

多楽器音のベースギターパート自動採譜

2010SE232 竹内 雅浩 2010SE255 鶴飼 立樹

指導教員：後藤 邦夫

1 はじめに

近年は、以前よりもギターやベースギターなどの楽器が一般に普及しており、演奏者はそれぞれがバンドを組んでライブやセッションを行い、コミュニケーションを図っている。しかし、音楽に関する知識や音感が乏しい演奏者は、楽譜なしで原曲通りに演奏をする事が困難だが、人間が音楽を聴いてメロディ・リズム・楽器の音色などを把握するという一連の動作を情報処理によって実現する事ができれば、楽曲の採譜や演奏者同士のセッションはより快適なものになるだろう。ポピュラー音楽では、ベースギターが和音の基音となることが多いので、ベースギター音の把握は他の楽器の演奏者にとっても重要である。また、ベースギター音は他の楽器と比べて音高が低く、音符情報の把握は比較的容易である。よって、本研究では多楽器で演奏された楽曲からベースギターパートのみの楽譜を作成するシステムを開発する。

先行研究 [6] ではベースギター音だけの音源が対象であり、複数の楽器で演奏された通常の楽曲からベースギターパートの譜面を作成する事はできなかった。この問題点を解決するために、当研究ではフーリエ変換や自己相関関数の計算による周波数の特定、バンドパスフィルタなどを用いて多楽器音中のベースギターパートを解析し、取得した音符情報を譜面化する。この手法は、多楽器で演奏された楽曲からベースギターパートのみの楽譜をどれだけ正確に作成できるかどうかで評価される。評価実験で用いる楽曲は、多楽器音で演奏されたものでなければならない。音符情報の認識率を確認するために、楽曲・譜面が存在する楽曲か自身で演奏した楽曲を用いる。また、ステレオフォニック形式の音源を対象とする。全てのシステムは共同で設計した。鶴飼立樹は譜面化システムの構築を担当し、竹内雅浩が周波数解析システムの構築を担当した。

2 提案するシステム

本システムの処理の流れを図1に示す。ステレオ音源を入力としてBPMと調を検出し、高速フーリエ変換と逆高速フーリエ変換を用いてベースギター音のみを抽出する。多楽器音の音源では、自己相関関数を用いて楽曲の基準ピッチを検出すると数値に誤差が生じてしまう。これを避けるためにベースギター音のみの音源に変換し、複数の音階から楽曲の基準ピッチを検出する。その後、楽曲の基準ピッチを基に各音符の音高を同定する。さらに、各音符の発音開始時刻と発音終了時刻を検出して発音時間を決定する。最後に、これらの情報を使用してベースギターパートの楽譜を作成する。

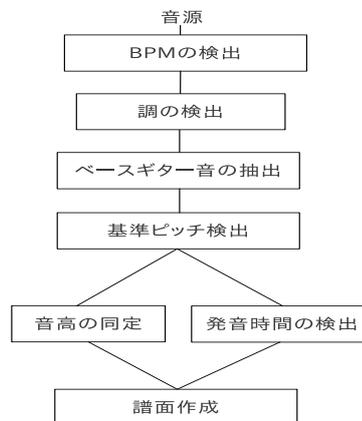


図1 本システムの処理の流れ

2.1 BPMの検出

譜面には演奏時の指針となる演奏のテンポが必要であり、これはBPM(Beats Per Minute)という数値で表記されることが多い。BPMは楽曲の1分間がある音符の何個分に相当するのかわを示す数値であり、1分間に4分音符が何拍あるかを示すのが最も一般的である。各音符の発音時間情報を譜面化するためには、楽曲で用いられているBPMが必要である。

参考文献[3]に記述されているBPMの自動検出方法には、音楽信号の所定の周波数成分の振幅のピーク間隔を計測してテンポを検出する方法や、所定の間隔で信号のレベル変化の相関を求め、相関関数が最大になる時間間隔を探索してテンポを検出する方法などがある。前者は、ビートの弱い楽曲や不規則な信号が含まれている楽曲のテンポを正確に検出できない。後者はテンポを正確に検出できるが、非常に大きな演算量が必要となる。

上記の課題を解決するために、当システムでは利用者が原曲から聴き取り体感した基準拍を利用者に入力させる。4分音符1つ分の発音時間を1回とし、原曲に合わせて任意の時間だけキーボード入力を行わせ、その回数を記録するプログラムを実装した。これにより読み取ったBPMを楽譜に表示する。また、楽譜の作成に必要な楽曲の拍子も当システムで利用者に入力させる。

2.2 調の推定

調とは、楽曲が中心としている音階のことである。調は楽曲の基となっているので、調が判明すると演奏者を支援でき、採譜時に調を定めると楽譜上のやの数を減らす事ができる。調の推定にはKeyFinder[2]を用いた。KeyFinderの処理手順の概要を以下に示す。

1. ステレオ音源をモノラル化する。
2. ローパスフィルタでサンプリングレートを減らす。

3. Constant Q Transform を用いて, 1 オクターブの各音符の音量と発音時間を示すクロマグラムを作成する.
4. 楽曲の和音進行や倍音から調が途中で変化するかどうかを判定する.
5. 各フレームで発音された音を音階の基音とし, 特に基音と基音から数えて三度離れた音高と五度離れた音高に着目して調と比較する.

KeyFinder を実行して出力される調名は, 1 から 12 までの数字 1 種類と A か B の文字 1 種類の組み合わせで表され, Disc Jockey が用いる Camelot Wheel という表に対応している.

2.3 高速フーリエ変換 (FFT)

解析法として先行研究 [6] では, ウェーブレット変換による解析が行われていたが, 本研究では対象となる楽曲は多楽器音であり, 処理に大幅な時間が掛かる. そこで, 本研究では高速フーリエ変換 (FFT) による解析を行う. FFT 実行前には, サウンドデータ切り出し時の波形の不連続性を最小限に抑えられ, 周波数特性が明確になるハニング窓処理を適用した. 処理は窓長の半分ずつ時間軸方向にずらしながら行った.

本研究で対象とするステレオフォニック方式の音源には, ステレオスピーカの出力音を左右に振り分ける PAN (Panoramic Potentiometer) が各楽器音に設定されている. ベースギター音は通常左右均等に出力され, 左右の位相と音量がほぼ等しくなる. よって, ベースギターで演奏できる基本周波数の範囲内で, 左右均等に出力されている音を抽出すれば良い. これは, ベースギターで演奏できる基本周波数の範囲外の音・左右の位相差の絶対値が閾値を上回る・PAN の差が振幅を上回る・左右の絶対値が近くないもののフーリエ係数を 0 にすることで実現する. 適用後の音源も左右均等の情報なので, 抽出後は左右どちらかの音のみを使用する.

2.4 基準ピッチの検出

楽器の調律や多楽器による演奏には基準となる音高が必要であり, エレキギターやエレキベースなどはこれらを基に調律されるため, 各音高の基本周波数を同定するためには楽曲の基準ピッチの検出が必要である. 多くの楽曲では, 基準ピッチが音高 A4=440Hz となっている. 周波数解析を行った音源を自己相関関数 (Autocorrelation Function (ACF)) を用いて基準ピッチ検出を行う. 波形を時間軸方向にずらして, 元の波形との一致度が高いところが波形の基準ピッチとなる.

なお, 最大値が 1 になるように正規化した自己相関関数の定義は次の通りである.

ここで, $x(n)$ は入力信号, $r(m)$ は自己相関関数, N は自己相関関数のサイズである.

$$r(m) = \frac{\sum_{n=0}^{N-m-1} x(n) \cdot x(n+m)}{\sum_{n=0}^{N-1} x(n) \cdot x(n)} \quad (1)$$

$$(0 \leq m \leq N-1)$$

自己相関関数の計算は, 入力信号とそれを m サンプルずらした信号を $N-m$ 個の区間に区切ってかけ合わせたものである. 信号に周期性がある場合, m を 1 周期の整数倍とすると, かけ合わされる信号が互いにほぼ等しくなるため, 自己相関関数の値は大きくなる.

しかし, それ以外の場合では, 自己相関関数の値は小さくなる.

2.5 音高の同定

音高の同定には, 1 オクターブの音程を均等な周波数で分割する音律である十二平均律を用いた. 十二平均律では, 隣り合う音の周波数比が等しく $2^{1/12} : 1$ となるため, 楽曲の基準ピッチに $2^{1/12}$ を音階の差分だけ掛けることで, 各音高の基本周波数を求めることができる. 抽出したベース音から推定した基準ピッチを 12 平均律の基準音 A4, 求めたい音程の基本周波数を X_n , 基準音と求めたい音高の音階の差を n とおいて, 以下の式で音高毎の基本周波数を導出する.

$$X_n = A4 \times 2^{\frac{n}{12}} \quad (2)$$

十二平均律を用いて音高 E0 から E3 全ての基本周波数を求め, 各音高の基本周波数の 100 分の 1 を誤差として認めてバンドパスフィルタ [7] を 3 オクターブ分の 36 種類作成する.

本研究で作成するバンドパスフィルタは, 各音高の基本周波数に対して通過域を設定した 36 種類のバンドパスフィルタを複合した楕円形のものである. 楽曲から抽出したベース音源にこれを適用する.

2.6 発音時間の検出

作成した楕円形フィルタでは, 二乗平均平方根 (Root Mean Square) の値を算出して音高と発音時間を判定した. RMS は, 発音された音声信号の音圧を各時間毎に自乗したものの平均値を算出し, その平均値の平方根を求めるものである. RMS の式は, 下記のように表現できる.

ここで, x_{rms} は二乗平均平方根, N は要素数, $x_1 \sim x_N$ は時間毎の発音信号の音圧を表す.

$$x_{rms} = \sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2} = \sqrt{\frac{x_1^2 + x_2^2 \cdots x_N^2}{N}} \quad (3)$$

バンドパスフィルタが適用された 36 種類の音声信号それぞれの RMS 値を算出し, 閾値を越えているものを発音された音高とみなす. 閾値を越えて音圧が極大値を示した時間が発音開始時間となり, 閾値を下回った時刻が発音終了時間となる. 発音時間はこれらの値から検出できる.

なお, フィルタが全く反応しない場合は, その時点で発音されていた音符がないと判断する. 以上の手法で楽曲中の全音符の音高と発音時間を同定する.

2.7 譜面作成

検出した BPM・音高・発音時間・調の情報を反映し, ベースギターパートの譜面を作成する. 検出した情報は,

音楽記述言語の一つである ABC 記譜法 [5] に基づき、下記のプログラムにより ABC 譜を表記する。表記したものを数多くの譜面作成ソフトウェアで読み込める MusicXML 方式のファイルに変換する。MusicXML 方式に変換するソフトウェアを用いてファイルを出力し、MuseScore[1] により譜面を作成する。

ABC 記譜法では、X はファイルナンバー、M は拍子数、L は 1 文字の音符長、K は調を表す。音高を示す文字列は、当システム実行時に入力する調名に応じて変換させる。1 オクターブ高い音は小文字で、さらにそれより高い音は右上に「^」を付け、1 オクターブ低い音は文字の直後に「,」を付けて表記する。休符は「z」、小節区切りは「|」、曲終わりは「:]」で表記する。

音長は、基準とする任意の音符によって変化し、基準の音符の何倍となっているのかを出力する。当システムでは 8 分音符を基準とした。音高を表す文字列の直後に連続して、音長が基準の何倍となるかを示す数字を記述することで、ABC 記譜法における音符となる。「C8」は全音符、「C6」は符点 2 分音符、「C4」は 2 分音符、「C3」は符点 4 分音符、「C2」は 4 分音符、「C3/2」符点 8 分音符、「C1」は 8 分音符、「C1/2」は 16 分音符を表す。

譜面は五線譜形式で出力し、ト音記号・拍子数・五線・音符と休符・BPM・調を表示する。BPM と拍子は、BPM 検出プログラムの実行結果を表示する。音符と休符は、同定した音程・推定した音長・BPM を表示する。調は、KeyFinder の実行結果に対応する調名を表示する。また、一般的にベースギターパートの譜面はヘ音記号で表記されるが、本研究では演奏経験や知識が乏しい演奏者への支援を目的としているので、ヘ音記号表記が読めない演奏者のためにト音記号で各音符を表記する。

3 実験結果

実験に使用する楽曲の種類は J-POP である。

3.1 BPM の検出

BPM 検出プログラムを 30 秒間実行し、楽曲の BPM を計測した。実験に使用した楽曲の BPM とプログラムの計測結果を表 1 に示す。

表 1 BPM の測定結果

アーティスト名	楽曲名	計測した BPM	楽曲の BPM
Janne Da Arc	Shinning Ray	159.727273	161
the brilliant green	Ash Like Snow	163.404255	166
RADWIMPS	トレモロ	189.068966	196
FLOW	GO!!!	134.4	135
B'z	it's show time!!	142.222222	147

検出した BPM と実際の BPM には約 2~7 の誤差が生じた。表 1 の 1 曲目のように、BPM が 161 の場合は 4 分音符 1 個分の発音時間は約 0.3726 秒である。検出結果のように、BPM が 159.727273 であれば 4 分音符 1 個分の発音時間が約 0.3756 秒となる。よって、上記の実験で使用した楽曲のようにテンポが早いでは、当実験で確認した程度の誤差は非常に小さいと言える。楽曲のテ

ンポが遅い場合は、プログラムの実行時間を長くして測定の精度を向上させる必要がある。

3.2 調の検出

KeyFinder を実行し、楽曲の調を検出した。実験に使用した楽曲の調と KeyFinder が検出した調を表 2 に示す。

表 2 調の検出結果

アーティスト名	楽曲名	検出した調	楽曲の調
ELLEGARDEN	スターフィッシュ	10B (D)	D
the brilliant green	Ash Like Snow	9A (Em)	G
FLOW	GO!!!	1B (B)	B
B'z	it's show time!!	9A (Em)	Am
L'Arc-en-Ciel	Driver's High	12B (E)	C#m

1 曲目と 3 曲目は、楽曲で用いられている調と同じ調を検出する事ができた。2 曲目と 5 曲目は、長調と短調を誤認識したが、楽曲で用いられる調と音が全く同じ調を検出した。4 曲目は調を正しく認識する事ができなかった。

3.3 基準ピッチの検出

多楽器で演奏された楽曲を対象にプログラムを実行した結果を表 3 に示す。

表 3 多楽器音の基準ピッチ検出結果

アーティスト名	楽曲名	検出した基準ピッチ	楽曲の基準ピッチ
ELLEGARDEN	スターフィッシュ	442 Hz	440 Hz
the brilliant green	Ash Like Snow	441 Hz	440 Hz
FLOW	GO!!!	441 Hz	440 Hz
B'z	it's show time	437 Hz	440 Hz
L'Arc-en-Ciel	Driver's High	441 Hz	440 Hz

多楽器で演奏された楽曲は 1~3Hz の誤差が生じた。

3.4 自動採譜した譜面

多楽器で演奏された音源のエレキベース音のみを自動で採譜した。当システムで作成した譜面 (図 2) と本来の譜面 (図 3) を次に示す。

楽曲は、the brilliant green の「Ash Like Snow」を使用し、0 秒から 15 秒までの譜面を作成した。

譜面作成プログラムで出力した ABC 譜面

```
X: 1
M: 4/4
L: 1/8
Q: bpm 1/4 = 166.000000
K: G
z8 | ^F,2 ^G,2 z2 ^g2 |
^g2 f2 ^c2 z2 | z2 z2 z2 f2 |
g2 c2 z2 e'2 | z2 z2 z2 ^f2 |
z2 ^G2 z2 z2 | ^d2 z2 z2 f2 |
z2 E,2 z2 E,2 | z2 z2 E,2 z2 |]
```

自動採譜で作成した譜面は本来の譜面とは大きく異なる内容であった。楽曲の第 1 小節と第 2 小節では、本来



図 2 Ash Like Snow の ABC 譜面を MuseScore で読み込んだ五線譜

はエレキギターのみしか発音されていないが、作成したベースギターの譜面では第 2 小節が発音されていたため、エレキギターの音を誤検出したと考えられる。また、発音時間が 16 分音符より短い場合や、発音開始時間が前の発音終了時間よりも早い場合は、発音していないと考えているため、本来発音されている箇所を消音と判定したと考えられる。作成したベースギターの譜面の第 5 小節の 2 つ目の音符はオクターブ違いの音高を検出したが、4 つめの音符は正しく認識できている。第 10 小節の音符はオクターブ違いの音高を検出した。音符情報の誤認識の原因は、エレキベース以外の音を排除しきれていない事とエレキベースの基本周波数ではなく倍音を検出した事だと考えられる。また、ベースギターは弦の状態やピッキングの強さによって、時間と共に第 2 倍音（オクターブ上）の音圧が基音の音圧よりも高くなる場合がある。オクターブ違いの誤認識はこれが原因だと考えられる。

3.5 自動採譜の評価

システムの実行結果は、[4] の式で音符情報の認識率を算出して評価した。\$Nn_t\$ を入力に含まれる総音符数、\$Nn_r\$ を出力に含まれる正しく認識された音符の数、\$Nn_w\$ を出力に含まれる音符の中で誤って認識されたか余分に出力された音符の数とする。

$$Rn = \left(\frac{Nn_r - Nn_w}{Nn_t} \cdot \frac{1}{2} + \frac{1}{2} \right) \times 100 \quad (4)$$

楽譜が存在する楽曲または自身で演奏した楽曲であれば、それらを参照にして認識率を確認する。3 曲の音源のそれぞれ 1 分間分だけを対象に、システムを実行して得られた譜面の音符情報の認識率を表 3 に示す。

表 4 音符情報の認識率

アーティスト	楽曲名	音高の認識率	音長の認識率
FLOW	GO!!!	8.33 %	10.41 %
the brilliant green	Ash Like Snow	13.63 %	21.81 %
the brilliant green	ENEMY	10.04 %	16.26 %

4 おわりに

本研究では、多楽器で演奏された楽曲からエレキベース音のみを抽出し、ベースギターパートの楽譜を自動作成するシステムの認識率を向上させるため、今後は、主に音高の同定と発音時間の検出システムを改良していきたい。今後の課題としては、基準ピッチ検出・音高の同定・発音時間の検出プログラムの改良である。

基準ピッチ検出システムは、多楽器音から抽出した連続的なベース音を対象としていたが、未だ精度が低い。実際の音源では、ベース音が連続して発音されていない場合が多い。また、多楽器音が完全に抽出しきれていない音源でベースギター音の基準ピッチを検出した事が原因だと考えられるので、多楽器音の抽出と発音時間の判定が課題である。

音高の同定システムは、本研究では楕円フィルタを用いて音高の同定を行ったが、精度が低いので、それぞれの音高の通過域に対する精度の向上が課題である。

発音時間の検出システムは、RMS の計算精度が低いため、計算精度を向上させ発音開始・終了時間をより正確に検出する必要がある。複数の音の発音が同時刻に開始された場合の判定なども課題である。

今後は、上記を中心にシステムを改良して音符情報の認識率を向上を目指していきたい。

参考文献

- [1] MuseScoreBVBA: *MuseScore*, accessed Jan.2014, <http://musescore.org/ja> (2011).
- [2] Sha'ath, I.: *Estimation of key in digital music recordings*, JISC Plagiarism Detection Service, accessed Jan.2014, <http://www.ibrahimshaath.co.uk/keyfinder/KeyFinder.pdf/> (2011).
- [3] Sugai, I.: *Tempo detection device and tempo detection program*, Pioneer Corporation, 特開 US 2011/0067555 A1, accessed Jan.2014, <http://www.google.com/patents/US20110067555/> (2011).
- [4] 中臺一博, 柏野邦夫, 田中英彦: 音楽音響信号を対象とする音源分離システム ~ 音モデルに基づくアプローチ ~, 情報処理学会研究報告 音楽情報科学 研究会, 93-MUS-1, accessed Jan.2014, http://lab.iisec.ac.jp/tanaka_lab/publications/kenkyukai.html#1993/ (1993).
- [5] 大黒 学: ABC 実習マニュアル, accessed Jan.2014, <http://tutorial.jp/music/abc/abcman.pdf> (2005).
- [6] 菊池淑晃, 後藤真孝, 村岡洋一: ベースギターの自動採譜システム, 情報処理学会全国大会講演論文集第 52 回平成 8 年前期 (1), 459-460, accessed Jan.2014, <http://ci.nii.ac.jp/naid/110002867064/> (1996).
- [7] 青木直史: C 言語ではじめる音のプログラミング- サウンドエフェクトの信号処理 -, オーム社 (2008).