

# 声部分離を用いたスコアリーディング支援のための 色付け楽譜の生成

石垣 麻子\*      松原 正樹†      斎藤 博昭†  
Asako Ishigaki   Masaki Matsubara   Hiroaki Saito

\* 慶應義塾大学理工学部 情報工学科

† 慶應義塾大学大学院 理工学研究科

E-mail: asako@nak.ics.keio.ac.jp

## 1 はじめに

ピアノ演奏において、複数の声部が重なり合うポリフォニー音楽に対し、声部ごとに表情付けを行うことが求められる。しかし初中級者には、このような音楽構造の読解と演奏を両立することが難しい。

また、音楽コーパスの大部分はポリフォニー音楽であるため、自動採譜や音楽情報検索等の精度向上のために声部分離や主旋律抽出の研究が近年多く行われている。

本稿では、音楽の初中級者にむけた音楽構造の提示のために、ピアノ楽曲の声部分離を行い色付け楽譜を生成するシステムを提案する。

## 2 関連研究

声部分離における課題は、楽曲内での声部数の変化に応じた声部の対応付けである。多くの研究は、楽曲を分割し主に音高、音長を基に声部分離を行った後、結合していくという処理を行っている。Chew らの手法 [1] では、楽曲を声部数の変化するタイミングで区切り、声部分離を行った上で隣り合う断片と次々に結合していく。他に、発音タイミングごとに音符をグループ化し、分枝限定法を用いる手法 [2]、決定木学習を用いて音符のペアごとに声部を求める手法 [3] が見られる。本研究では、声部数の増減に応じた楽曲の分割および結合を行う。

## 3 提案手法

本研究では、楽曲を声部数の変化するタイミングで区切ったものに対し、隣り合うものを全て結合するのではなく、声部数の増加時点で結合を行ったものを並べていくことで、楽曲全体の声部分離を完成させる。

そのため、音符情報を時間軸に沿って声部数が減少するまで読み込んで声部分離を行う、という処理を繰り返し行い、声部数が減少するたびに出力していく。

## 3.1 前処理部

まず、入力 MIDI ファイルから音符情報を読み込み、幅 0.125 拍 (=32 分音符) の窓で順次音符情報を調べ、各窓内に存在する音符を配列に格納する。各窓に含まれる音符数が、その時点での声部数を示す。

## 3.2 声部分離部

次に、窓の配列を順に調べ、声部数が減少するまでの声部分離の一時的な結果をフレーズのリストとして記憶する。ここでフレーズとは、同一の声部に含まれる連続した音符のまとまりとする。

まず、声部数が一定の間は、窓内の音符を音高順にフレーズに追加する。このとき、0.125 拍を超える音長の音符は複数の窓に重複して格納されているため、一度読み込んだ音符は無視する。また、各声部が単旋律となるよう、時間的に重なる音符が同一のフレーズに割り当てられないように調整する。

フレーズへの音符の追加を行うアルゴリズムを以下のように定義する。

- (1) 窓に含まれる音符  $N_0 \dots N_S$  を入力とする、このとき、その時点での声部数は  $S$  となる。
- (2) フレーズをそれぞれ  $P_0 \dots P_S$  とする。  $N \leftarrow 0$  とする。
- (3)  $N = S$  の場合、ステップ (10) に飛ぶ。
- (4)  $N_N \in P$  かつ窓内で  $N_N$  が一意である場合、ステップ (9) に飛ぶ。
- (5)  $i \leftarrow 0$  とする。
- (6)  $P_i = \emptyset$  の場合、 $P_i$  に  $N_N$  を追加し、ステップ (9) に飛ぶ。
- (7)  $P_i$  に  $N_N$  と時間が重複する音符がない場合、 $P_i$  に  $N_N$  を追加し、ステップ (9) に飛ぶ。
- (8)  $i \leftarrow i + 1$  とし、ステップ (6) に戻る。
- (9)  $N \leftarrow N + 1$  とし、ステップ (3) に戻る。
- (10) 窓内の全音符が、全フレーズの末尾へ全て振り分けられる。

声部数が増加する際は、それまでの声部に加えて新たな声部が現れることになる。そこで、音高の飛躍が最小となるよう考慮して、リストに空のフレーズを声部の増加数だけ加え、図 1 のようにフレーズと窓内の音符の対応付けを行う。

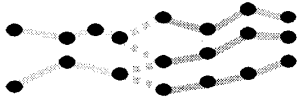


図 1: 声部数の増加と対応

本研究では声部の交差は想定しない。そのため、例えば 3 声から 4 声への増加時の、フレーズ  $P_0 \dots P_2$  と窓内の音符  $N_0 \dots N_3$  (いずれも音高順にソート済) の対応は表 1 のパターンが求められる。

表 1: 3 声から 4 声への増加時の対応パターン

窓内	pattern1	pattern2	pattern3	pattern4
$N_0$	$\emptyset$	$P_0$	$P_0$	$P_0$
$N_1$	$P_0$	$\emptyset$	$P_1$	$P_1$
$N_2$	$P_1$	$P_1$	$\emptyset$	$P_2$
$N_3$	$P_2$	$P_2$	$P_2$	$\emptyset$

ここで、例えばパターン 2 は、 $P_0$  と  $P_1$  の間に新たに空のフレーズを 1 つ挿入する場合を示している。

次に、パターンごとに、フレーズの末尾の音符から窓内の対応する音符への音高の飛躍距離の和を求める。ただし、窓内の音符とフレーズ末尾の音符が一致する場合の距離は  $-1$  とし、空のフレーズと窓内の音符との距離は  $0$  とする。

求められた飛躍距離の和が最小となるパターンに基づき、空のフレーズをリスト内の適切な位置に挿入し、窓内の音符を対応するフレーズの末尾に追加する。

声部数が減少する場合には、それまでに記憶したリストの中身を出力し、リストを空にする。出力の際に、各音符のチャンネル情報を書き換え、それまでの声部分離の結果を確定させる。

### 3.3 楽譜出力部

最後の拍まで声部分離が終了したら、声部数の減少時と同様にリストの中身を出力し、LilyPond<sup>1</sup>を用いて色分けされた楽譜を出力する。

## 4 実験

実験には、J.S. バッハによる Invention(2 声)15 曲 8264 音、Sinfonia(3 声)15 曲 9239 音の MIDI データ

<sup>1</sup><http://lilypond.org/>

を用い、全音符について、正しい声部に割り当てられた音符の割合を求めた。

声部分離の正解率を表 2 に、色分け楽譜の生成例を図 2 に示す。

表 2: 声部分離の結果

	Invention	Sinfonia	平均
本研究	99.02%	95.30%	97.16%
Chew ら	99.29%	93.35%	96.32%

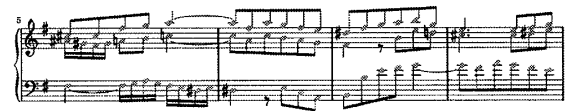
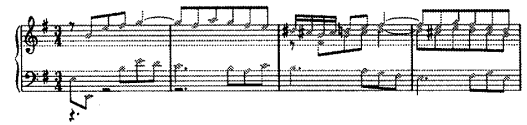


図 2: 色分け楽譜の出力例 (Sinfonia No.7 BWV793)

## 5 考察

声部分離では声部数が増えるほど精度は落ちるが、本研究において、3 声のポリフォニー音楽である Sinfonia に対しても 9 割を超える精度で声部分離を行うことができた。既存手法の中でも精度の高い Chew らの手法と比較しても、高い正解率となった。

本研究では音高情報を用いてに声部分離を行っているが、音高が等しい音符に対する処理を指定していない。また、実際の楽曲には稀に声部の交差や入れ替わりがあるため、今後は類似する旋律の検索等により声部の交差も考慮する必要がある。

また、本研究ではバッハの楽曲に対して高い精度を出したが、他の作曲家によるポリフォニー音楽に対しての精度を確かめるべきである。

## 参考文献

- [1] E. Chew and X. Wu :Separating voices in polyphonic music: A contig mapping approach, Computer Music Modeling and Retrieval, 2005.
- [2] S. Madsen and G. Widmer :Separating voices in MIDI, International Conference on Music Information Retrieval, 2005.
- [3] P. Kirilin and P. Utgoff :VoiSe: Learning to segregate voices in explicit and implicit polyphony, International Conference on Music Information Retrieval, 2005.