Vol.2010-MUS-86 No.27
2010/7/30

Colorscore：クラシック楽曲構造の可視化と圧縮表示

林亜紀† 伊藤貴之† 松原正樹††

クラシック楽曲において，とりわけオーケストラなどの大編成向けの楽譜（スコア）は，段数が多いため，特に初学者にとって，短時間で楽曲の音楽構造を理解するのは容易ではない．それ故，大編成の楽曲を扱う作曲者や編曲者，演奏者の多くは曲の全体像を短時間で捉えたいという要求や効率的に他の編成にアレンジしたいという要求を持っている．そこで本研究ではそのような要求を直感的に支援することができるような可視化手法と圧縮表示手法およびインタフェースを提案する．提案手法では，あらかじめ専門家によって与えられた主旋律や伴奏のパターンとのマッチングを行うことで，各パートの類似性やその役割の変遷に着目して楽曲の音楽構造を可視化する．また，提案するインタフェースでは，ユーザの操作に応じて縦横両方向の圧縮表示を行うことで，楽曲の中でより重要な部分を強調し，楽曲の理解だけでなくアレンジも支援する．

Colorscore - Visualization and Condensation of Structure of Classical Music -

Aki Hayashi† Takayuki Itoh† and Masaki Matsubara††

It is not always easy to quickly understand musical structure of orchestral scores for classical music, for the scores contain many staves of instruments. This paper presents Colorscore, a technique for visualization and condensation of musical scores. Colorscore supports two requirements for composers, arrangers and players: overview and arrangement. Colorscore divides each track of the score into note-blocks, and then determines their roles by matching their notes to patterns given by expert users. Colorscore then displays all the note-blocks in one display space to provide the overview, so that novice people can quickly understand the musical structures. In addition, our system provides an interface to support vertical condensation which reduces the number of displayed tracks, and horizontal condensation which saves the display space. It is especially useful as hints to rearrange music to smaller bands.

1. はじめに

1 つの楽曲，特にオーケストラ向けの大編成の楽曲を扱う際に，演奏者や編曲者，作曲者，鑑賞者の多くが，短時間で楽曲の全体像を把握したいという欲求や，楽曲を別のパート編成にアレンジしたいという欲求を持つ．楽曲の全体像とは，楽曲が持つ繰り返し構造や，どのパートがいつどのような主旋律や伴奏を担うかといった楽曲の音楽構造である．これらを把握することにより，作曲者の意図を反映した演奏や鑑賞が可能になるだけでなく，楽曲の要約や編曲を効率的に行うことができる．しかしながら，大編成の楽曲のスコア（総譜）の多くは，10 段を超える形式であり，ページ数も数十ページにわたるため，音楽構造を把握するためには経験が必要である．同様に編曲のために楽曲のパート編成を変える作業も初学者には困難である．そこで本研究では，スコアを読んだ経験のない初学者でも音楽構造理解を直感的に行うことができ，さらにアレンジ作業も支援することができるような支援ツールの開発を目的とする．

スコアを読む方法や，曲を通して聴く方法で楽曲の構造を把握するためには，多くの時間が必要である．その問題に対して我々は，初学者でも直感的に楽曲の構造理解ができる手段として，図 1 のように楽曲構造を一画面上に可視化する方法を提案する．

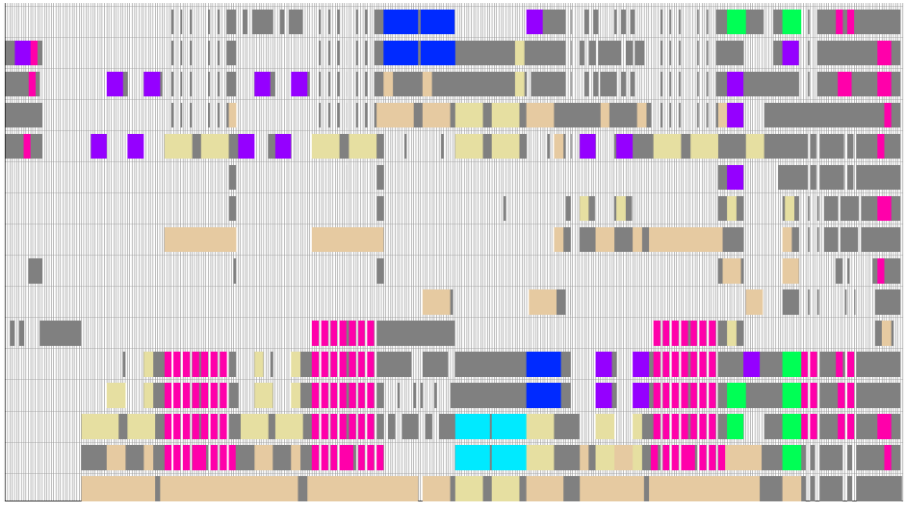


図 1　提案手法の可視化結果例

† お茶の水女子大学大学院 人間文化創成科学研究科
Graduate School of Humanities and Sciences, Ochanomizu University
†† 慶應義塾大学大学院 理工学研究科
Graduate School of Science and Technology, Keio University

提案手法ではパートを縦軸，時間を横軸として各パートが担う旋律を色分けして表示することで，楽曲構造の可視化を行う．この結果により，初学者でも直感的に短時間で楽曲の全体像を理解することができると期待される．

さらに，可視化結果の圧縮表示を行うことで，楽曲のアレンジ作業も支援する．ここでは，重要でない小節を縮めて表示する横方向の圧縮機能や，楽曲の骨格となるフレーズだけを残して段数を減らす縦方向の圧縮機能を実現する．これにより，初学者でも楽曲の要約や異なるパート編成への編曲ができると期待される．

なお，本研究の題名「Colorscore」は，Color Score（色つきのスコア）と Colors Core（音楽の中核部分に色をつける）を意味するものである．

2. 関連研究

2.1 音楽構造の可視化

本研究のような 1 楽曲を対象とした音楽構造の可視化や，楽譜情報の圧縮について，既に多くの手法が提案されている．

音楽構造の可視化について，パートには着目せず，全体の繰り返しに着目して楽曲構造を可視化する研究が多く行われている[1][2][3][4]．パートに着目しないため，MIDI だけでなく音響データにも適用できるという利点があるが，類似したメロディが複数のパートに受け継がれて繰り返し構造を成すことが多い大編成の楽曲では，より複雑な繰り返し構造の把握が必要な場合も多いことが懸念される．一方，パートに着目した可視化手法もいくつか研究されている．comp-i[5]ではパートごとに音程と音量，テンポに着目して三次元空間での楽曲構造の可視化を提案している．三次元空間を用いた可視化結果から楽曲構造を理解するのには慣れが必要だが，斬新な可視化結果で見る者を楽しませることができる．また，ScoreIlluminator[6]では，パート間のクラスタリング結果を反映して五線譜を色付けすることで音楽構造を可視化し，スコアの可読性を向上させている．BRASS[7]では，パートをまたいだリンク構造の可視化を実現している．本研究では，ScoreIlluminator や BRASS で着目しているようなパートをまたがったリンク構造の提示に加えて，それらの旋律が主旋律であるのか，伴奏であるのかといった情報も提示し，直感的に把握しやすい二次元で，一画面上に表示する．

楽譜情報を圧縮する手法も提案されている．BRASS では，手動操作による五線譜の圧縮・強調表示を行う．また，リズム特性の似ているパートをグループ化する方法でスコアを要約し，コンデンススコアを生成する手法[8]や，楽曲の中心となる旋律パートとベースパートの推定を行い，習熟度を考慮して演奏可能なピアノ楽譜を生成する手法[9]もある．本研究では，上述した音楽構造の分析結果を利用して，楽曲の骨格となる部分を選ぶ．これにより，音楽的な重要度に応じた自動的な縦横両方向の圧縮を実現する．加えて，ユーザの指定に応じて圧縮の強さを調節できる機能も実現する．

2.2 主旋律の抽出

本手法では，楽曲の音楽構造を可視化するために，各パートが担う役割の変遷を分析する．ここでいう役割とは数種類の主旋律や伴奏である．役割決定に際して，本手法では楽曲に固有の数種類の主旋律と伴奏のうち，基本的な音符列をあらかじめパターンとして手動で与えて，そのパターンとの類似度を算出する．

楽曲の主旋律を抽出する研究も提案されている．skyline アルゴリズム[10]が発表されているが，これは一番高い音を抜き出すという単純な手法であるため，オーケストラには適用できない．この問題を解決するために，どのパートにメロディが来やすいかをクラスタリングによって求めた後に主旋律を抽出する方法[11]が提案された．また，音域や隣り合う 2 音の音程差など複数の特徴量を考慮した Friberg らによる方法[12]も存在する．しかしながら，主旋律の抽出については，主旋律の長さや，何種類のフレーズを主旋律とみなすかが楽曲によって異なる上に，本手法ではこのようなパターンはユーザとして想定される初学者ではなく，経験者があらかじめ与えることを想定しているため，現段階では手動で与えたパターンを用いている．パターンを手動で与えることで，より初学者にも楽曲構造が理解しやすいような正当性の高い可視化結果が得られるだけでなく，可視化結果を見ながら与えるパターンを変更し，より有用な可視化結果を導くこともできると考える．

2.3 旋律パターンとの類似度算出

本手法ではあらかじめ付与したパターンと楽譜に書かれた旋律との類似度を求める．この類似度の計算についても，いくつかの既存の手法がある．Implication-Realization Model を用いる方法[13]や，GTTM[14]を用いる方法[15]，Earth Mover’s Distance を用いる方法[16]，Inner Metric Analysis を用いる方法[17]，また松原らが提案した発音のタイミングや音の響き方，音型と和声を考慮する方法[6]がある．本手法では，あらかじめ経験者が手動で与えたパターンとの類似度を求めるため，松原らの方法を参考に発音のタイミングと音型を比較して類似度を求める．類似度を求める際に，楽譜上のフレーズをどう区切るかという問題も存在するが，本手法ではパターンマッチによる類似度計算を行うため，合致したパターンの長さでフレーズを区切る．

3. 提案手法

本節では，提案手法について述べる．

3.1 可視化手法の概要

本手法では，楽曲データとして SMF(Standard MIDI file)を用いる．なお，提案手法の概要は以下のようになっている．

(1) 役割判定のためのパターン付与
(2) 初期ブロックの作成
(3) ブロックごとにパターンマッチを行い，役割を決定
　・パターンが見つかった場合には，そのパターンの前後でブロックを再分割
　・残りの部分の役割を再帰的に決定する
(4) 縦方向圧縮のためのデータ作成
(5) 横方向圧縮のためのデータ作成
(6) 分析結果の描画

3.2 役割判別のためのパターン付与

このフェーズでは，旋律や伴奏等フレーズの役割を区別するための基本的なパターンをシステムに与える．パターンとは 1 パートで構成される数小節単位の音符列である．本手法では例えば図 2 のように，可視化する楽曲に固有のパターンを，ユーザが MIDI 形式で与えることとする．役割とは，主旋律，伴奏（和音），伴奏（低音）などである．主旋律に関しては，可視化の対象楽曲中に出現する高々数種類のメロディに対して，最も基本的なものをパターンとして与える．伴奏に関しては，主旋律とは異なり典型的なリズムのみをパターンとして与え，音程に関する情報は与えない．これは，多くの楽曲において伴奏の特徴は，主として音程の動きよりも，反復的なリズムにあると考えるためである．

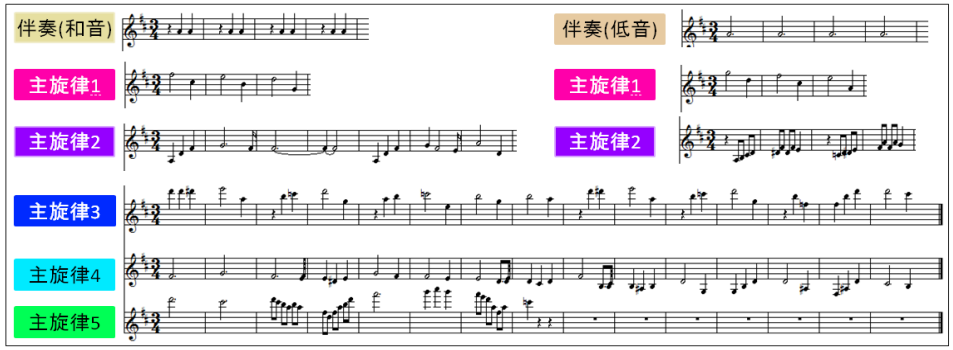


図 2　付与するパターンの例

3.3 初期ブロックの作成

続いて，初期ブロックを生成する．図 3 に初期ブロックの生成例を示す．以後登場するブロックという概念は，一般的にいうフレーズに該当するもので，最終的には同じトラックの連続した音符で構成される，同じ役割の音符群を示す．以下の手順で初期ブロックを生成する．

(1) 1 つのトラックを 1 つのブロックとみなす
(2) ブロック内の最初の小節から順番に全休符がある小節の前までを 1 つのブロックとする．また，再び音符が登場する小節以降を新しいブロックとする．
(3) ブロックに全休符が一つも含まれなくなるまで，再帰的に(2)を繰り返す．
(4) 全てのトラックに(2), (3)を適用する

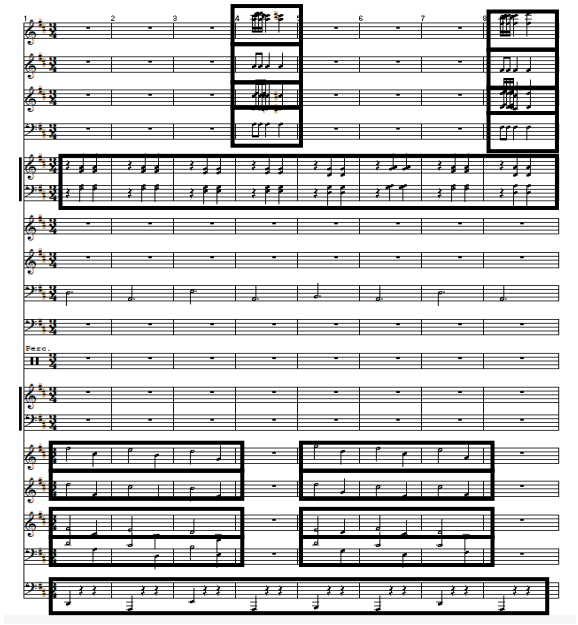


図 3　初期ブロックの生成例

初期ブロックの生成により，音符のない小節とのパターンマッチ（3.4 参照）を防ぐことができる．なお，各ブロックの情報として，トラック番号，ブロックの開始小節と拍，終了小節と拍を格納する．

3.4 パターンマッチによる役割の決定

各パートが担う役割を決定するために, 本手法では N 個のブロックと M 個のパターンの任意の組み合わせについて，距離を算出する．具体的には，小節数 L_j を有する j 番目のパターンに対して，小節数 L_i を有する i 番目のブロックの k 小節目から $(k+L_j-1)$小節目までの部分との距離 $D(i,j,k)$を算出する．このとき，距離を近いほど類似度が高いとし，パターンとブロックが合致すると判断するためのしきい値 S を設ける．役割判定の流れ図を図 4 に示すとともに，手順の説明も示す．

Vol.2010-MUS-86 No.27
2010/7/30

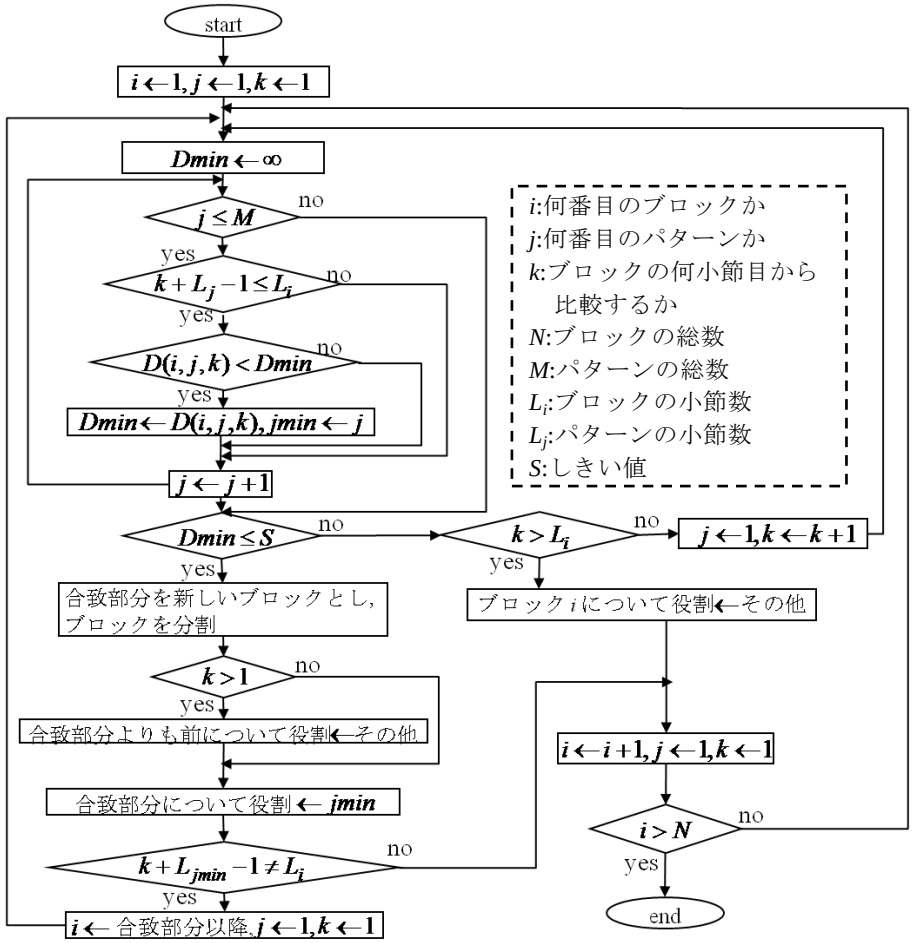


図 4　パターンマッチによる役割の判定

(1) $i = 1,\ k = 1$とする．
(2) M 個の全てのパターン$(1 \le j \le M)$ について距離 $D(i,j,k)$を求める．
(3) ブロック i との距離が最小となるパターンを $jmin$, その時の距離を $Dmin$ とする．
ただし，$(k + L_j - 1) > L_i$となるパターンについては考慮しない．

(3-1) $Dmin \le S$であれば，ブロック i のパターン $jmin$ と合致した部分を新しいブロックとして，ブロックの分割を行う．

- $k = 1$の場合 2 つのブロックに分割される．1 つ目のブロックの役割を $jmin$ とする．2 つ目のブロックについて(2)(3)を適用する．
- $k > 1$の場合は 3 つのブロックに分割される．1 つめのブロックの役割を「その他（装飾）」と定義し，2 つ目のブロックの役割を $jmin$ とする．3 つ目のブロックについて(2)(3)を適用する．
- $(k + L_{jmin} - 1) = L_i$であるとき，つまりブロック i の最終小節まで役割が決定した場合には，$i = i + 1,\ j = 1, k = 1$として(2)に戻る．

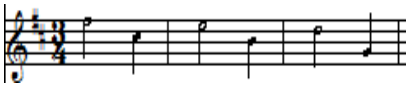
(3-2) $Dmin > S$であれば，ブロック i の途中から合致するパターンが見つかるまで $k = k + 1$として処理を反復する．ただし$k > L_i$となったら，ブロック i 全体の役割を「その他（装飾）」と定義して終了．$i = i + 1,\ j = 1, k = 1$として(2)に戻る．

(4)　全てのパターン i について役割判定を完了したら反復処理を終了

以上の処理を終えた段階で，ブロックの情報にそのブロックの役割と，そのパターンへの距離を加える．

なお，パターン i とブロック j の距離は，文献[6]の RhythmicActivity と MelodicActivity を用いコサイン類似度によって求めた．

RA ベクトル，*MA* ベクトルの例を図 5 に示す．*RA* ベクトル，*MA* ベクトルはそれぞれタイミング，音程遷移に関する特徴量である．



RA = [3 0 2　　3 0 2　　3 0 2]
MA =　[0 -5　　3 0 -5　　3 0 -7]

図 5　*RA*, *MA* の作成例

3.5　縦方向圧縮のためのデータ作成

続いて，縦方向圧縮のためのデータを作成する．縦方向圧縮表示では，16 段のスコアを 1 段，2 段，4 段，6 段の 4 種類の構成に圧縮する．できあがった圧縮結果をコンデンススコアと呼ぶ．例えば，1 段のコンデンススコアを作成するにあたっては，以下の手順で描画するブロックを決定する．

(1) $i = 1$ とする．
(2) 全チャンネルを対象に，i 小節目から始まるブロックを探す．
(3) ブロックが見つかった場合：
(3a) 見つかったブロックの中に主旋律がある場合：パターンとの距離が最も近い

ブロックを当てはめ，i = (当てはめたブロックの終了小節 + 1)として(2)に戻る．

(3b) 見つかったブロックの中に主旋律がない場合：$i = i + 1$として(2)に戻る．

(4) (2)(3)を最終小節まで終えた段階で，まだブロックが配置されていない小節があれば，そこに収まるような伴奏ブロックを(2)(3)と同様の手順で当てはめる．

(5) (4)を最終小節まで終えた段階で，まだブロックが配置されていない小節があれば，そこに収まるような「その他（装飾）」ブロックを同様の手順で当てはめる．

n 段のコンデンススコアを作る際には，元のスコアが N 段であった場合に，これを区切る$(n-1)$個の境界$\{M_1, ..., M_{n-1}\}$ $(M_i < M_{i+1})$を設け，1 段目から M_1 段目までをコンデンススコアの 1 段目，(M_1+1)段目から M_2 段目までをコンデンススコアの 2 段目…とし，最後に$(M_{n-1}+1)$段目から N 段目までをコンデンススコアの n 段目にする，というようにしてコンデンススコアの各段を生成する．

ブロックの情報に，4 種類のコンデンススコアを作る際，それぞれ何段目に表示するか，あるいは表示しないか，といった内容を加える．

3.6 横方向圧縮のためのデータ作成

横方向の圧縮表示では，より簡潔に音楽構造を可視化するために，各パートやその役割が前の小節から変化しているか否かに着目し，圧縮表示と強調表示の 2 種類を行う．圧縮表示では変化のなかった小節を圧縮して表示することで，全体の長さを短く描画する．強調表示では，変化のあった小節を引き延ばして表示することで，その小節を強調する．このフェーズでは，横方向圧縮表示を行うために，各小節について前の小節から登場するパートやその役割が変化しているか否かをデータとして格納する．

3.7 分析結果の描画

以上の分析結果をインタフェース上で描画し，音楽構造を可視化する．縦方向をパート，横方向を時間（小節），色を役割として，各ブロックを描く．この後，インタフェースを通じたユーザの操作により，縦横両方向の圧縮や，楽曲の再生などの機能を提供する．これらの操作については，4 節で詳細を述べる．

4. ユーザインタフェース

3 章の役割分析結果および圧縮結果を対話的に表示するために，Colorscore のインタフェースを実装した．図 6 に表示画面例を示す．ユーザは GUI によるインタラクション操作を通じて，表示された可視化結果に，縦横両方向の圧縮表示をはじめとする様々な操作を加えることができるとともに，楽曲の再生も行うことができる．

インタフェースの機能は結果の描画と操作，楽曲の再生，横方向圧縮表示，縦方向の圧縮表示の 4 つに大別できる．各操作の説明を以下に示す．

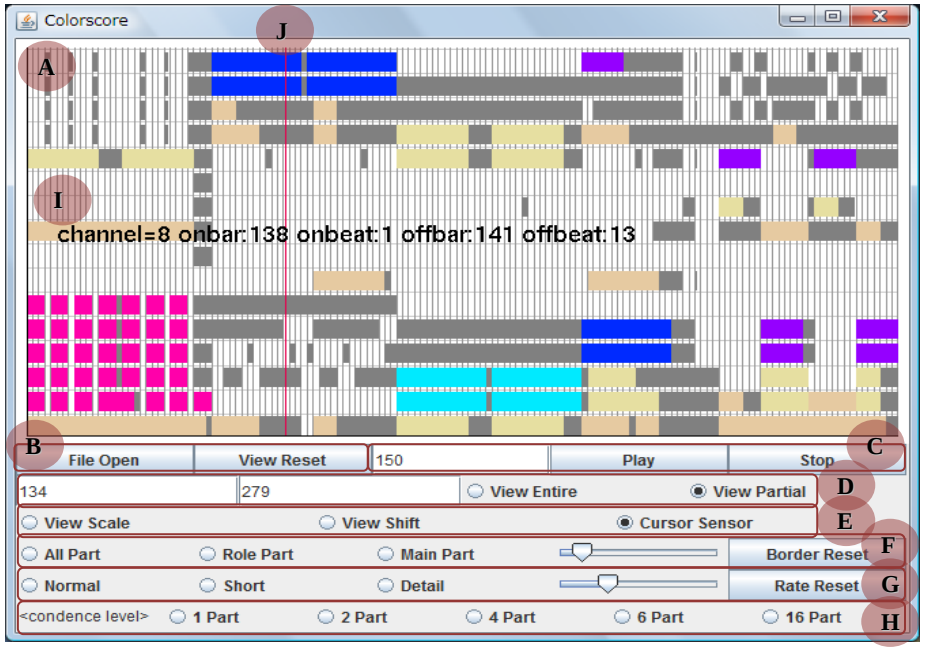


図 6　Colorscore の表示画面例

【結果の描画と操作】

最初に，図 6 Ⓑで楽曲の選択を行い，楽曲全体の分析結果を図 6 Ⓐに描画する．ユーザは可視化結果の拡大や縮小(図 6 Ⓔの View Scale)，平行移動(図 6Ⓔの View Shift を行う．図 6 Ⓓのテキストフィールドで開始小節と終了小節を指定して図 6 Ⓓの View Partial を選択すると，指定小節のみを描画することができる．また，図 6 Ⓔの Cursor Sensor を選択にすると，カーソルをあてたブロックのトラック名や開始位置，終了位置を文字情報として表示する（図 6 ①）．また，図 6 Ⓕで Role Part を選択すると，表示小節内で装飾以外の役割を持つブロックだけを表示し，描画できるブロックが存在しないパートは段全体を非表示にする．さらに，図 6 Ⓕで Main Part を選択すると，装飾以外のブロックで，パターンとの類似度が特に高かったブロックだけを選択して表示する．この時，表示するブロックを選ぶためのしきい値は，ユーザが図 6 Ⓕのスライダ操作で調整することができる．

【楽曲の再生】

図 6 Ⓒの Play ボタンを押すと，楽曲を再生する．表示されている最初の小節また

は，図 6 ⓒのテキストフィールドでユーザが指定した場合には，指定した小節から再生を行う．再生中は，ユーザの指定した開始小節が表示されていない場合を除き，可視化結果上のどこを再生しているかが，赤いバーによって表示される（図 6 ⓙ）ため，スコアを読めない初学者でも，可視化結果の各小節がどのように演奏されるかを把握することができる．

【横方向の圧縮表示】

図 6 ⓖにて横方向の圧縮表示を適用する．ユーザは前小節と比べて役割に変化のなかった小節を縮めて描くことで必要なスペースを削減する圧縮表示（図 6 ⓖの Short）と，逆に変化のあった小節を拡大して描くことで変化のあった小節を強調する強調表示（図 6 ⓖの Detail）のどちらかを選択できる．この時，圧縮または強調の度合いをユーザが図 6 ⓖのスライダ操作で調整することができる．また，例えば楽曲の最初の方，終わりの方，開始から半分位の位置であるといった楽曲全体を考えた時のおおよその位置情報を圧縮によって損なわないために，楽曲全体の小節数に対して 1/4，2/4，3/4 の位置の小節線をハイライトする．

【縦方向の圧縮表示】

図 6 ⓗにて 1 段，2 段，4 段，6 段のコンデンススコアをユーザの選択に応じて表示する．

5. 実行結果

3 節の役割分析を GCC3.4 で，描画と圧縮表示を Java JDK1.6.0 で実装した．本節では，チャイコフスキーの「花のワルツ」に Colorscore を適用した結果を考察する．

5.1 音楽構造の可視化結果

前述した図 1 は「花のワルツ」全体を可視化した結果である．桃色・紫色・青色・水色・黄緑色はそれぞれ図 2 に示したパターンの主旋律 1-5 に対応し，黄色と橙色はそれぞれ図 2 の伴奏（和音），伴奏（その他）に対応する．また，灰色は主旋律や伴奏には該当しないが，その他（装飾）に分類される音符があることを示す．ここで主旋律 1 と 2 に付与したパターンが複数存在するのは，一連の主旋律が途中で別パートに移動する場合や，途中に全休符が入り，ブロックが分かれてしまう場合が見受けられたためである．

この可視化結果により，例えばこの楽曲をソナタ形式に見立てれば，図 10 の 赤色の線(1)までが序奏，(5)までが提示部，(8)までが展開部，(10)までが再現部，それ以降がコーダであるといった楽曲の音楽構造を把握することができる．

また，ユーザが類似度のしきい値を変えることで，よりパターンとの類似度が高かったブロックのみを表示する機能を用いると，基本的な旋律から変奏されたフレーズや，いわゆるハモリのようなフレーズを消すことができる．図 7(左)は「花のワルツ」の 314-328 小節をそのまま可視化したものである．この可視化結果では，紫の主旋律と緑の主旋律のどちらかしか演奏できない編成にアレンジする時に，どちらを優先すればよいかが分からない．そこで，描画するブロックを選ぶための類似度のしきい値を上げていくと，図 7(右)が得られた．これにより，緑の主旋律の方が変奏されていない重要な主旋律であることが分かる．

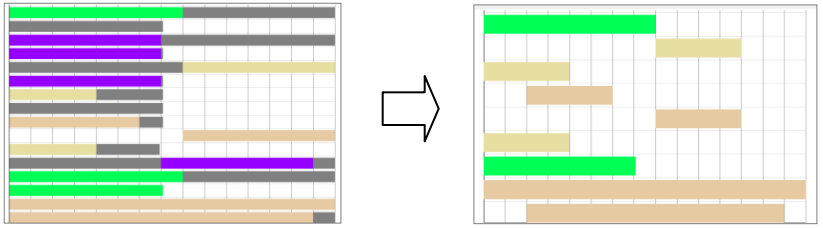


図 7　(左)元の可視化結果 (右)類似度の高い主旋律だけを描画した結果

5.2 横方向の圧縮表示結果

図 8(左)は「花のワルツ」の 211-250 小節の可視化結果である．これに横方向の圧縮を適用した結果が図 8 (右)である．横方向の長さを約 60%縮めることができたが，丸で囲まれているような，新しくパートが増える箇所などは損なわれていない．このような箇所は楽曲をアレンジする際に重要になってくるだけでなく，指揮を学ぶものが新しく登場するパートに合図を行うなどの場面で注目すべき箇所となっている．

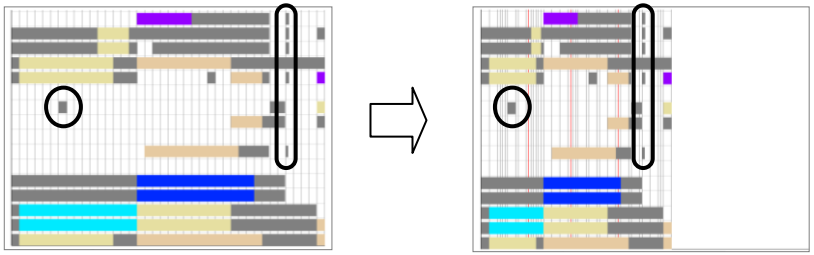


図 8　(左)圧縮前の可視化結果 (右)横方向の圧縮結果

5.3 縦方向の圧縮表示結果

図 9(上)は「花のワルツ」全体を 1 段に要約した例である．パターンとの類似度が高い主旋律のみを残して，1 段に要約されている．図 9(下)は，6 段のコンデンススコアである．元のスコアでの上下関係を損なわずに数段の情報を 1 段に要約することで，管楽器同士，弦楽器同士を近くに保ったまま楽譜情報の要約ができている．各ブロックをクリックすると元のトラックが表示されるので，楽曲の編成を変える際に，どのフレーズをどのパートに割り当てるかを考える作業を直感的に支援することができる．

Vol.2010-MUS-86 No.27
2010/7/30

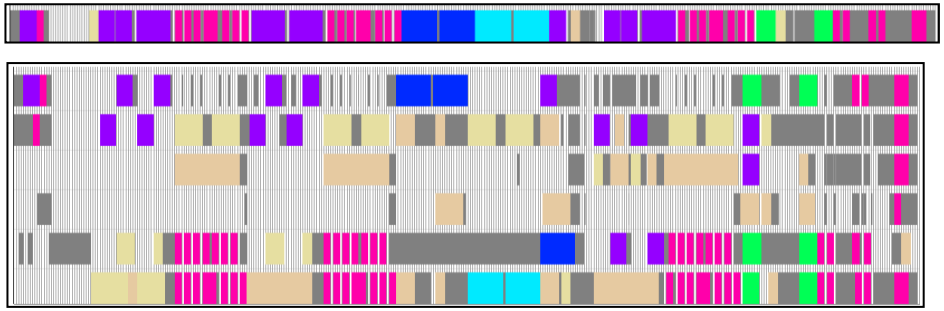


図 9　(上) 1 段のコンデンススコア，(下) 6 段のコンデンススコア

6. 評価と考察

本手法を評価するために，初学者を被験者とした評価実験と，音楽情報の可視化に精通した研究者からのコメント聴取を実施した．本節では評価実験の結果と考察をまとめるとともに，インタフェースの有用性についても考察する．

6.1 初学者への評価実験

初学者が本研究の出力結果からどの程度音楽構造を把握できるかを知るために，「花のワルツ」全体の可視化結果を提示し，音楽構造にしたがって楽曲を 10～20 個程度に分ける縦線を引くよう依頼した．被験者は学生 27 名で，そのうちスコアを読んだ経験はないが音楽経験のある 19 名をグループ A，音楽経験のない 8 名をグループ B とした．図 10 にて赤線で示されている 12 本を正答として，適合率や再現率を求めたところ，表 1 のようになった．なお，3 名の著者のうち 2 名以上が線を引いた箇所を正答とした．

表 1　初学者への評価実験結果

	A グループ	B グループ	グループ A+B
総本数	11.83	12.11	11.93
適合本数	9.44	8.44	9.11
適合率	0.81	0.70	0.77
再現率	0.79	0.70	0.76

スコアを読むことができない被験者でも比較的高い正答率で楽曲の構造を理解できたことが伺えるが，ここでは音楽経験の有無によって正答率に差が出た原因について考察することで，提案手法の改善点を提示したい．

経験者と未経験者で正答率に大きく差がでたのは，(5), (8), (10), (12)で，典型的な誤答例が青色の(a)-(c)である．未経験者に最も多かった誤答として，(5), (10)を(b)と回答する傾向があった．経験者は次の旋律の開始に注目したため，このような誤答は目立たなかったが，パターンマッチによる自動彩色を行ったがゆえに，15 段目だけ(b)の直後に色づけられてしまっていることが誤答を招いたと原因であると考えられる．ここでは 15 段目だけが特殊な動きをしているという訳ではないが，与えたパターンと 15 段目だけリズムが似ていたため，このような可視化結果になっている．そこで，今後はより正当性の高い結果を出せるようアルゴリズムの修正を行いたい．

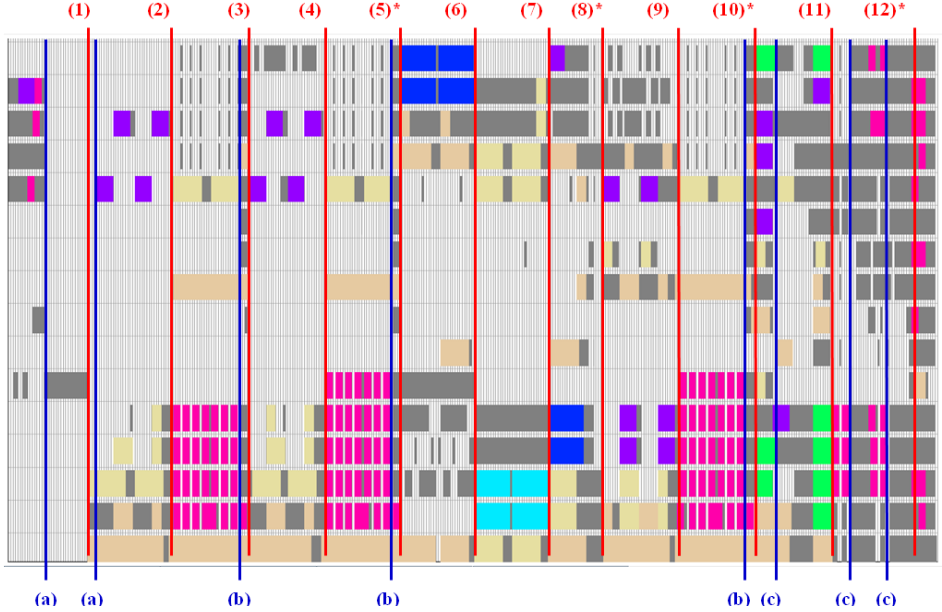


図 10　初学者への評価実験の正答例と誤答例

6.2 音楽情報可視化に精通した研究者からのコメント聴取

可視化結果から発見できること，実用例，考えられる課題などについて，参考文献[5][7][8]の著者へのコメント聴取を行った．3 名はともに，音楽情報可視化に精通した研究者であるだけでなく，アマチュアオーケストラでの演奏経験を持っている．以下は，コメントの要約である．

【可視化結果から発見できること】

- 例えば桃色で描画された主旋律が 3 回繰り返され，コーダでアレンジされることなどの，繰り返し構造
- 例えば桃色の主旋律は主に弦楽器に演奏される，コントラバスとチューバは一度も主旋律を担当しない，トロンボーンと打楽器は休みが多いなどの特徴

・自分のパートが全体に対してどういう役割なのか．例えばホルンが主旋律の時は中低弦が伴奏をしている，和音を演奏している時は弦楽器が主旋律を担い，合間に木管の装飾が入ってくるなど
・楽曲後半に向けて楽器が増え，音楽が盛り上がっていく様子

【実用例】
・音楽プレーヤーやデジタルスコアソフトでのディスプレイ
・音楽構造理解の教育目的での利用
・舞台演出・カメラワーク・振り付けなどの考案支援

【今後の課題】
・楽譜の段数に対して，小節数の方が圧倒的に多い場合の表示方法の工夫
・結果の有用性が付与する役割パターンに大きく影響されるという問題の解決
・特定位置の和声を再生する機能
・特定箇所を指定すると，類似した別のフレーズにジャンプする機能
・楽曲の最も重要な部分を示す機能
・圧縮をスムーズに実現するアニメーション機能

可視化結果の有用性について，著者らが目指したことが実現されていることが分かった．本研究の実用例や今後の課題についても実装を検討したい．

6.3 インタフェースの有用性に関する考察

本インタフェースの目的は，スコアから音楽構造を把握するのが困難な初学者でも，音楽構造の把握や楽曲のアレンジを可能にするような支援を行うことであった．実装されたインタフェースでは楽曲構造の可視化を行うだけでなく，可視化結果をユーザがインタラクティブに操作することができる．これにより，例えば可視化結果を見ながら，表示するブロックをよりパターンとの類似度が高いものに制限する機能を適用することで，ユーザは楽曲構造の理解ができるだけでなく，新たなパターンを思いつく場合もある．同様にして，縦方向の圧縮結果を見ることで，新たなパターンに気づく場合も考えられる．このようにして，可視化結果を操作しながら音楽構造を理解するだけでなく，繰り返してシステムを使い続けることによって，新たな発見を促すことができる可能性を本インタフェースは秘めていると考える．

7. まとめと今後の課題

本研究では，クラシック楽曲の音楽構造可視化手法と，縦横 2 方向の圧縮表示手法およびインタフェースを提案した．今後は引き続き，より効果的な分析手法や可視化方法を考案したい．また，役割パターンを手動で与えずに有用な結果が出せるような工夫もしたい．そして，他の楽曲への適用も行いたい．さらに，メロディの類似構造を可視化するだけでなく，類似したメロディ同士の細かい違いを，各メロディに対応するブロックに模様をつけることで可視化する機能も考案していきたい．

参考文献

1) J. Foote, Visualizing Music and Audio using Self-Similarity, 7th ACM international conference on Multimedia, Part 1, pp. 77–80 (1999).
2) M. Goto, SmartMusicKIOSK: Music Listening Station with Chorus-Search Function, 16th Annual ACM Symposium on User Interface Software and Technology, pp. 31–40 (2003).
3) M. Wattenberg, Arc diagrams: Visualizing structure in strings, Proc. IEEE Symposium on Information Visualization 2002, pp. 110–116 (2002).
4) C. S. Sapp, Harmonic visualizations of tonal music, Proc. International Computer Music Conference (ICMC, Havana, Cuba 2001), pp. 423–430 (2001).
5) R. Miyazaki, I. Fujishiro, and R. Hiraga, comp-i: A System for Visual Exploration of MIDI Datasets, Transactions of Information Processing Society of Japan, vol. 45, no. 3, pp. 739–742 (2004).
6) 松原正樹，岡本紘幸，佐野智久，鈴木宏哉，延澤志保，斎藤博昭, ScoreIlluminator：スコア色付けによるオーケストラスコアリーディング支援システム，情報処理学会論文誌,vol. 50, no. 12, pp. 1–12 (2009).
7) 渡邉ふみ子，藤代一成，平賀瑠美，デジタルスコアによる楽曲学習支援インタフェース，情報処理学会論文誌, vol. 45, no. 3, pp. 710–718 (2004).
8) 遠山紀子，松原正樹，斉藤博昭，リズム特性を用いたコンデンススコア自動生成手法の提案，情報処理学会科学研究会研究報告, 2007-MUS-69-10, Vol.2007, No.15, pp. 41–44 (2007).
9) 藤田顕次，大野博之，稲積宏誠，習熟度を考慮した複数楽譜からのピアノ譜生成手法の提案，情報処理学会音楽情報科学研究会研究報告, 2008-MUS-77-10, Vol.2008, No.89, pp. 47–52 (2008)
10) A. Uitdenbogerd, and J. Zobel, Melodic matching techniques for large music databases, 7th ACM international conference on Multimedia, Part 1, pp. 57–66 (1999).
11) G. Ozcan, C. Isikhan, A. Alpkocak, Melody extraction on MIDI music files, Proc. 7th IEEE International Symposium on Multimedia table of contents, pp. 414–422 (2005).
12) A. Friberg, and S. Ahlbäck, Recognition of the Main Melody in a Polyphonic Symbolic Score using Perceptual Knowledge, Journal of New Music Research, 38:2, pp. 155–169 (2009).
13) E. Narmour, The Analysis and cognition of basic melodic structures : the implication-realization model, University of Chikago Press (1990).
14) F. Lerdahl, and R. Jackendoff, A Generative Theory of Tonal Music, MIT Press (1983).
15) M. Hamanaka, K. Hirata, and S. Tojo, Melody Morphing method based on GTTM, ICMC2008, pp. 155–158 (2008).
16) R. Typke, R. Giannopoulos, C.R.Veltamp, F. Wiering, and R. Oostrum, Using Transportation Distances for Measuring Melodic Similarity, 4th ISMIR, pp.107–114 (2003).
17) A. Colk, J. Garbers, P. Kranenburg, F. Wiering, C.R. Veltamp and P.L. Grijp, Applying Rhythmic Similarity Based on Inner Metric Analysis to Folksong Research, 8th ISMIR, pp. 293–296 (2007).