算出された$E(p,n)$と直前解析フレームにおける$E(p\text{-}1,n)$との差分$dE(p\text{-}1,p)$を以下のように算出し、$dE(p\text{-}1,p)$が所定のしきい値(例えば、40)未満であれば、2.1節で次の解析フレームに進み、所定のしきい値以上であれば、次の2.2.3節の周波数解析・一般化調和解析へ進む。

$$dE(p-1,p)=\frac{100}{N}\sum_{n=0}^{N-1}\left\{\frac{|E(p,n)-E(p-1,n)|}{E(p,n)+E(p-1,n)}\right\} \qquad (3)$$

2.2.3. 周波数解析・一般化調和解析 (B-3)

解析フレームpはq番目に一般化調和解析を行う可変解析フレームであるとし、解析フレームID配列を$P(q)$とすると、$P(q)=p$と設定し、可変解析フレームqにおいて、ノートナンバー分の相関配列$E_o(q,n)$ ($-\alpha\leq n\leq 127-\alpha$)を定義し、初期値を全て-1とする。

(a) $E_o(q,n)<0$でかつ$E(p,n)$が最大になる$E(p,n_{max})$を求め、$m_{max}=S(p,n_{max})$とする。式(2)を簡素化した以下式(4)を用いて$A(p,n_{max},m_{max})$および$B(p,n_{max},m_{max})$を再計算する。

$$A(p,n_{max},m_{max})=\frac{1}{T(n_{max},m_{max})}\sum_{i=0}^{T(n_{max},m_{max})-1}\left(x(p,i)\sin\left(\frac{2\pi f(n_{max},m_{max})i}{f_s}\right)\right)$$

$$B(p,n_{max},m_{max})=\frac{1}{T(n_{max},m_{max})}\sum_{i=0}^{T(n_{max},m_{max})-1}\left(x(p,i)\cos\left(\frac{2\pi f(n_{max},m_{max})i}{f_s}\right)\right)$$

$$E_o(p,n_{max},m_{max})=A(p,n_{max},m_{max})^2+B(p,n_{max},m_{max})^2 \qquad (4)$$

(b) 上記決定した$A(p,n_{max},m_{max})$および$B(p,n_{max},m_{max})$を用いて、以下式でサンプル配列$x(p,i)$の全ての要素$(0\le i\le T(n_{max},m_{max})-1)$を更新する。

$$x(p,i)=x(p,i)-A(p,n_{max},m_{max})\bullet\sin\left(\frac{2\pi f(n_{max},m_{max})i}{f_s}\right)$$
$$-B(p,n_{max},m_{max})\bullet\cos\left(\frac{2\pi f(n_{max},m_{max})i}{f_s}\right) \qquad (5)$$

(c) 再度(a)の処理に戻り、$0\le n\le 127$の全ての$E_o(q,n)$の値が0以上の値に決定されるまで(a)から(c)までの処理を繰り返す。

2.3. 倍音成分補正(C)

上記算出された$0\le n\le 127$の全ての$E_o(q,n)$の値に対して、2, 3, 4, 5, 6, 7, 8, 9, 10倍の周波数に対応する9個のノートナンバー・オフセットテーブル$N_o(b)$ $(b=0,\cdots,8)$を定義して、次の通り補正を行う。ノートナンバー・オフセットテーブル$N_o(b)$の具体例は、$N_o(b)=\{12, 19, 24, 28, 31, 34, 36, 38, 40\}$である。そして、ノートナンバー$n$に対応する強度値$E_o(q,n)$を次式の通り補正する。

$$E_o'(q,n)=E_o(q,n)-\gamma\sum_{b=0}^{9}\sqrt{E_o(q,n)E_o(q,n-N_o(b))} \qquad (6)$$

γは、$0\le\gamma\le 1$の実数値で倍音補正強度を与える。通常の楽曲では$\gamma=1$に設定し、ボーカルを含む音響信号で、ボーカルを生成する必要がある場合は$\gamma=0$に設定する。下方にシフトさせたノートナンバー$n-N_o(b)$が、$n-N_o(b)<0$か$E_o(q,n-N_o(b))$の値が存在しない場合、$E_o(q,n-N_o(b))=0$として計算し、補正後の$E_o'(q,n)$が$E_o'(q,n)<0$の場合、$E_o'(q,n)=0$とする。補正された相関配列$E_o'(q,n)$を$E_o(q,n)$として次ステップの(D)単音成分連結処理以降は補正後の値を適用する。

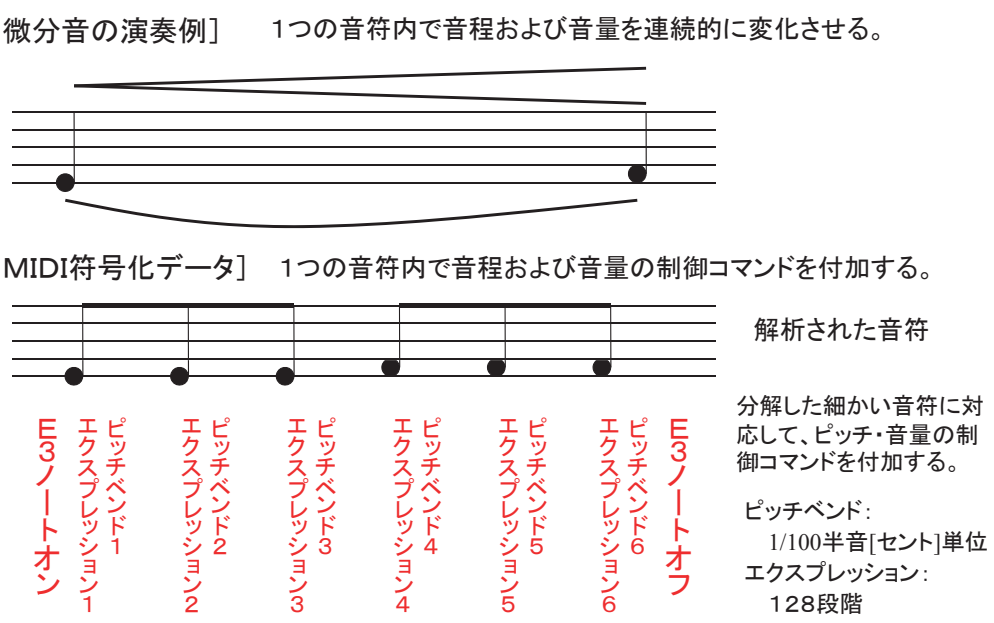


図３　MIDI規格におけるピッチベンド、エクスプレッション・イベントの発行方法

２．４．単音成分連結処理（D）

q番目とq+1番目の可変解析フレームにより周波数解析されたノートナンバーnの単音成分を［時刻（q），時間長（q），主周波数n，副周波数$S(P(q),n)$，強度$E_o(q,n)$］および［時刻（q+1），時間長（q+1），主周波数n，副周波数$S(P(q+1),n)$，強度$E_o(q+1,n)$］とする。時刻（q）および時刻（q+1）は各々$P(q)$番目および$P(q+1)$番目の解析フレームの第1サンプルの原音響信号上の絶対サンプルアドレスをサンプリング周波数で除算することで得られる。時間長（q）は時刻（q+1）－時刻（q）で、時間長（q+1）は時刻（q+2）－時刻（q+1）で与えられる。また、連結する最初の可変解析フレームをq_o番目とし、同様に、ノートナンバーnの単音成分を［時刻（q_o），時間長（q_o），主周波数n，副周波数$S(P(q_o),n)$，強度$E_o(q_o,n)$］と定義する。

連結する最初の$P(q_o)$番目の解析フレームの単音成分と、後続する互いに時間的に隣接する解析フレームの2つ単音成分に対して、ノートナンバーnにおいて上下±1の変移を考慮し、副周波数を考慮した、q_o番目の可変解析フレームとの周波数の差が所定値N_{adif}未満で、隣接する解析フレーム間の周波数の差が所定値N_{dif}未満で、双方の強度が所定のしきい値L_{min}以上でかつ双方の強度の差L_{dif}が所定値以下で両者の連続性が認められる場合、即ち、以下（7-1）～（7-3）の3条件のいずれかを満たす場合、後続単音成分を前方単音成分に連結統合する。

$$S(P(q_o),n)-S(P(q+1),n)|<N_{adif}\text{かつ}|S(P(q),n)-S(P(q+1),n)|<N_{dif}\text{かつ}$$
$$E_o(q,n)>L_{min}\text{かつ}E_o(q+1,n)>>L_{min}\text{かつ}E_o(q+1,n)-E_o(q,n)<L_{dif} \qquad (7\text{-}1)$$

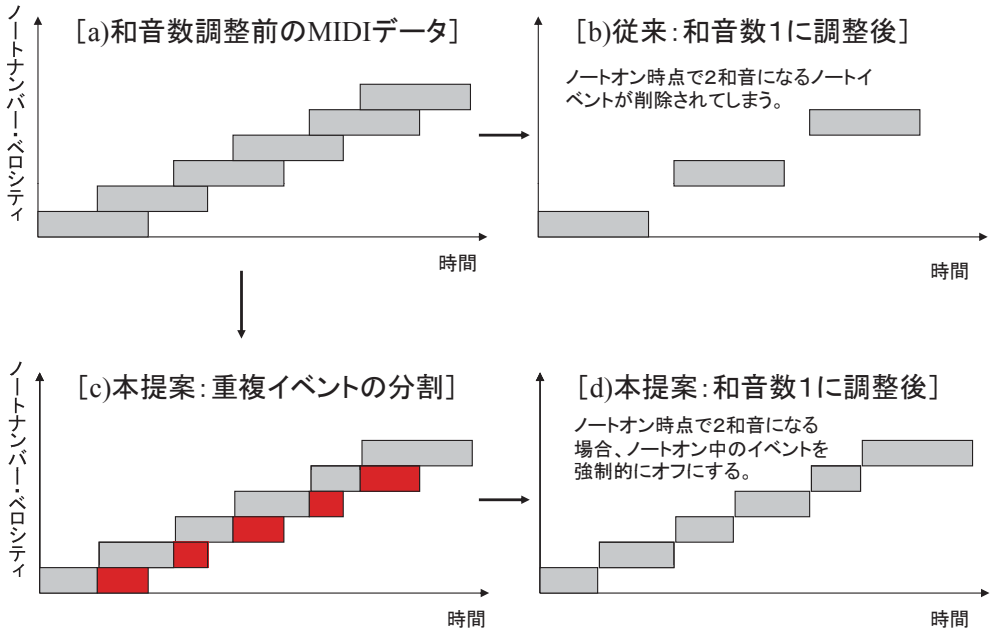


図４　従来の和音数調整機能の問題点と本稿で提案する和音数調整機能の効果

$$|S(P(q_o),n)-S(P(q+1),n-1)-M|<N_{adif}\text{かつ}|S(P(q),n)-S(P(q+1),n-1)-M|<N_{dif}\text{かつ}$$
$$E_o(q,n)>L_{min}\text{かつ}E_o(q+1,n-1)>L_{min}\text{かつ}E_o(q+1,n-1)-E_o(q,n)<L_{dif} \qquad (7\text{-}2)$$

$$|S(P(q_o),n)-S(P(q+1),n+1)+M|<N_{adif}\text{かつ}|S(P(q),n)-S(P(q+1),n+1)+M|<N_{dif}\text{かつ}$$
$$E_o(q,n)>L_{min}\text{かつ}E_o(q+1,n+1)>L_{min}\text{かつ}E_o(q+1,n+1)-E_o(q,n)<L_{dif} \qquad (7\text{-}3)$$

上記連結条件のしきい値の標準的な設定値は、N_{adif}=8/25［単位：ノートナンバー］、N_{dif}=4/25［単位：ノートナンバー］、L_{min}=1［単位：ベロシティ］、L_{dif}=10［単位：ベロシティ］である。連結後の主周波数・副周波数・強度はq_o番目の可変解析フレームの単音成分の各値を採用し、時間長は双方の和、即ち時刻（q+2）－時刻（q_o）で与える。

２.５．ノートイベントとピッチベンド符号化（E）

前節の時系列の単音成分連結処理は、不連続性が認められるまで後続する複数の単音成分に対して繰り返し行い、最終的に統合された［時刻（q_o），時間長（q_o），主周波数n，副周波数$S(P(q_o),n)$，強度$E_o(q_o,n)$］に対して、２つのMIDIノートイベントに変換する。時刻（q_o）で、ノートナンバーnのノートオン・イベントを発行し、ベロシティ値は$E_o(q_o,n)$の最大値をE_{max}として、$128\bullet\{E_o(q_o,n)/E_{max}\}^{1/4}$で与える。時刻については、Standard MIDI Fileでは、直前イベントとの相対時刻（デルタタイム）で与える必要があり、その時刻単位は任意の整数値で定義でき、例えば、1/1536［sec］の単位に変換して与える。そして、時刻（q_o）+時間長（q_o）で、ノ

（a）鳥の声（こまどり）の音響波形

（b）既開発のツール[8]によるMIDI符号化結果

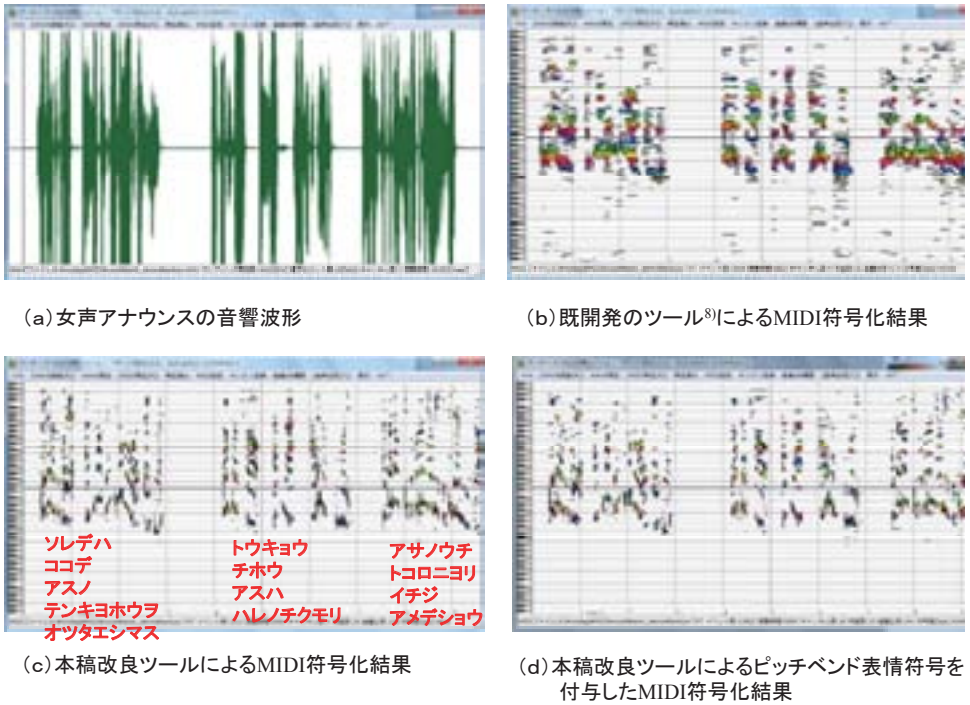
（c）本稿改良ツールによるMIDI符号化結果

（d）本稿改良ツールによるピッチベンド表情符号を付与したMIDI符号化結果

図５　鳥の声（こまどり）に対するMIDI符号化結果

ートナンバーnのノートオフ・イベントを発行する。

より表情豊かなMIDIイベントを作成するためには、図3に示されるように、前節で行った連結処理を行う前の各単音成分を保存しておき、ピッチベンド・イベント（ノートオン後のピッチを1/100半音単位で制御できる）あるいはエクスプレション・イベント（ノートオン後の音量を128段階で制御できる）に符号化して、ノートオンおよびノートオフイベントの間に挿入する。例えば、連結統合された［時刻（q），時間長（q），主周波数n，副周波数$S(P(q),\mathrm{n})$，強度$E_o(q,n)$］に対して、連結前の単音成分の1つを、［時刻（q+1），時間長（q+1），主周波数n，副周波数$S(P(q+1),n)$，強度$E_o(q,n)$］とすると、ピッチベンドの値を$4096\bullet\{S(P(q+1),n)-S(P(q),n)\}/M+4096$、エクスプレッションの値を$128\bullet[\{E_o(q+1,n)/E_{max}\}^{1/4}-\{E_o(q,n)/E_{max}\}^{1/4}]+127$と設定して、ノートオン・イベント発行後のデルタタイム$time(q+1)-time(q)$の時刻にピッチベンド・イベントおよびエクスプレッション・イベントを発行する。この時、ノートオン・イベントとピッチベンド・イベントおよびエクスプレッション・イベントとはチャンネル番号で対応付けを行う。MIDI規格では最大16チャンネルまで使用できるが、第10チャンネルは通常はパーカッション系の非音階楽器に割り当てられているため、このチャンネルを除く15種類のチャンネルのいずれかを各ノートイベント、ピッチベンド・イベントおよびエクスプレッション・イベントに割り当てる。そのため、ピッチベンド・イベントおよびエクスプレッション・イベントを使用する場合、同時に発音できるノートイベントは15和音に制限される。



（a）女声アナウンスの音響波形

（b）既開発のツール[8]によるMIDI符号化結果

（c）本稿改良ツールによるMIDI符号化結果

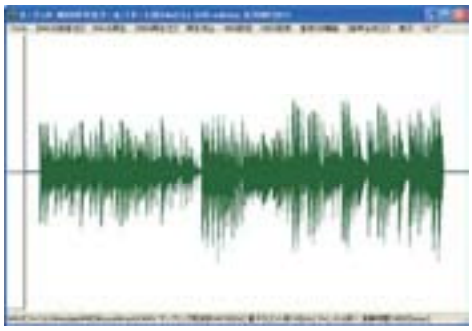
（d）本稿改良ツールによるピッチベンド表情符号を付与したMIDI符号化結果

図６　ヒト女声アナウンスに対するMIDI符号化結果

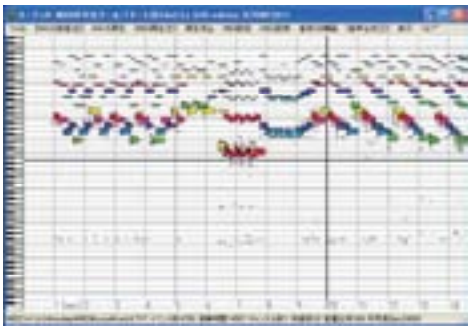
２．６．和音数の調整（F）

MIDI符号に変換する段階で、MIDI音源で処理可能な同時発音数についても考慮する必要がある。時間軸方向に発音期間中（ノートオン状態）のノートイベントの個数を連続的にカウントし、例えば32和音（前節のピッチベンドを使用している場合は15和音）を超えている箇所が見つかった場合は、強制的に優先度の低いノートイベントを削除する処理を行う。従来は、同時に発音している各ノートイベント対のベロシティ値とデュレーション値（ノートオフ時刻－ノートオン時刻）の積（エネルギー値）で優先度を評価し、優先度の低いノートイベントを1対ごと削除する方法をとっていた。そうすると、図4-a）に示されるようにノートイベントが隣接するノートイベントと時間的に重複すると、重要なノートイベントが過剰に削除されてしまう。そこで、本稿では、図4-c）d）に示されるように、優先度の低いノートイベントを分割して部分的にノートイベントを削除する方法を提案する。ノートオン時のベロシティ値に対してノートオン時刻からの経過時間で補正した補正ベロシティ値を算出し、補正ベロシティ値で優先度を評価し、指定和音数以下になるよう優先度の低いノートイベント対を強制的にノートオフさせる補正処理を行う。この際、ベロシティ値またはデュレーション値のいずれかが所定の下限値より低い場合、優先度に関係無く削除する処理も加える。

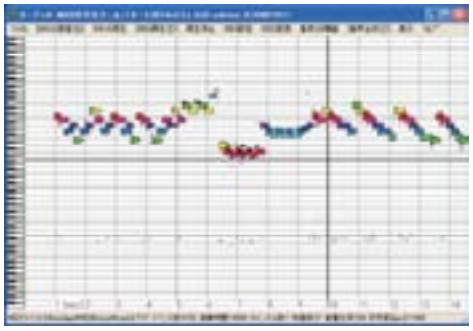
i番目のノートイベント$E_v(i)$のノートオン時刻を$E_v(i).time$、ベロシティ値を$E_v(i).velocity$とすると、時刻t（$>E_v(i).time$）におけるノートイベント$E_v(i)$の補正ベロシティ値$V_c(i,t)$は、



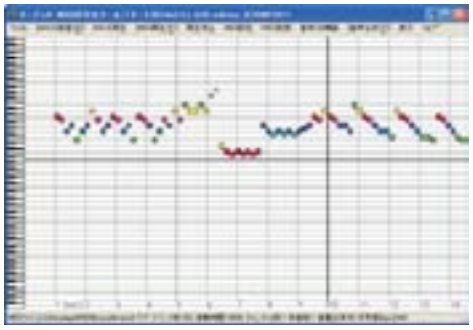
（a）ピアノソロ演奏音の音響波形



（b）既開発のツール[8]によるMIDI符号化結果
倍音除去機能未使用：32和音



（c）本稿改良ツールによるMIDI符号化結果
倍音除去機能使用：32和音



（d）本稿改良ツールによるMIDI符号化結果
倍音除去＋1和音に和音数削減

図７　ピアノソロ演奏音（モーツアルト「きらきら星」）に対するMIDI符号化結果

$$V_c(i,t) = E_v(i).velocity \bullet \exp\{(t - E_v(i).time) \bullet \tau\} \qquad (8)$$

で定義する。τ は減衰係数で例えば－1/1536を与える。（時刻の単位を1秒あたり1536とすると、1秒後に1/2.7に減衰する。）

２．７．ビットレートの調整（G）

MIDI符号に変換する段階で、MIDI音源で処理可能なビットレートについても考慮する必要がある。時間軸方向に例えば1秒間隔にノートオンまたはノートオフイベントの個数をカウントし、各々の符号長を平均5バイト（40bits）としMIDI音源で処理可能な最大ビットレートを9000[bps]とすると（前節のピッチベンドを使用している場合は2倍の18000[bps]程度に設定する）、1秒間あたりイベント数が9000/40=225個を超えている区間が見つかった場合は、その区間に存在するノートオンまたはノートオフイベントと各々対になるノートオフまたはノートオンイベントを近傍区間内で探索し、各ノートイベント対のベロシティ値とデュレーション値（ノートオフ時刻－ノートオン時刻）の積（エネルギー値）で優先度を評価し、指定イベント個数（225）になるよう優先度の低いノートイベント対を局所的に削除する処理を行う。この際、ベロシティ値またはデュレーション値のいずれかが所定の下限値より低い場合、優先度に関係無く削除する処理も加える。

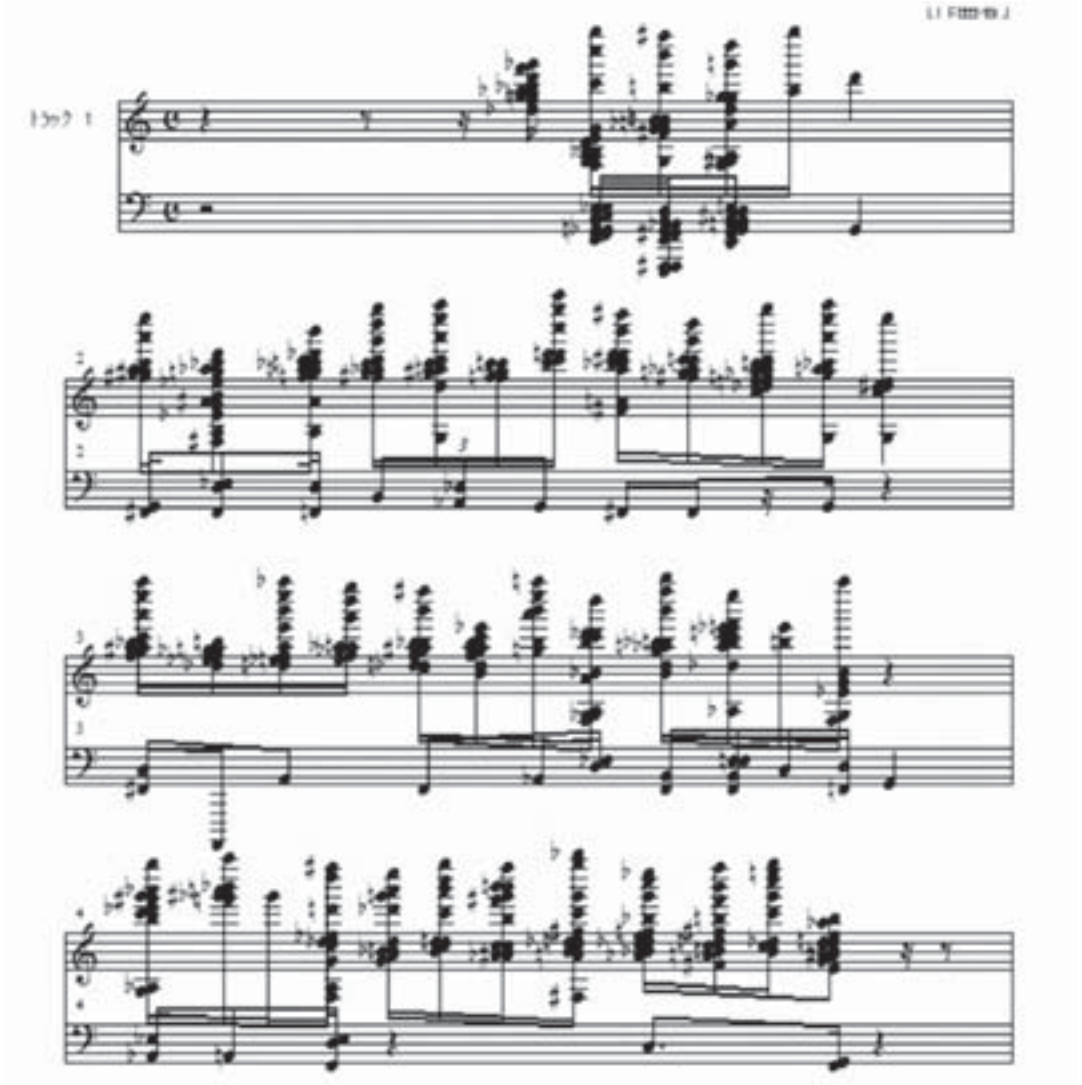


図８　図７（ｂ）のMIDI符号化データに対する五線譜変換結果

３．おわりに

公開サイト[9]で提供している「オート符version2.6」と前節で述べた改良方法を実装したツールを用いて、鳥「こまどり」の声に適用した符号化結果を図5（a）～（d）に示す。図5（b）～（d）の符号化画面内の着色された小さな矩形は音符（ノートイベント）を示し、横軸は時間で、横幅はノートオンからノートオフ区間を示す。縦軸は音高（ノートナンバー）を示すとともに、縦方向の幅でベロシティも示している。図5（b）の従来の「オート符version2.6」で符号化を行った場合、図5（a）の波形で示される音素パターンに追従できていなかったが、本稿提案の改良手法を実装したツールで符号化を行うと、図5（c）（d）で示されるように原音波形の音素パターンに適切に追従できていることがわかる。図5（c）（d）とも、時間軸拡大倍率N=4に設定

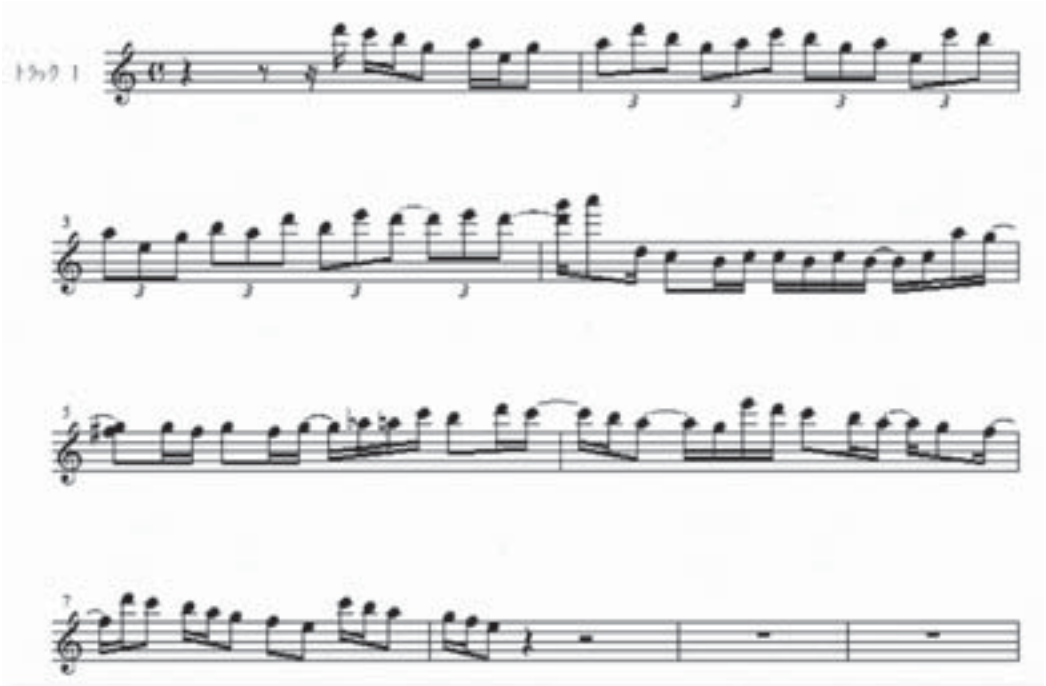
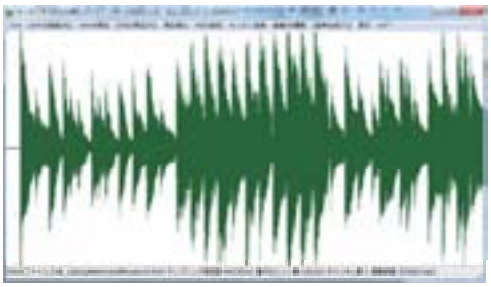


図９　図７（ｄ）のMIDI符号化データに対する五線譜変換結果

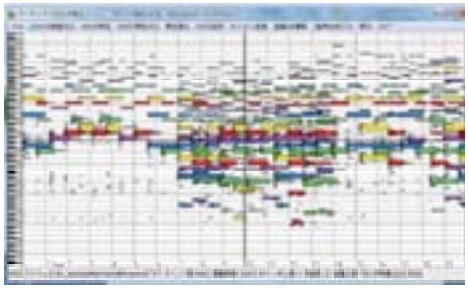
し、図5（c）では最大和音数32でノートイベントのみによる符号化を行った結果である。図5（d）では、最大和音数15でピッチベンドおよびエクプレッション・イベントの符号化も行った結果である。ただし、画面上にはノートイベントしか図示されていない。特に、図5（d）ではGM標準MIDI音源を用いて、最も原音に近い再生音を実現できていることを確認した。

同様な処理をヒト女声のアナウンスに適用した結果を図6に示す。図6（b）の従来の「オート符version2.6」で符号化を行った場合、図6（a）の波形に含まれる各音節パターンが分離できておらず、再生音声が不明瞭であったが、本稿提案の改良手法を実装したツールで符号化を行うと、図6（c）（d）で示されるように原音波形の音素パターンに適切に追従できていることがわかる。図6（c）（d）とも、時間軸拡大倍率N=4に設定し、図6（c）では最大和音数32でノートイベントのみによる符号化を行った結果である。図6（d）では、最大和音数15でピッチベンドおよびエクプレッション・イベントの符号化も行った結果である。ただし、画面上にはノートイベントしか図示されていない。特に、図6（d）ではGM標準MIDI音源を用いて、最も明瞭な音声を再生できることを確認した。

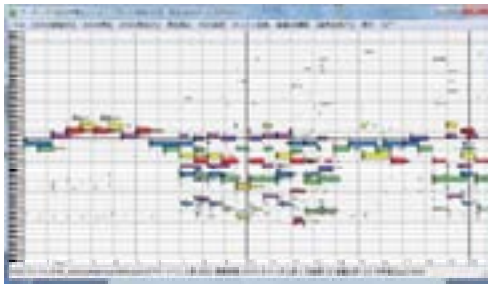
次に、図7（a）に示される単旋律のピアノ独奏曲（モーツアルト「きらきら星変奏曲」、冒頭約15秒、44.1kHz/16bits/1-ch）に対して、従来のツールを用いて最大和音数32に設定して符号化を行った結果を図7（b）に示す。また、同データをイーフロンティアPrintMusic2008使用して五線譜に変換した結果を図8に示す。倍音が目立ち判読性の悪い譜面になっている。図7（b）



（a）ピアノソロ演奏音の音響波形



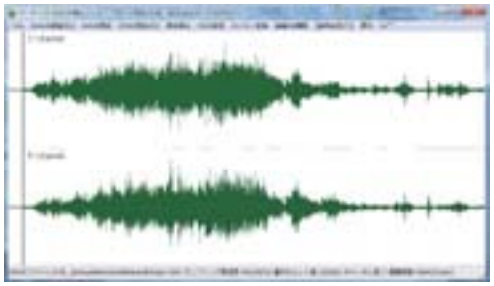
（b）既開発のツール[8]によるMIDI符号化結果
倍音除去機能未使用：32和音



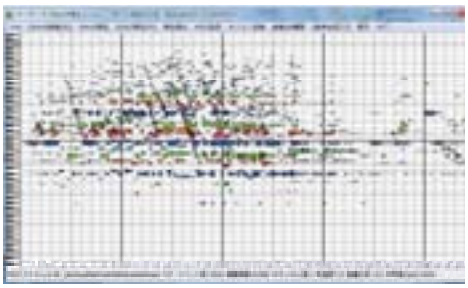
（c）本稿改良ツールによるMIDI符号化結果
倍音除去機能使用：32和音

（d）楽曲（a）に対するMIDI打ち込みデータ
ベロシティ・パラメータ均一

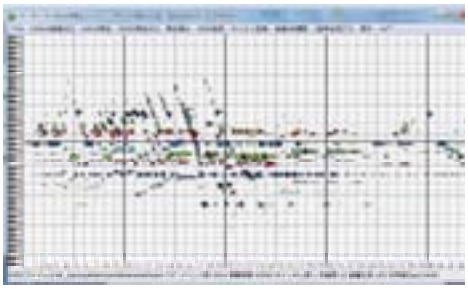
図10　ピアノソロ演奏音に対するMIDI符号化結果その2
（ムスルグスキー組曲「展覧会の絵」、ピアノ独奏版、プロムナード冒頭約20秒、
44.1kHz/16bits/2-ch）



（a）ピアノソロ演奏音の音響波形



（b）既開発のツール[8]によるMIDI符号化結果
倍音除去機能未使用：32和音



（d）本稿改良ツールによるMIDI符号化結果
倍音除去機能使用：32和音

図11　ピアノソロ演奏音に対するMIDI符号化結果その3
（ショパン「幻想即興曲」、冒頭約43秒、44.1kHz/16bits/2-ch）

に示される従来ツールを用いて最大和音数32で符号化したMIDIデータに対して最大和音数1に削減するように設定すると、倍音の代わりに基音が除去されたり、直前の音符がオーバーラップしている基音が一部欠けたり、して、演奏者が弾いた音符を正しく再現できなかった。

一方、図7(c)に、本稿提案ツールを用いて最大和音数32で符号化したMIDIデータを示す。同データに対して最大和音数1に削減するように設定した結果、図7(d)に示すようになり、同データを五線譜に変換した結果を図9に示す。図9の結果は演奏者が弾いた音符を完全に反映できている。

和音を含む一般的なピアノ独奏曲で試したところ、図10に示されるテンポが比較的遅い曲(ムソルグスキー「展覧会の絵」ピアノ独奏版、プロムナードなど)では、図7には存在しない最大6重和音の全てが、演奏者が弾いた通り正確に再現できることを確認した。しかし、図11に示されるようなショパンの幻想即興曲のようにテンポが速い楽曲については、倍音除去を行う前段階で正確に拾えない音符があり、(B)周波数解析における時間分解能の更なる改善と(D)単音成分連結処理の高精度化が今後の課題となる。

以上、本稿で紹介したツールは、過去約10年間にわたって筆者が担当している本学・情報表現学科の授業の教材として使用してきたもので、並行して改良開発も進めてきた。本学の学生や教職員の皆様とのインタラクションが本研究開発の推進に多大な影響を与えたものと考えており、改めて本学関係者の皆様に謝意を示す。また、前述の通り、本稿で紹介したツールについては、筆者が担当している「マルチフィールド体験演習」等の授業で既に活用しているが、教育・研究・その他にご活用されたい場合はソースコードを含め提供可能ですので、ご連絡ください。

本稿の付録として、最新版の「オート符version3.6」のソフトウェア(日本語WindowsGUI版およびバッチ処理版実行形式とマニュアル)“オート符v36体験版.zip”がありますので、興味を持たれた方は、電子版をご覧ください。

引用文献

1. 茂出木敏雄『聴覚芸術への情報学的アプローチと音楽情報処理ツールの開発事例』尚美学園大学芸術情報研究, Vol.18, pp.15-35, November 2010.
2. T. Modegi, S. Iisaku : “Proposals of MIDI coding and its application for audio authoring,” Proceedings of IEEE International Conference on Multimedia Computing and Systems, IEEE Press, USA, pp.305-314, June 1998.
3. Toshio Modegi : “Multi-track MIDI Encoding Algorithm Based on GHA for Synthesizing Vocal Sounds,” Journal of Acoustic Society of Japan, Vol.20, No.4, pp.319-324, April, 1999.
4. Toshio Modegi : “Very low bit-rate audio coding technique using MIDI representation,” Proceedings of the ACM 11th international workshop on Network and operating systems support for digital audio and video, pp. 167-176, New York, USA, June 2001.
5. 茂出木敏雄 『音響信号の平均律音階に基づく汎用解析ツール「オート符」の開発』電気学会・電子情報システム部門誌 Vol.123-C, No.10, pp. 1768-1775 October 2003.
6. 茂出木敏雄『音声MIDIコードを用いたSMFファイルへの情報埋め込み手法』2009電子情報通信学会・総合大会, 情報システム講演論文集2, DS-3-3, pp.S35-S36, March 2009.
7. 茂出木敏雄 『MIDI符号化ツール「オート符」を用いた音素MIDIコードの設計と楽器音による音声合成機能の実現』電気学会・電子情報システム部門誌 Vol.130-C, No.7, pp.1159-1167, July 2010.
8. Toshio Modegi : “Evaluation Method for Quality Losses Generated by Miscellaneous Audio Signal Processings Using MIDI Encoder Tool 'Auto-F',” Proc. of IEEE TENCON2010, pp. 2066-2071, Fukuoka Japan, November 2010.
9. 財団法人デジタルコンテンツ協会d-CON Support, http://www. dcaj. org/d-con/frame09. html（「オート符®SA」の無償配布元）.