2008－MUS－76 (19)
2008／8／8

フーリエ係数の時間変動に着目した音符の連結処理の検討

坂内秀幸† 田所嘉昭†

我々は，自動採譜システムの構築にあたって，くし形フィルタに基づく音高推定アルゴリズムを提案している．自動採譜システムに必要な処理として，音高推定の他に，音符の長さを決定する音価割当や，曲のテンポや拍節を推定するビートトラッキングなどが挙げられる．本稿では，楽曲の短時間フーリエ変換 (STFT) を行い，フーリエ係数の時間変動を利用して，音価割当に必要な音符の連結処理について検討する．実験の結果，オクターブを限定した 4 和音以下のピアノ曲において，約 94 %の正答率で音符の連結処理が可能であることを確認した．

Consideration of Note Connection Processing Noticing Time Changes of Fourier Coefficients

Hideyuki Sakauchi † and Yoshiaki Tadokoro†

We proposed the pitch estimation algorithm based on comb filters for construction of the automatic transcription system. A note value assignment which determines a length of each note, and a beat tracking to estimate tempo and beat of a music are necessary quoted for an automatic transcription system other than a pitch estimation. In this paper, the musical data are processed short-time Fourier transform (STFT), and the connection processing of each note required for a note value assignment is considered using time jitters of Fourier coefficients. As the experimental result, the connection processing of each note was realized with the accuracy of about 94 % in piano music data which are limited in one octave and composed of chords including four notes or less.

1. まえがき

音楽において，楽曲を楽譜として記述することを採譜という．採譜を行うには，音楽的な知識や経験が必要であるが，計算機で自動採譜を行うことができれば，音楽的な知識や経験のない人にでも楽譜を作成することができる．また，自動採譜システムが確立されれば，MIDI ケーブルで接続することのできる楽器（MIDI キーボードなど）でなくても，演奏から楽譜を作ることができ，有用であるといえる．

我々は今まで，自動採譜に必要な技術として，音高推定に関する研究を主に行っており，くし形フィルタに基づいた推定手法によって，良好な結果を得ている[3]．しかし，音高推定だけでは，各時刻における音高の検出によって，ピアノロール形式の結果が得られるだけであり，音符の拍の長さ（音価）を求めるためには音価割当の処理が必要になる．そこで本稿では，MIDI データから作成した楽曲の理想的な音高推定結果と，検出した onset（発音開始時刻）を元に，音価割当の手法を提案する．音価割当を行う際に，最も問題となるのが音符の連結処理であり，フーリエ係数の時間変動を利用した手法によってこの問題の解決を図る．

† 豊橋技術科学大学
Toyohashi University of Technology

2. 音符の連結処理の必要性

音価割当に必要な処理として，音符の連結処理がある．楽曲において一度に発音される和音が，全て同時に発音され，全て同時に鳴り終わるような演奏（いわゆるコード演奏）のみであれば，音符の連結処理を行わなくても正しい採譜が可能であるが，実際の楽曲はそうはならない．ある音符が発音され，鳴り終わる前に別の音符が発音されることは頻繁に生じる．例として，図 1(a) のような演奏と，図 1(b) のような演奏を考える．ここで，横軸は時間，縦軸は音高を表している．この演奏を採譜するとき，図 1(c) の onset を音符の開始位置とすると，どちらも図 1(b) のように認識されてしまう．そこで，図 1(a) のよう

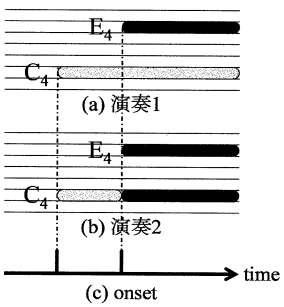

図 1 連結処理の必要性

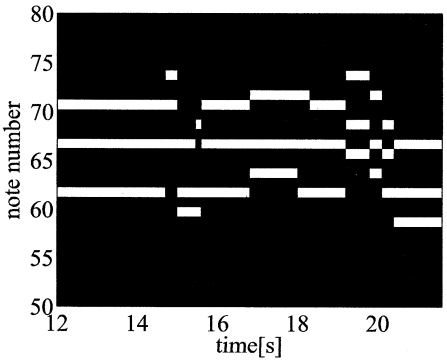

図 2 理想的な音高推定結果

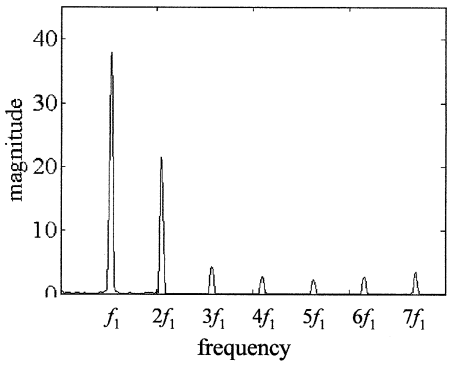

図 3 楽音の調波構造

な演奏において，実際には C_4 の音符が繋がっているということを検出できれば，正しく採譜を行うことができる．この繋がりの検出を音符の連結処理と呼び，図 1(a),(b) のように同じ音高が続く場合の検出箇所を連結点と呼ぶことにする．図 1(a) の場合，C_4 の音符は連結点で接続として検出されれば正答であり，図 1(b) の場合，C_4 の音符は連結点で切断として検出されれば正答である．

3. 入出力と処理の流れ

検出手法の前に，用いる入出力データと大まかな処理の流れについて説明しておく．

3.1 入力楽曲

テンポの一定な 4/4 拍子のピアノ曲を MIDI で用意し，サンプリング周波数 $f_s = 44100$[Hz] としてモノラル録音したものを入力楽曲とする．このとき，楽曲は様々な連結パターンを含むように作成している．なお，広いオクターブを使用する楽曲にはまだ対応しておらず，一度に発音される和音が，1 オクターブ以内におさまる演奏を用いることとする．

3.2 出力シーケンス

自動採譜を行う際に最低限必要な情報として，各音符の発音開始拍，音高，音価，楽曲のテンポが挙げられる．[2] ここでは各音符の情報を一連の流れのシーケンスとして扱い，i 番目のシーケンスは，音高 $N_{note}(i)$，発音開始拍 $B_{note}(i)$，音価 $V_{note}(i)$ 拍の音符を表すこととする．なお，音高 $N_{note}(i)$ の値は，MIDI ノートナンバーに合わせて C_4 を 60 とし，半音上がるごとに +1，半音下がるごとに −1 された値を用いる．

3.3 理想的な音高推定結果

MID2TXT[5] を利用して入力楽曲の MIDI データをテキストデータに変換し，そのテキストデータを元に理想的な音高推定結果を出力するようなスクリプトを組んだ．図 2 が理想的な音高推定結果の例であり，縦軸は音高 $N_{note}(i)$，横軸は時間 [s] であり，白色の部分が音高の存在する箇所を表す．この出力からわかることは，ある時刻にどのような音高が存在するかという情報だけであり，音符の途中に発音のし直しがあっても，その切れ目の存在はわからない．なお，我々が提案している音高推定アルゴリズムで実際に楽曲の音高推定を行った場合，ここまで誤差のない結果にはならないが，楽曲の onset と音高推定結果を照らし合わせることで，多少の誤差が存在しても正しい採譜結果となるような処理を行う．

3.4 処理の流れ

全体的な処理は以下の手順で行う．

(1) MIDI データから理想的な音高推定結果を出力し，入力楽曲から onset およびテンポを検出する．

(2) 音高推定結果，onset，テンポから，onset で分割された音符のシーケンスを求める．

(3) 音符の連結処理によって，分割されたシーケンスに連結を施す．

4. 楽音の調波構造

楽音の定常部分における周波数スペクトルは，図 3 に示すように，基本周波数 f_1 と，その倍数に位置する倍音周波数 $2f_1, 3f_1, \ldots$ に等間隔にピークを持つという特徴があり，この構造を調波構造と呼ぶ．また，音高 $N_{note}(i)$ の楽音の 2 倍音は，音高 $N_{note}(i)+12$ の楽音の基本周波数に重なり，音高 $N_{note}(i)$ の楽音の 3 倍音は音高 $N_{note}(i)+7$ の楽音の 2 倍音に重なるなど，和音が演奏された場合，倍音に重なりが生じることがある．なお，楽音の発音開始直後など，過渡的な部分においては調波構造のピークの間にノイズのようなスペクトルが現れることが多い．このノイズは，onset の検出においては特に問題ないが，音符の連結処理においては，どの音高の発音によるものなのかわからないため，処理の妨げとなることがあり，音符の連結処理を困難にしている原因の 1 つといえる．

5. 提案手法

5.1 テンポとビート長

テンポとは，楽曲の演奏などにおいて，1 分間に何拍が刻まれるかを表す値であり，ビート長は 1 拍の時間長

を表す．例えばテンポ120[bpm]といえば，1分間に120拍（ビート長0.5[s]）の速さで拍が刻まれることになる．ビート長をL[s]，テンポをT[bpm]とすると，式(1)のような関係が成り立ち，どちらかがわかれば，もう片方も算出することができる．

$$T = \frac{60}{L} \tag{1}$$

テンポは，STFT(Short Time Fourier Transform: 短時間フーリエ変換)によって周波数ごとのスペクトルの急変部分を求めることによって検出できることが知られている[4),1)]が，ここでは説明を省略し，テンポTは既知として扱う．

5.2 音価割当

5.2.1 onset

テンポTの値を用い，音符の最小単位をR_{note}分音符(16分音符など)として楽曲中のonsetを求める．

まず楽曲信号を$x(n)$とし，窓幅W_{wx}サンプルのハミング窓をシフト幅W_{sx}サンプルずつシフトさせて，楽曲全体にSTFTを適用する．STFT結果のスペクトル値を$s(l_x, k_x)$とする．l_xはSTFT結果の時間のインデックスであり，窓のシフト幅W_{sx}と楽曲信号のサンプル数nを用いて式(2)で表される．

$$l_x = \left\lfloor \frac{n}{W_{sx}} \right\rfloor \tag{2}$$
$$(\lfloor\ \rfloor : \text{integer by rounding down})$$

また，k_xはFFT(STFT)結果の周波数インデックスであり，サンプリング周波数f_sと楽曲信号の周波数fを用いて式(3)で表される．

$$k_x = \left\lfloor \frac{fW_{wx}}{f_s} \right\rfloor \tag{3}$$
$$(\lfloor\ \rfloor : \text{integer by rounding down})$$

STFT結果$s(l_x, k_x)$において，式(4)を計算する．

$$s_h(l_x) = \sum_{k_x=0}^{W_{wx}-1} s(l_x+1, k_x) - s(l_x, k_x) \tag{4}$$
$$(l = 0, 1, 2, \ldots)$$

$s_h(l_x)$は，時間軸方向の急変部分，すなわちonsetに大きなピークを持つ波形となる．このとき$s_h(l_x)$は，サンプリング周波数f_sの入力信号をW_{sx}おきに抜き出して計算した値であるので，$s_h(l_x)$はサンプリング周波数f_s/W_{sx}の時間波形であると見なすことができる．ここで，スペクトルの小さな変化，すなわちR_{note}分音符の時間長よりも短い周期で変化する値を除去することでonsetの検出を行う．そのために，$s_h(l_x)$を式(5)のf_c以上の周波数をカットするLPFに通し，この波形を$s_{hl}(l_x)$とする．

$$f_c = \frac{R_{note}}{4} \frac{T}{60} \left(\frac{f_s}{W_{sx}}\right)^{-1} \tag{5}$$

$s_{hl}(l_x)$に含まれるピークは，音符が発音されることで生じたピークである．$s_{hl}(l_x)$の極大値の多くは発音開始時刻に生じるが，楽音のスペクトルは減衰時に小さなピークを生じることがあるため，これを考慮して大きなピークだけを取り出す必要がある．そのためにまず，$s_{hl}(l_x) > s_{hl}(l_x-1)$且つ$s_{hl}(l_x) > s_{hl}(l_x+1)$となる$l_x$を$j$番目の極大値が存在するインデックス$p_{max}(j), (j = 0, 1, 2, \ldots)$として取り出す．また，$l_x > p_{max}(0)$で$s_{hl}(l_x) < s_{hl}(l_x-1)$且つ$s_{hl}(l_x) < s_{hl}(l_x+1)$となる$l_x$を$j$番目の極小値が存在するインデックス$p_{min}(j)$とする．このとき，式(6)に当てはまる極大値は楽音の発音によるものではないとしてonsetから除外し，残った極大値の存在する時刻をonsetであるとする．

$$\frac{s_{hl}(p_{max}(j-1)) - s_{hl}(p_{min}(j-1))}{s_{hl}(p_{max}(j)) - s_{hl}(p_{min}(j-1))} > 10 \tag{6}$$

5.2.2 onsetで分割されたシーケンス

5.2.1で求めたonsetの存在する時刻が音符の発音開始時刻となるので，onsetおよびビート長Lを音高推定結果と照らし合わせることで，onsetで分割されたシーケンスの検出を行う．ただし，このとき音高推定結果の時間軸のインデックスはl_xに統一させておく．

音高推定結果において，$p_{max}(j)$から$p_{max}(j+1)$に$\{p_{max}(j+1) - p_{max}(j)\}/2$個以上の音高が存在すれば，その音高を$i$番目の音高$N_{note}(i)$とする．また，発音開始拍および音価は式(7),(8)によって求められる．

$$B_{note}(i) = \left[\frac{p_{max}(j) - p_{max}(0)}{R_{index}}\right] \left(\frac{R_{note}}{4}\right)^{-1} \tag{7}$$
$$([\ ] : \text{integer by rounding})$$

$$V_{note}(i) = \left[\frac{p_{max}(j+1) - p_{max}(j)}{R_{index}}\right] \left(\frac{R_{note}}{4}\right)^{-1} \tag{8}$$
$$([\ ] : \text{integer by rounding})$$

ここでR_{index}は，R_{note}分音符の時間長をインデックスl_x上での長さに換算した値であり，式(9)で与えられる．

$$R_{index} = L \left(\frac{R_{note}}{4}\right)^{-1} \frac{f_s}{W_{sx}} \tag{9}$$

以上の手順により得られた$B_{note}(i)$,$N_{note}(i)$,$V_{note}(i)$は，全ての音符がonsetで分割されたシーケンスとなる．

5.2.3 音符の連結処理

全ての音符がonsetで分割されたシーケンスに対し，周波数スペクトルの時間変動を利用した音符の連結処理によって繋がりを検出する．まず，入力波形$x(n)$において，連結点の周辺部分（onsetを中心に，onsetが1つだけ入る範囲）のみを抜き出した波形を$y(m), (m = 0, 1, 2, \ldots, M-1)$とする．$y(m)$に対して，窓幅$W_{wy}$サンプルのハミング窓を，$W_{sy}$サンプルずつシフトさせてSTFTを行う．ここで，窓幅W_{wy}の値は，楽音の周波数を区別するのに必要な分解能を得るために，極力大きな値を取ることが望ましいが，onsetを1つしか含まないように$y(m)$を抜き出す場合，$y(m)$のサンプル数はテンポ60~ 180[bpm]の楽曲で7000~ 20000サンプル程度しか取ることができず，W_{wy}を増やしすぎると見られる時間変動が少なくなってしまう．そのため，本稿では窓幅を$W_{wy} = 4096$とし，また，時間変動のサンプル数

を増やすために $W_{sy}=100$ とする．　次に，検出対象音の音高は既知であるので，STFT の結果から，検出する音高の基本周波数，2 倍音の周波数，...，7 倍音の周波数におけるスペクトルの時間的な変化を見る．ここでは，時刻 l_y における基本周波数から 7 倍音の高調波までのスペクトルの大きさを，それぞれ $s_1(l_y), s_2(l_y), \ldots, s_7(l_y)$ とし，この 7 つの波形を用いることにする．なお，l_y は STFT 結果の時間のインデックスであり，m を用いて式 (10) で表される．

$$l_y = \left\lfloor \frac{m}{W_{sy}} \right\rfloor \tag{10}$$

($\lfloor\ \rfloor$: integer by rounding down)

連結点において検出音が発音されている場合には，連結点以降のスペクトルに急激な変動が生じ，発音していない場合には，スペクトルの変動が比較的小さくなると考えられる．ここで，先に倍音の重なりを考慮しておく．前述の通り，楽音の倍音には重なりが生じる場合があり，検出対象音高を $N_{note}(i)$ とすると，$N_{note}(i)$ の 3 倍音および 6 倍音の周波数は $N_{note}(i)+7$ の 2 倍音および 4 倍音の周波数に重なるため，音符の連結処理の妨げとなる．したがって，$N_{note}(i)+7$ が同時に発音されている場合は，3 倍音と 6 倍音のスペクトル変動 $s_3(l_y), s_6(l_y)$ を使わずに検出を行う．同様に考え，1 オクターブの範囲内で倍音の重なるケースを表 1 に示す．これらの音高が含まれる場合，対応する倍音のスペクトル変動を用いないこととし，残った倍音のスペクトル変動のみを用いる．残った倍音のスペクトル変動 $s_r(l_y)$(r は残った倍音) において，次の処理を適用することで，接続・切断を判断する．

(1)　各 $s_r(l_y)$ の最大値が全て，l_y の長さの 1/3 以上 ($l_y \geq \frac{M}{3W_{sy}}$) に存在する場合，発音し直されたと判断し，切断として出力する．

(2)　(1) 以外で，1 つでも最大値が l_y の長さの 1/10 以内 ($l_y \leq \frac{M}{10W_{sy}}$) に存在する場合，波形は減衰しており発音はないと判断し，接続として出力する．

(3)　(1),(2) のどちらにも当てはまらない場合，不明として出力する．

接続として検出された場合，シーケンスを接続する必要がある．i 番目のシーケンスと $i+q$ 番目のシーケンスが接続される場合，代入 $V_{note}(i) \leftarrow V_{note}(i)+V_{note}(i+q)$ を行い，$V_{note}(i+q)$ を除去する．

表 1　倍音の重なり

$N(i)$ との音高差	重なる $N(i)$ の倍音
+7	3,6
+5	4
+4	5
−5	3,6
−7	2,4

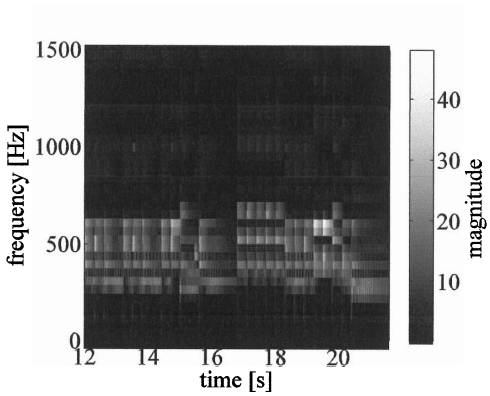

図 4　楽曲全体の STFT

6. 実　　験

3.1 で述べたように，MIDI で作成した 4 和音以内のピアノ楽曲を合計 6 種類用意し，それぞれサンプリング周波数 $f_s=44100$[Hz] で録音して入力信号とした．例として，楽曲 1 について結果を示していく．

6.1　onset 検出

楽曲 1 はテンポ $T=100$[bpm] であり，式 (1) よりビート長 L は式 (11) となる．

$$L = \frac{60}{T} = 0.6[\mathrm{s}] \tag{11}$$

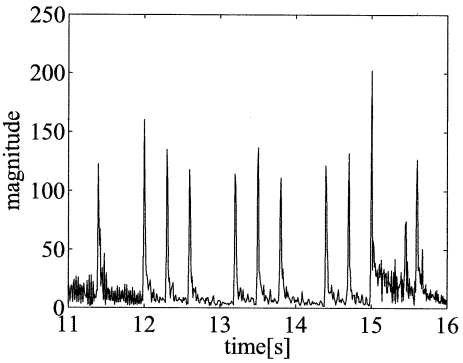

図 5　楽曲 1 における s_h

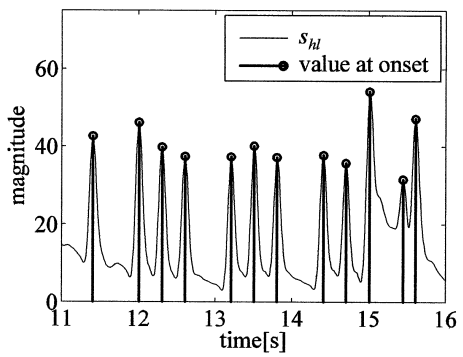

図 6　楽曲 1 における s_{hl} と onset

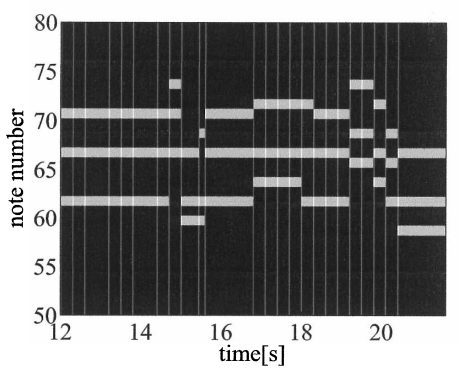

図 7 onset を重ねた楽曲 1 の音高推定結果

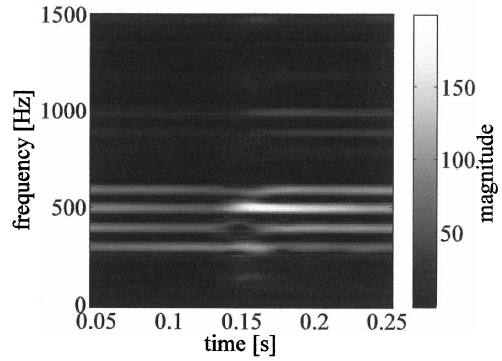

図 8 1 つ目の onset における $y(m)$ の STFT

$B_{note}(i)$	$N_{note}(i)$	$V_{note}(i)$
0	62.0000	0.5000
0	67.0000	0.5000
0	71.0000	0.5000
0.5000	62.0000	0.5000
0.5000	67.0000	0.5000
0.5000	71.0000	0.5000
1.0000	62.0000	1.0000
1.0000	67.0000	1.0000
1.0000	71.0000	1.0000
2.0000	62.0000	0.5000
2.0000	67.0000	0.5000
2.0000	71.0000	0.5000
2.5000	62.0000	0.5000
2.5000	67.0000	0.5000
2.5000	71.0000	0.5000
⋮	⋮	⋮

図 9 シーケンスの一部

表 2 各楽曲における正答率

	楽曲 1	楽曲 2	楽曲 3
検出箇所数	61	99	172
誤接続数	0	0	13
誤切断数	0	1	2
不明数	0	0	4
誤り総数	0	1	19
正答率	100 %	99.0	89.0
	楽曲 4	楽曲 5	楽曲 6
検出箇所数	51	45	242
誤接続数	2	2	2
誤切断数	0	3	15
不明数	0	0	10
誤り総数	2	5	27
正答率	96.1	88.9	88.8

楽曲信号 $x(n)$ に，窓幅 $W_{wx} = 1024$ のハミング窓をシフト幅 $W_{sx} = 400$ ずつシフトさせて STFT を適用する．図 4 は STFT 結果 $s(l_x, k_x)$ の一部であり，横軸は時間 [s]，縦軸は周波数 [Hz]，色でスペクトルの大きさを表している．時間と共に周波数スペクトルが変動していることが見て取れる．STFT 結果より，式 (4) を用いて $s_h(l_x)$ を求めると，図 5 のようになり，onset の位置に大きなピークを持つ波形となる．この波形を，式 (5) で得られるカットオフ周波数を持った LPF に通して $s_{hl}(l_x)$ を求め，5.2.1 で述べた手法によって極大値の存在するインデックス l_x から onset を検出する．図 6 に $s_{hl}(l_x)$ と onset の存在する極大値を重ねて示す．図 6 から，$s_{hl}(l_x)$ において大きなピークのみを検出していることがわかる．

6.2 onset で分割されたシーケンスの検出

図 2 の理想的な音高推定結果に，求めた onset を重ねて表示すると図 7 のようになる．縦軸と平行に引かれた直線が onset の存在する時刻であり，この onset ごとに音高推定結果を分割することで，分割されたシーケンス $B_{note}(i)$,$N_{note}(i)$,$V_{note}(i)$ を得ることになる．

6.3 音符の連結処理

次に，音符の連結処理を行う．楽曲 1 の入力波形 $x(n)$ の onset 周辺を抜き出した波形 $y(m)$ において，窓幅 $W_{wy} = 4096$ のハミング窓をシフト幅 $W_{sy} = 100$ ずつシフトさせて STFT を適用する．図 8 は，1 つ目の onset 周辺を抜き出した波形おける STFT 結果であり，同様に横軸に時間 [s]，縦軸に周波数 [Hz]，色でスペクトルの大きさを表している．このように，時間軸中央の onset 付近において，スペクトルに必ず大きな変動の生じる周波数が存在する．この STFT 結果から $s_r(l_y)$ を求め，5.2.3 で述べた条件を用いて音符の接続・切断を判断して得られたシーケンスの一部を図 9 に示す．また，得られたシーケンスから音符の連結処理の結果を，横軸拍，縦軸音高のピアノロール形式で図 10 に示す．灰色部分が音符の鳴り続けている箇所を示し，白色部分が音符の発音開始拍の位置を示している．この結果から，同じ onset において，接続されている音符と切断されている音符が存在していることがわかる．この連結処理の正答は図 11 に示すとおりであり，丸で囲った部分に誤りが生じている．

6.4 各楽曲での検討結果

各楽曲における連結処理の正答率を表 2 に示す．6 曲の正答率の平均は 93.6 %となり，誤りの少ない検出が可能であった．

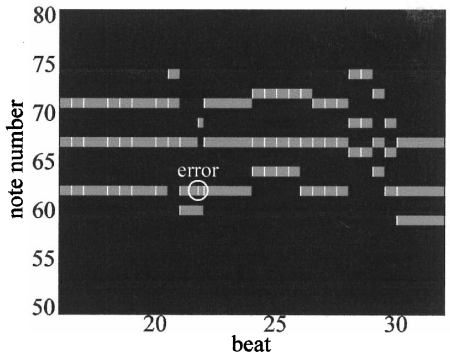

図 10 音符の連結処理結果

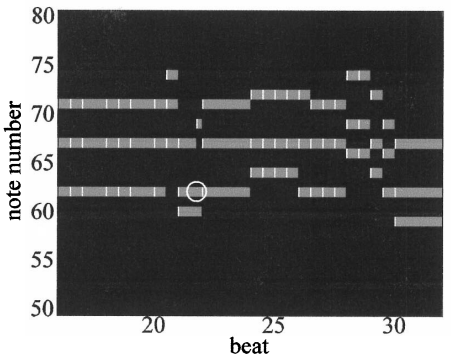

図 11 音符の連結処理の正答

6.5 考　　察

倍音の重なりを考慮した手法を用いない場合についても実験を行ったところ，音符の連結処理における正答率が約 80 %であった．このことから，倍音の重なりを考慮した手法を用いたことによって，正答率を向上させることができたといえる．

誤った切断については前述の通り，onset における調波構造の間のノイズによって，弾き直しをしていない楽音の周波数においてもスペクトルの急激な変動が生じ，切断として判断されていた．誤った接続については，検出音高の楽音が発音されているのにも関わらず，基本周波数あるいは倍音周波数においてスペクトルが増加せずに，逆に減少するといった想定外の現象が生じ，これにより接続として判断されていた．

今回の検出手法は，不明として判断された連結点が非常に少なく，ほぼ全ての連結点で接続あるいは切断として判断されていたことから，判断条件が緩かった可能性がある．よって，先に確実に接続・切断といえる箇所のみを判断しておき，不明と判断された残りの連結点については後処理で判断するといった手法を取り入れる必要があるといえる．

7. ま と め

自動採譜システムに必要な音価割当において，特に音符の連結処理を行う手法について提案した．提案手法は，倍音の重なりを考慮した上で周波数スペクトルの時間変動から検出する手法であり，提案手法を用いた実験の結果，オクターブの限定された 4 和音以下のピアノ曲において，平均 93.6 %の正答率で音符の連結処理が可能であることを確認した．

現在オクターブを限定しているために，倍音の重なりを考慮することで音符の連結処理が可能であるが，例えば，$N(i) - 12$ などの音高が存在した場合，全ての倍音が重なってしまうなどの問題が生じる．したがって今後の課題としては，この問題を解決するための手法を新たに取り入れる必要があるといえる．

なお，本研究の一部は，平成 20 年度科学研究補助金 (基盤研究 (C)19500082) により行われた．

参 考 文 献

1) A.P.Klapuri and M.Davy, “Signal Processing Methods for Music Transcription, ” Springer, 2006.
2) A.Sterian and G.H.Wakefield, “Music transcription systems: from sound to symbol, ” AAAI-2000 workshop on artificial intelligence and music, July. 2000.
3) 森田健夫，山口満，田所嘉昭，“並列構成くし形フィルタの出力値に注目した採譜のための音高推定法，” 信学論 (D-II)，vol.J87-D-II，no.12，pp.2271-2279，Dec. 2004.
4) 後藤真孝，“コンピュータと音楽の世界-基礎からフロンティアまで-:拍節認識（ビートトラッキング），” 共立出版，pp.100-116，1998.
5) 野口博司，“MID2TXT : 標準 MIDI ファイルをテキストに変換する (MS-DOS 汎用)，” インターネットホームページ:http://homepage1.nifty.com/nogue/