社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2006－MUS－68 (12)
2006／12／16

事例に基づく演奏表情生成システムにおける MusicXML からの旋律断片自動生成

清水 厚志[†] 原田 杏奈[†] 鈴木 泰山[††] 野池 賢二[†††] 金子 雄介[††††] 徳永 幸生[†]

我々は，事例に基づく演奏表情生成手法を用いた演奏表情生成手法の研究を行っている．事例に基づく演奏表情生成システムでは，楽譜の類似度を評価するために楽譜を旋律断片に分割する必要がある．そこで，本報告では，MusicXML によって記述された楽譜の小節構造を解析し旋律断片を自動生成するアルゴリズムの設計と実装方法について述べる．また，本アルゴリズムと従来のアルゴリズムの比較実験を行い，本アルゴリズムの有効性を検証した．

Automatic Music Piece Segmentation from MusicXML in a Case-Based Performance Rendering System

Atsushi SHIMIZU[†] Anna HARADA[†] Taizan SUZUKI[††]
Kenzi NOIKE[†††] Yusuke KANEKO[††††] Yukio TOKUNAGA[†]

We have been developing a case based approach to the generation of musical expression. A case based performance rendering system needs segmentation of scores to evaluate similarity of scores. This report describes algorithm design and implementation which analyzes structure of measures of a score written in MusicXML and generates musical pieces automatically. We compared this algorithm and conventional algorithm, and checked the validity of this algorithm.

1. はじめに

計算機による情緒あふれる演奏の自動生成は音楽情報科学の主要なテーマであり，表情付き演奏の自動生成システムが 1980 年代後半から多数発表されている．演奏表情生成手法は，規則に基づく手法と事例に基づく手法に大別される．事例に基づく演奏表情の自動生成では，あらかじめ用意された演奏データ集の中から対象曲に類似した楽曲を検索し，それらに見られる演奏表情を対象曲に適用することで演奏データを生成する．演奏表情を適用する際には，検索した演奏事例の中から対象曲に最も類似する事例を選択し，対象曲に適用する必要がある．現在我々が開発を進めている事例に基づく演奏表情生成システム"Kagurame Phase-II" [1][2]では，演奏事例として，楽譜を階層的に分割した"旋律断片"を用いる．

† 芝浦工業大学 工学部 情報工学科
Department of Information Science and Engineering, Shibaura Institute of Technology
†† 株式会社ピコラボ
PicoLab Co., Ltd.
††† 株式会社トランス・ニュー・テクノロジー
Trans New Technology, Inc.
†††† 株式会社日本総合研究所
The Japan Research Institute, Limited

小節の分割方法として，人手で明示的に境界情報を与える方法と，小節構造を旋律断片自動生成アルゴリズムによって解析し，自動的に分割する方法が挙げられる．現在の Kagurame Phase-II では，後者を採用している．事例の選択は旋律断片単位で行われるため，旋律断片は楽譜の構造を反映していることが望ましい．また，旋律断片自動生成アルゴリズムの良し悪しが，演奏表情生成システムの能力に影響すると考えられる．そこで，小節構造を考慮した旋律断片自動生成アルゴリズムを構築し，実装した．また，演奏表情生成の実験を行い，本アルゴリズムの有効性について明らかにした．

以下，2 章で事例に基づく演奏表情システム Kagurame Phase-II について概説する．3 章で，楽譜記述用の言語"MusicXML"[6]について概説する．4 章で，MusicXML からの旋律断片自動生成について述べる．5 章で，今回実装した旋律断片自動生成アルゴリズムと従来のアルゴリズムによる演奏表情生成の比較と考察を行う．また，6 章では 2006 年 10 月に開催された Rencon 2006 の結果を報告する．

2. Kagurame システム

本章では，我々が作成した事例に基づく演奏表情生成システム Kagurame Phase-II の，概要および構成について説明する．

2.1 特徴

Kagurame Phase-IIは，鍵盤楽器による複旋律の楽曲の対象とした演奏表情システムである．本システムでは演奏状況を加味した演奏表情の生成が可能であるが，本報告では演奏状況については対象としない．本システムの特徴として，対象曲や演奏事例を様々な長さの旋律断片に分割し，旋律断片を対象として参考事例の検索を行うことが挙げられる．これにより，事例の総数が増加するため，少数の演奏事例を効率的に利用することが可能である．また，旋律断片に分割することで個々の検索単位の長さが短くなるため，類似旋律の検索に成功する可能性が向上する．

2.2 構成

Kagurame Phase-IIの構成を図1に示す．本システムでは，演奏表情の知識として演奏データ集を利用する．演奏データ集は，人間が演奏した実演奏を集めたものである．個々の演奏データは，演奏曲情報と演奏データからなる．演奏曲情報は，楽譜情報と境界情報からなる．楽譜情報は，従来は独自フォーマットのテキストファイルで与えていたが，現在はMusicXML で与えている．演奏データは，標準MIDI フォーマット形式(SMF)のシーケンスデータで与える．演奏表情の生成の際には，対象曲情報を入力として与える．対象曲情報は演奏曲データと同形式である．楽曲構成情報は，対象曲を旋律断片に分割する際に使用する．入力が与えられると，まず，演奏データ集から対象曲に類似した楽曲を検索する．類似した楽曲の検索は，対象曲全体だけではなく，対象曲を構成するフレーズや小節などの旋律断片に対しても行う．これによって，対象曲の旋律断片ごとに類似した旋律断片の集合が得られる．この集合を参考事例集とする．

次に，参考事例集の中の全ての事例について，重要度を評価する．重要度は，対象曲の演奏表情を生成する際に，その事例がどの程度参考になるかを表すスコアである．事例の重要度は，対象曲と事例との旋律断片の類似性から決定する．入力と参考事例とが類似しているほど，重要度は高くなる．旋律断片の事例の重要度は，1) 対象曲の断片との類似性，2) 前後に隣接する断片どうしの類似性，3) 親となる断片どうしの類似性，から求める．旋律断片の重要性は，参考事例の検索と同じ方法で評価する．旋律の重要性を正規化して，指数関数に適用したものを重要度とした．また，重要度の評価と同時に，全ての参考事例に対して演奏表情の分析を行う．演奏表情は，演奏データと楽譜とのずれという形で取り出すことができる．本手法では演奏表情の生成の際に，テンポや音の強さなどの絶対的な数値ではなく，これを相対的な変化量の比率に変換したものを利用している．この相対的な変化量を，演奏表情の大局比率と呼ぶ．

それぞれの参考事例に対して，重要度の評価と演奏表情の大局比率の分析を行うと，重要度でスコア付けされた演奏表情の大局比率の集合が得られる．この集合の中における個々の演奏表情の大局比率を重要度で加重平均して合成し，対象曲の演奏表情を生成する．最後にこの対象曲の演奏表情を対象曲の楽譜に適用し，演奏表情の与えられた対象曲の演奏データを作成する．生成された演奏データは，SMF形式のシーケンスデータで保存する．

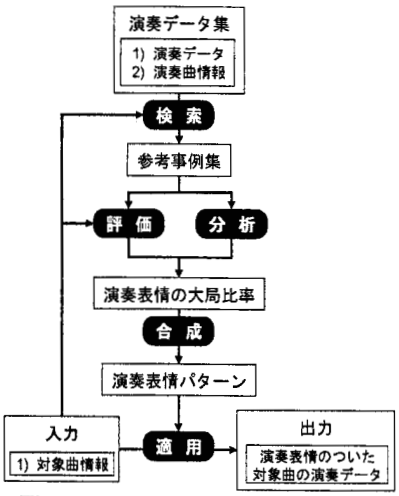

図 1 "Kagurame Phase-II" の構成

Kagurame Phase-IIが類似事例を検索するときに必要とする旋律断片の大きさはさまざまであるが，基本的には小節及び拍の2の累乗の長さである．例えば，1小節，2小節，4小節，8小節や，1拍，2拍，4拍，8拍，さらに半拍や1/4拍，1/8拍などを分割の単位とする(図2)．従って，ほとんどの断片は二つのより小さい断片を内包している．もっとも小さい断片は1音符である．

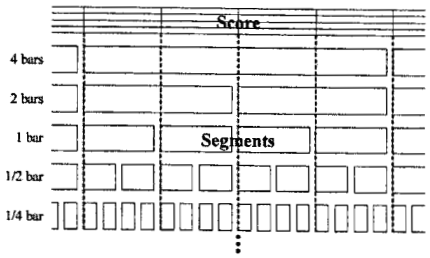

図 2 旋律断片への分割

3. MusicXML

MusicXML は，Recordare 社によって定義された，一般的な西洋音楽楽譜を記述するための XML 定義である．MusicXML は休符を明示的に記述すること

やスラーやタイなどの演奏情報を記述することが可能であるなど，一般的な西洋音楽の楽譜を記述するには十分な記述力をもっている．また，MusicXMLはテキストファイルなので，テキストエディタで容易に閲覧・編集できるといった利点がある．

MusicXMLの基本的な構造は，図3に示すように，パートを表わす <part>タグの下に小節を表わす <measure>，<measure>の下に音符を表わす <note>を列挙するという構造になっている．

MusicXMLでは，音符を表す<note>要素は<measure>要素の下に楽譜の音符の出現順に列挙されている．また，<measure>要素と違い，IDを持たない．<note>要素が何拍目であるかは，<note>要素の出現順と，それらの<note>要素の子である<duration>要素に依存する．したがって，「1小節目の1拍目」のように，拍単位で<note>要素を直接に指定することはできない．

```
-<part id="P1">
  -<measure number="1">
    +<attributes>
    ......
    +<note>
    +<note>
    ......
  -<measure number="2">
    ......
```

図 3　MusicXMLの基本的な構造

<duration>要素は，この音符の実際の長さを，MusicXMLのdivisions（MusicXML内部の解像度)の値に対する相対的な値として表している．

例えば，divisions値が24のとき，ある<note>要素の<duration>要素の値が12の場合，その<note>要素が表す音符の長さは4分音符の12/24=0.5倍の長さ，すなわち8分音符の長さとなる．

MusicXMLをサポートするソフトウェアは，市販品を含め，充実しつつある．中でも，カメオインタラクティブ社の楽譜作成ソフト"Finale"用にRecordare社が提供する"Dolet for Finale"の完成度が高い．

4. 旋律断片自動生成アルゴリズム

Kagurame Phase-IIは，演奏事例の検索や表情の適用を，楽曲を分割した旋律断片単位に行うことを特徴としている．そのため，楽曲をどのように旋律断片に分割するかによって，演奏表情の分析や適用が変化し，生成される演奏表情が異なると考えられる．また，本システムの「検索」プロセス(2.2節参照)では，対象曲の個々の断片(対象断片)すべてに対して，演奏データ集中にある類似した断片(参考事例)を検索する．対象曲の旋律断片総数をm，DBの旋律断片総数をnとすると，全ての事例についての検索にかかるコストはO(mn)と表すことができる．

そのため，「検索」プロセスを短時間に行うためには，楽曲を出来るだけ効率良く，自動的に旋律断片に分割することが求められる．

そこで，我々は，現在の旋律断片分割手法を見直し，旋律断片自動生成アルゴリズムを新たに構築し，Kagurame Phase-IIに実装した．

4.1 入力データの構成

現在のKagurame Phase-IIでは，入力となる対象曲情報と演奏事例データベースとなる演奏データ集は，図4のように構成されている．

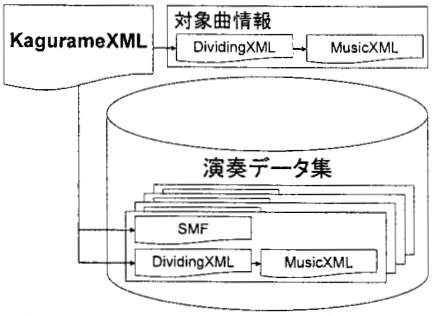

図 4　Kagurame Phase-IIで使用するデータの構成

生成対象曲および演奏データ集は，KagurameXMLを用いて指定する．KagurameXMLは以下の図5に示す文書型定義 (DTD, Document Type Definition)によって定義される．

```
<!ELEMENT kagurame (target, sampledataset+)>
<!ELEMENT target (#PCDATA)>
<!ELEMENT sampledataset (sample+)>
<!ELEMENT sample (structure, smf)>
<!ELEMENT structure (#PCDATA)>
<!ELEMENT smf (#PCDATA)>
```

図 5　KagurameXMLの文書型定義

<target>要素は，対象曲を表す．<sampledataset>要素は，演奏データ集を表す．

KagurameXMLの記述例を図6に示す．

```
<?xml version="1.0" standalone="no"?>
<kagurame>
  <target>(演奏表情生成対象曲のDividingXMLのパス)</target>
  <sampledataset>
    <sample>
      <structure>(DB1曲目のDividingXMLのパス)</structure>
      <smf>(DB1曲目のSMFのパス)</smf>
    </sample>
    <sample>
      <structure>(DB2曲目のDividingXMLのパス)</structure>
      <smf>(DB2曲目のSMFのパス)</smf>
    </sample>
  </sampledataset>
</kagurame>
```

図 6　KagurameXMLの記述例

DividingXML は旋律断片を定義する XML であり，MusicXML を小節単位で分割し，階層状にグループ化することで作成される．DividingXML の文書型定義を図 7に示す．

```
<!ELEMENT dividing (score, group)>
<!ELEMENT score (#PCDATA)>
<!ELEMENT group (group+ | measure+)>
<!ELEMENT measure>
<!ATTLIST score id ID #REQUIRED>
<!ATTLIST group id ID #REQUIRED>
<!ATTLIST measure number CDATA #REQUIRED>
```

図 7　DividingXML の文書型定義

<dividing>要素は，DividingXML のルート要素である．<score>要素は，この DividingXML に対応付けられている MusicXML のパスを表す．<group>要素は，小節のまとまりを示す．id 属性は DividingXML 内でユニークに設定する必要がある．<measure>要素は，小節を表す． number 属性によって，MusicXML の<measure>要素を指定する．

DividingXML では，小節内部の分割方法は定義しない．小節内部の分割方法に関しては，4.2 節で説明する．

DividingXML の記述例を図 8に示す．

```
<?xml version="1.0" standalone="no"?>
<dividing>
  <score id = "title">(MusicXMLのパス)</score>
  <group id = "main">
    <group id = "partA">
      <group id = "partA_1">
        <measure number="1"/>
        <measure number="2"/>
      </group>
      <group id = "partA_2">
        <measure number="3"/>
        <measure number="4"/>
      </group>
    </group>
  </group>
</dividing>
```

図 8　DividingXML の記述例

4.2 従来の小節分割アルゴリズム

Kagurame Phase-II へ MusicXML を導入した時点では，MusicXML の divisions 値(MusicXML 内部の解像度)に基づいて，機械的に分割する手法をとっていた．このアルゴリズムでは，まず小節を 2 で分割していき，2 で割り切れなくなったら 3 で分割していくという具合に素数で分割していき，最終的に旋律断片の長さが 1 になるまで分割していく．

L.V.ベートーヴェン「悲愴」第 2 楽章第 8 小節を例に，従来のアルゴリズムによって生成された旋律断片の概念図を図 9に示す．

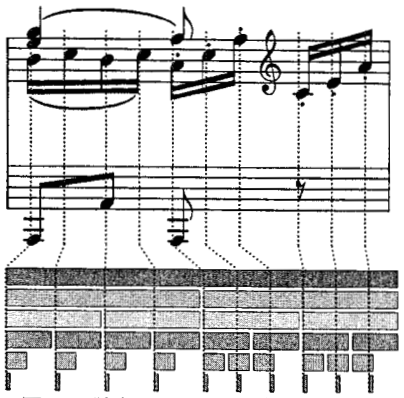

図 9　従来のアルゴリズムによって生成された旋律断片の概念図

従来の旋律断片分割手法では，音符の配置など，小節内の構造を十分に考慮していないため，三連符などでは適切な旋律断片が生成されないという問題点がある．

また，音符の有無を問わず一律に旋律断片が作成されるため，冗長な旋律断片が生成され，演奏生成のコストが無駄に増えるほか，SMF の出力編集処理にも悪影響を及ぼす可能性がある．

以上の問題点を解決するため，小節分割アルゴリズムの改良を行った．

4.3 改良した小節分割アルゴリズム

小節内の音符の拍位置を記述するリスト “Beat List” を導入する．この楽譜の divisions 値は 24 であるから，16 分音符の duration は 24/4 = 6，16 分 3 連の音符の duration は 24/6=4 となる．従って，この小節の BeatList は{0,6,12,18,24,28,32,36,40,44}となる．(図 10)

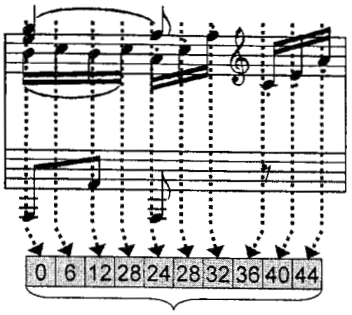

図 10　小節の BeatList

「悲愴」第 2 楽章は 4 分の 2 拍子の曲なので，最初は BeatList を 2 分割する．その後，BeatList を 2 で割り切れなくなるまで 2 分割していく．(図 11)

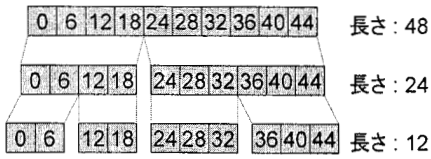

図 11　BeatList の分割(1)

途中で 2 分割できない BeatList が出現した場合，それぞれの BeatList の要素数が 1 になるように分割する．(図 12)

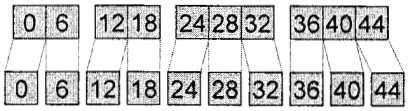

図 12　BeatList の分割(2)

このようにして，最終的に，それぞれの BeatList の要素数が 1 になるまで分割していく．これらの BeatList の分割方法に基づいて，旋律断片を生成する．

改良したアルゴリズムによって生成された旋律断片の概念図を図 13に示す．

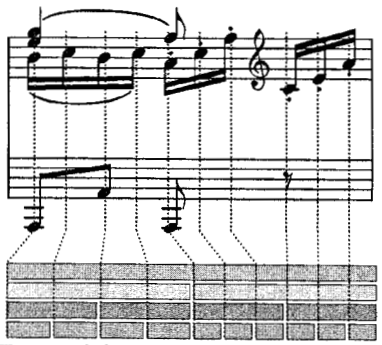
図 13　改良したアルゴリズムによって生成された旋律断片の概念図

5. 小節分割アルゴリズムの比較

5.1 参考事例数の比較

従来のアルゴリズムと改良したアルゴリズムとで，生成された旋律断片の数を比較した．参考事例数の比較には，以下の 5 曲を用いた．

- L.ボッケリーニ「メヌエット」
- J.S.バッハ「メヌエット ト長調」
- E.ローデ「あやつり人形」
- R.シューマン「兵士の行進」
- R.シューマン「メロディ」

従来のアルゴリズムと改良したアルゴリズムでの参考事例数の比較を表 1に示す．

表 1　参考事例数の比較

曲名	従来方式	改良方式	事例数の比
メヌエット(ボッケリーニ)	676	498	0.74
メヌエット(バッハ)	525	448	0.85
あやつり人形	1098	842	0.77
兵士の行進	724	354	0.49
メロディ	575	571	0.99

「兵士の行進」では，参考事例数がほぼ半分近くに減少した．その一方，「メロディ」のように，参考事例数がほとんど減少しない曲もみられる．

実験で用いた 5 曲を平均して，従来のアルゴリズムと比較して旋律断片数が 75.4%に減少した．また，楽曲によって旋律断片を削減する割合が異なることがわかった．

5.2 小節分割アルゴリズムの妥当性の検証

小節分割アルゴリズムの妥当性の検証を行うため，それぞれのアルゴリズムで対象曲と同一曲を DB に用いて演奏表情を生成し，比較した．生成された演奏と人間の実演奏データとの比較には，以下の式を用いた．

$$E_v = \frac{1}{n}\sum_{i\le n}\left\{E_{kp2}(i) - E_h(i)\right\}^2$$

$E_{kp2}(i)$ は Kagurame Phase-II が生成した演奏データの i 番目の音符の Velocity の値であり，$E_h(i)$ は人間の実演奏の i 番目の音符の Velocity の値である．また，n は楽曲中に含まれる音符の総数である．E_v は $E_{kp2}(i)$と $E_h(i)$との差分の自乗平均である．

E_vが小さいほど人間の実演奏と Kagurame Phase-II の出力が近いことを表す．

旋律断片数の比較をした曲について，演奏対象と同一曲を DB に用いた Closed-Test 形式で E_vを算出した．

Velocity 値の差分の自乗平均の比較を表 2に示す．

表 2　Velocity 値の差分の自乗平均の比較

曲名	従来方式	改良方式
メヌエット(ボッケリーニ)	57.67	54.64
メヌエット(バッハ)	9.85	12.38
あやつり人形	24.08	16.88
兵士の行進	72.63	62.31
メロディ	19.61	19.62

5.3 考察

実験結果から，改良したアルゴリズムの採用によって，E_vが小さくなり，システムの出力が改善されているという傾向がみられる．しかし，「メヌエット(バッハ)」のように，Velocity 分散値が大きくなってしまった曲もある．

以上の結果から，旋律断片自動生成アルゴリズムが，実際に出力 SMF に影響を及ぼしていることが明らかになった．また，曲によって，最適な旋律断片自動生成アルゴリズムが異なると考えられる．

6. 生成された演奏の評価 - Rencon2006

本システムで生成した演奏表情を評価し，他の演奏表情生成システムに対する位置づけを明らかにするために，"Rencon2006"(ISMIR'06 Performance Rendering Contest)[7]に参加した．"Rencon"とは，"演奏生成システムの生成演奏をコンテスト方式で評価するプロジェクト(とそのコンテスト)"のことであり，その名称は"Performance Rendering Contest "に由来している．

Rencon2006 では二つの部門が設けられ，課題曲が"Chopin Etude, op. 10, no. 3, bars 1-21"に設定された"Compulsory Section"と，曲目や演奏楽器を自由に選択できる"Open Section"からなっている．我々は，Kagurame を他のシステムとなるべく同じ条件下で評価をするために，Compulsory Section に参加した．Kagurame Phase-II で参加用演奏データを生成したとき，以下の曲を演奏データとして使用した．

- Chopin Prelude Op.28 No. 15, bars 1-8.

Rencon2006 の Compulsory Section には，4 システムによる 7 演奏データの参加があり，評価時にはこれらに加え，"機械的な演奏(Deadpan)"と，22 人のピアニストの演奏から生成された"平均的な演奏(Average)"が運営者によって加えられ，9 演奏データが聴き比べられた．聴き比べ評価時の音源を統一するために，参加者には SMF または，CEUS 形式による提出が求められた．提出された演奏データは，Rencon2006 開催前に，運営者によって Bösendorfer Computer Concert Grand Piano CEUS[8]で演奏され，MP3 形式で収録された．このようにして評価用音源を統一された全参加者の提出演奏データは，「どの演奏が，どのシステムによる演奏なのか(誰による演奏なのか)」がわからないように，Web 投票フォームを通じて Rencon2006 聴衆によって聴き比べ評価がなされた．評価項目は，次の 3 項目である．

- Overall quality of performance
- Extent to which the performance sounds human
- How interesting is the performance

評価結果を，表 3に示す．

表 3 Rencon2006 Results

SYSTEM	QUALITY	HUMAN	INTEREST
(Average)	1.8	1.5	2.2
Pop-E	3.1	2.5	3
Kagurame Phase-II	3.6	2.9	3.6
(Deadpan)	3.9	3.8	4.5
Ha-Hi-Hun (closed)	4.1	3.1	4.2
Coper (open)	4.1	3.2	2.9
Coper (closed)	4.1	3.5	3.5
Ha-Hi-Hun (deadpan)	4.5	4	4.5
Ha-Hi-Hun (open)	4.6	4.2	4.2

聴き比べ評価順位では Pop-E のほうが Kagurame よりも上位であったが，Pop-E は前処理において人手が介入する部分が多く残っているため，「Kagurame は人手の介入度合いが低く，自律性の高い事例ベース型表情付けシステムの中では最上位の成績を収めた」ということができよう．

しかし，(Average)との差は格段にあり，今後検討すべき課題の多さを物語っている．

7. おわりに

本報告では，事例に基づく演奏表情生成システム Kagurame Phase-II における，MusicXML からの旋律断片自動生成アルゴリズムの改良について述べた．旧アルゴリズムと新アルゴリズムについてそれぞれ Closed-Test について演奏表情生成を行い，新アルゴリズムの有効性について明らかにした．

今後は，対象曲と異なる曲を DB に用いた Open-Test 形式で生成を行い，さまざまな組み合わせで生成した演奏を定量的，定性的に評価することで，旋律断片自動生成アルゴリズムの改良を進めていきたいと考えている．

参 考 文 献

[1] T. Suzuki: Kagurame Phase-I, musical expression generation system with case-based method, Proc. ICAD2002 Rencon Workshop, pp.13-20 (2002).

[2] T. Suzuki: The second phase development of case based performance rendering system "Kagurame", Proc. IJCAI03 Rencon Workshop (2003).

[3] 金子雄介，鈴木泰山，徳永幸生: 事例に基づく演奏表情生成システムにおける表情生成式の最適化，情報処理学会研究報告，2004-MUS-58 (Dec 2004).

[4] 金子雄介，鈴木泰山，徳永幸生: 事例に基づく演奏表情生成システムにおける演奏類似性と試聴評価，情報処理学会研究報告，2005-MUS-59 (Feb 2005).

[5] 鈴木泰山，金子雄介，徳永幸生: 事例に基づく演奏表情生成手法の演奏表情生成アルゴリズムの分析，情報処理学会研究報告，2005-MUS-59 (Feb 2005).

[6] Recordare MusicXML Web Site: http://www.recordare.com

[7] Rencon Web Site: http://shouchan.ei.tuat.ac.jp/~rencon

[8] Bosendorfer Computer Concert Grand Piano CEUS: http://www.boesendorfer.com/_english_version/startseite/ceus_praes.html

[9] 平賀瑠美, 平田圭二, 片寄晴弘: 蓮根, めざせ世界一のピアニスト, 情報処理, Vol.43, No.2, pp.136-141 (2002).

[10] 野池賢二，平田圭二，片寄晴弘: Rencon エントリキット第 1 版の仕様の考察，2003-MUS-50 (May 2003).

[11] 浜中雅俊，平田圭二，東条敏: GTTM に基づく楽曲構造分析の実装：グルーピング構造と拍節構造の獲得情報処理学会研究報告，2004-MUS-56 (Aug 2004).

[12] 清水厚志，鈴木泰山，野池賢二，徳永幸生: 事例に基づく演奏表情生成システムへの MusicXML の適用，第 68 回情報処理学会全国大会 2L-7 (Mar 2006).