

自動採譜装置を用いた点字楽譜変換に関する研究

Research of the automatic music scoring system for Braille-Score

○工藤裕太朗、小坂谷壽一

○Hirotao Kudo, Juichi Kosakaya

八戸工業大学大学院工学研究科電子電気・情報工学専攻

Graduate School of Electrical, Electronic and Information Engineering,
Hachinohe Institute of Technology

キーワード: 自動採譜装置 (automatic scoring system), 伝統楽器 (traditional instrument),
点字楽譜 (Braille-Score), ABC 記譜法 (ABC notation)

連絡先: 〒031-8501 青森県八戸市大字妙字大開 88-1

八戸工業大学工学部システム情報工学科 小坂谷壽一

TEL: 0178-25-8034 FAX: 0178-25-1691 E-mail: kosakaya@hi-tech.ac.jp

1. はじめに

現在、全国の視覚障害者の数は約 40 万人とされている。先天的に視覚障害を持つ人は、盲学校等における音楽教育で何らかの楽器に触れる機会が与えられている。しかし、この視覚障害者向け音楽教育において、圧倒的に点字楽譜の専門家が少ない為に教育環境が整わず、優れた素質がありながら音楽に深く触れることが出来ず、専門家への道を断念する人が多いとされている。一方、近年のヴァン・クライバーン国際ピアノコンクールにおいて日本の視覚障害者が優勝するなど、多くのハンディを乗り越え文化や芸術面において非健常者が優れた素質を開花させ、様々な分野で活躍している事も事実である。このような背景において、「専門の点字楽譜知識」が無くとも、「楽器を弾けば自動的に点字楽譜が

生成」出来れば、視覚障害者にとって一層音楽が身近になり、この分野の音楽教育に画期的な進歩がもたらされる事が予想される。この「視覚障害者に優しい点字楽譜自動生成装置の実現」を目指し、本研究開発を行った。

2. 研究概要

演奏した楽曲が自動的に電子譜面に変換されるには、装置と繋がれた楽器から入力された音源が、装置内部で適切に処理される必要がある。今回実際に演奏した音源から楽譜を出力する為に、[1]に示す自動採譜手法を導入した。本研究での自動採譜とは、楽曲の入力・解析・譜面化までの流れの事を示し、この処理を行う装置のことを自動採譜装置と呼ぶ。自動採譜装置では、専用の楽器を演奏するだけで、一般的な楽譜を出力することが可能となっている。

自動採譜処理の流れは、音源が入力されてから、周波数分析処理によって音源の周波数解析を実行し、三味線用音階判読処理によって予め登録してある音階周波数と解析した周波数との比較、周波数が一致した部位を音階として特定し、採譜処理によって西洋譜面または三味線用文化譜面として出力を行うというものである。解析された音源の演奏情報の出力をMIDIデータ生成という形で行った。

本研究では、譜面化を行う際に自動採譜装置が出力するMIDIデータを変換して点字楽譜を出力することを目的としている。以下の機能の実装に向け研究を行った。

(1) MIDI データをテキストデータに変換

自動採譜装置から出力されたMIDIデータを点字変換処理用のテキストデータに変換する。

(2) 点字記号データベース

MIDI テキストデータの内容に対応した点字記号の出力を行う為に必要であると考えた。

(3) 点字記号出力処理部

変換されたMIDI テキストデータの情報を元に、点字記号データベースから画像を選択し、配置する。

本研究の装置構成を図1に示す

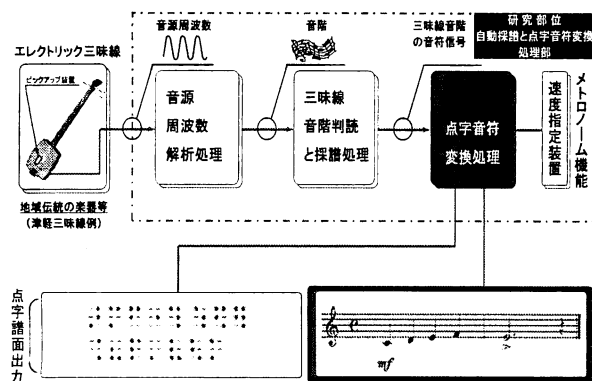


図1 自動採譜及び点字音符変換装置構成

2.1 エレクトリック三味線について

入力に際し、一般的な三味線を使用した場合、弦同士での共振が起こる、撥で楽器を打つために不要な音まで入力されてしまう等の録音上の不都合が確認されている。

今回音源の入力に用いているエレクトリック三味線では、写真1に示すように、弦毎にピックアップマイクロフォンを取り付けることによって独立した音階が抽出できるようになっている。また、弦同士の共振を避けるため、胴体を演奏にあたって必要な部分を除き空洞構造としている。

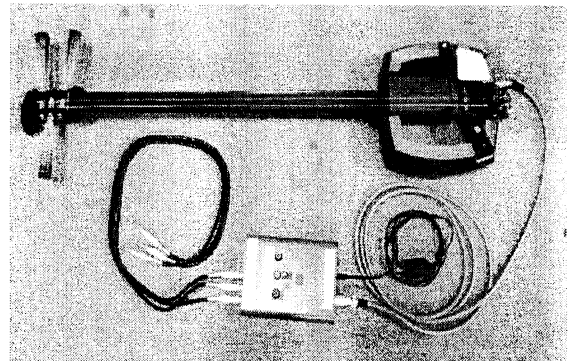


写真1 エレクトリック三味線

3. 点字での表現方法

点字の代表的な表現について記す。

点字楽譜は、他の点字表記と同様に、6つの点の組み合わせによる記号を左から右へ横1列に表記する記号譜である。音の高低や長さを表す際に、譜面上の音符の上下や形によって表現する西洋譜面とは全く異なった表記法で書かれている。

例として、図2に4分音符と64分音符のドの音を挙げる。



図2 4分音符、64分音符：ド

上から一段目と二段目で音の高さを表現し、下の一段で音の長さを表現している。図2の点字記号が4分音符であるか64分音符であるかは、通常は1小節内の音符と休符の長さや数によって判断する。ただし、区別がつかない場合には分別記号を用いて判断する。休符についても同様である。

音の高さについての対応は図3の通りである。

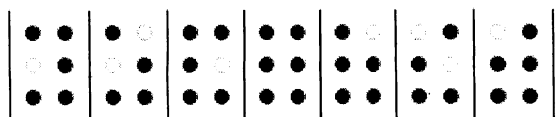


図3 点字による音階の表現

図3の左からド、レ、ミ、ファ、ソ、ラ、シとなっている。これら七つを幹音と呼ぶ。この幹音がどの音域に属するのかわかる音列記号が存在し、音階の判別を可能にしている。

西洋譜面によく用いられる記号として、以下の点字記号を図4に示す。

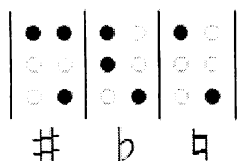


図4 臨時記号と調号

これらを臨時記号として扱う場合は、音符を表す点字記号の前(音列記号がある場合は音列記号の前)に置く。西洋譜面と同様に臨時記号の有効範囲は1小節内である。

調号として扱う場合、#なら#の点字記号を必要な数だけ並べて記述する。ただし、4つ以上調号が並ぶ場合は、調号の前に4~6の数字を添えて表現する。

4. MIDI データのテキスト化と点字の対応

楽曲の典型的なMIDIテキストデータ例を図5に示す。

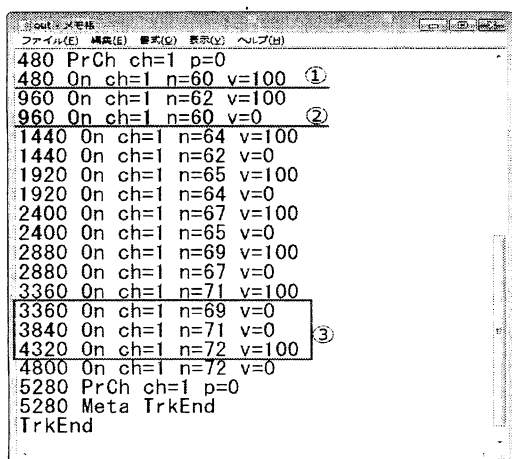


図5 MIDIテキストデータ

この曲では、1つの音を表現するために①「480 On

ch=1 n=60 v=100」という情報が使用されている。これは、音の始まりが480の位置にあり、この音を演奏する楽器が使用するチャンネルが1番で、音階nが60(一般的なドの音)、である事を示している。v=100はベロシティ(音の強さ)を表しており、0が無音で128が最大音量を表す。位置の右隣のOnは音を鳴らすことを示している。この音は次の音の情報が読み込まれるまではひたすら音を出し続ける。音を止めるためにはv=0の音情報を加えるか、Onの代わりにOffを用いる必要があり、実際に最初に示した音の後ろには②「960 On ch=1 n=60 v=0」という情報が書き込まれている。480と960の差である480だけ音を伸ばしていることを示している。この曲では、480という数値が四分音符を表しており、①で音が始まってから②で終了するまでにちょうど四分音符一つ分の時間がかかっていることが示されている。また、③のテキストでは、二段目に記述してある位置3840から三段目の位置4320まで音が無い状態を示している。この差である480が無音であることから、この区間を四分休符にあてることができる。点字楽譜に使用される記号のデータベースから、nの数値が一致している位置の差を音符に置き換え、全ての音がv=0またはOffの情報を加えられたことで無音状態であることを判別し、次の音が鳴るまでの区間の長さから休符の種類を選択していけば、点字楽譜での音符と休符の表示が可能であると考えられる。

5. 譜面変換方法考察

5.1 1対1対応法

点字記号データベースを作成し、テキストデータとの対応付けによって点字楽譜を表現する方法。MIDIテキストデータと点字記号をひとつひとつ照らし合わせて点字楽譜を生成する。

現状の自動採譜装置の運用範囲では致命的な判断ミスは生じないと考えられるが、楽曲のトラック数の増加と共に音符や休符の処理において判断ミスが生じてくる可能性がある。

5.2 Python スクリプト利用法

LilyPond というオープンソースの楽譜組版ソフトがこの Python スクリプトを用いている[3]。このソフトは、ソースにテキストデータを利用して楽譜を生成するもので、今回の MIDI テキストを用いた点字楽譜表示に関して有効であると考えられる。西洋楽譜で用いられる記号から点字楽譜で用いられる記号への画像の差し替えを行い、MIDI テキストデータを用いてソフト側に対応付けをさせるようにするといった手法を取ることが可能で、複雑な譜面の生成にも利用できる。

実装に際して、点字楽譜フォーマットの整備、MIDI テキストデータがそのままソフト側に流用できないことにより生じるテキストの翻訳作業等、元になるソフトに大幅な変更が必要になる。

楽譜情報の入力から西洋譜面、点字譜面への変換例を以下に示す。

(1) LilyPond での楽譜情報入力

以下のような情報が書き込まれたテキストデータ (test.txt とする) の拡張子を書き変えて LilyPond に読み込ませる形式 (test.ly) とし、変換を実行する。

```
{  
  ♪time 2/4  /4 分の 2 拍子  
  ♪clef bass  /へ音記号  
  c4 c g g a a g2  /音階と音の長さの指定  
}
```

(2) 上記のテキストデータ入力を受けて、図 6 のような西洋譜面を pdf 形式で出力する。

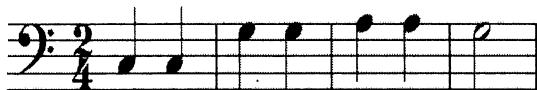


図 6 西洋譜面出力例

点字記号データベースを組み込むことによって、結果的に得られる点字楽譜を図 7 に示す。

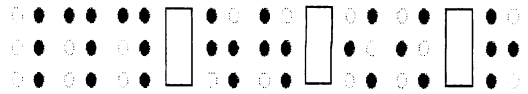


図 7 点字楽譜出力例

この方法を用いることによる最大のメリットは、実際の演奏とのズレが生じた場合、MIDI テキストデータの書き換えを行い、再度 LilyPond に読み込ませることによって簡単に修復が行えることである。6.1 で示した 1 対 1 対応法では、元になっている MIDI テキストデータの音階情報が数値に依存しているために、修復すべき点の発見がし辛くなる。さらに、高音や低音になればなるほど、数値によって音階を判別するのに時間を費やすことになることも予想される。

LilyPond が楽譜情報として読み込める形式に変換されているテキストデータでは、C や G といったコードを元に音の判別が可能である。音階が大きく変わる場合でも、C が一般的なドの音であるとした場合、オクターブ上のドの音は C' で表される。数値で音階を表す場合には 0 から 128 の数字全ての音を把握する必要があるが、文字によるコードで音階を表すと幹音である c, d, e, f, g, a, b の他に、音の高さを表す記号 2 種類、臨時記号である # と b を表す記号 2 種類があれば良く、音階を表現するには十分であると判断した。

6. テキストデータ変換

当初、図 5 で示した MIDI テキストデータから音符の長さや休符の有無等の情報を得ることを想定していたが、非常に困難であったため、図 5 のようなテキストデータではなく、ABC 記譜法によるテキストデータを出力する方法に切り替えることにした[4]。

ABC 記譜法は文字譜を表現する際に用いられる方法の一つである。MIDI データから ABC 記譜法に従って文字で表現された楽曲の例を図 8 に示す。

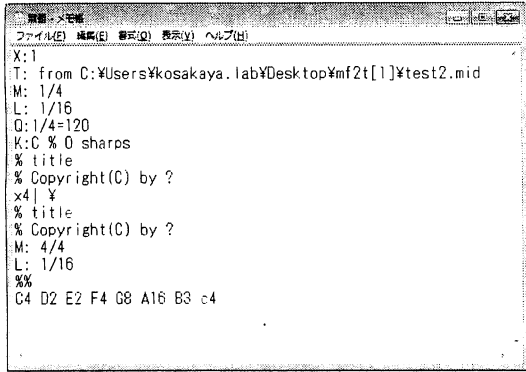


図8 ABC記譜法による文字譜表現例

このテキストデータを西洋譜面にする図9のようになる。

from C:\Users\kosakaya.lab\Desktop\mf2t[1] est2.mid



図9 西洋譜面

LilyPondに読み込ませるテキストデータと比べて音符の表現の仕方に大きな差が見られず、テキスト変換が容易である。

今回は、テキスト変換を行うプログラムを利用し、ABC記譜法で記述されたテキストデータをLilyPond用のテキストデータに変換する際のルールを定め、変換を行った。ABC記譜法での「c1」と、LilyPond用の記述法での「c1」では音の長さが正反対になる。ただし、符点のついた音符を表す数字はこの限りでないため、個別に対応させた。作成した変換ルールの一部を図10に示す。

MIDI Note	LilyPond Note
G16	g1
A16	a1
B16	b1
C8	c2
D8	d2
E8	e2
F8	f2
G8	g2
A8	a2
B8	b2
C3	c8.
D3	d8.
E3	e8.
F3	f8.
G3	g8.
A3	a8.
B3	b8.
C2	c8
D2	d8
E2	e8

図10 変換ルール表

今回の例では図8の最下段にある8つの音符を表すテキストデータを図11のように変換した。

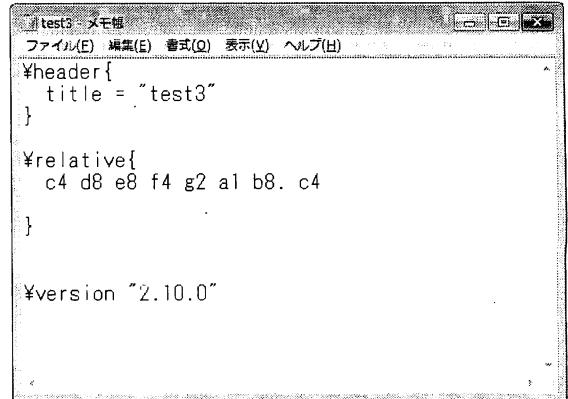


図11 LilyPond用に変換したテキストデータ

このテキストデータをtest3.lyにリネームし、LilyPondで出力すると図11のような西洋譜面が得られた。

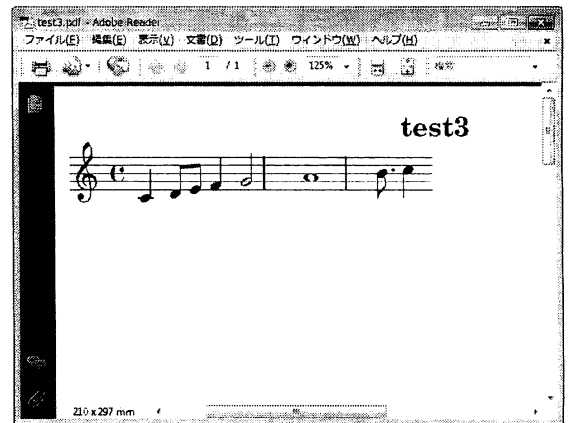


図12 LilyPondから出力された西洋譜面

図9と比較して相違無いことが確認できた。

7. 点字記号変換

7節でLilyPond用に変換されたMIDIテキストデータから点字記号への変換処理を行うにあたって、点字記号画像とテキストデータとの間で対応付けを行った。対応付けは、MIDIテキストデータ内の{}で囲まれた部分中から音符を表す文字列を検索し、対応する音階の点字記号画像を並べていく手法を取った。テキストデータを受け取り、点字楽譜の出力までの動作の流れを以下に示す。

- (1)変換済みテキストデータ内の文字列「¥relative」の検知
- (2)変換対象である文字列の検索
- (3)変換対象があれば、その文字列とデータベースを照会し、対応画像の配置を繰り返す
- (4) ”}”を検出することで、変換対象がこれ以上ないことを判別
- (5)終了処理を行う

図 11 で示したテキストデータから出力された点字楽譜を図 13 に示す。

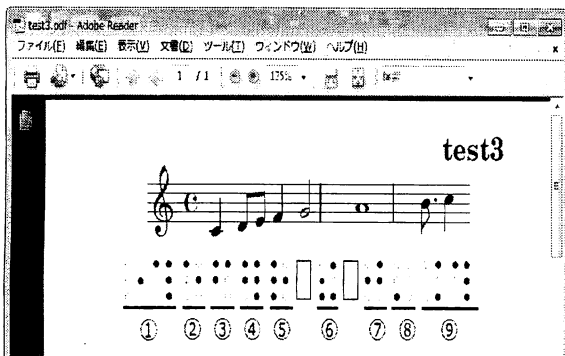


図 13 出力された点字楽譜

①の部分ではこの音が第 4 音列の音であることを示す音列記号と音階名 c に対応したドを表す点字が出力された。②から⑧までは音列が同じ音であるので音列記号は省略されている。⑨の部分では、①と音階名は c で同一であるが、①よりオクターブ高い音列の音と認識されたために第 5 音列を表す音列記号が出力され、音階を正しく示していることが確認できた。

全体の変換処理の流れは以下の通りとなった。

- (1)エレクトリック三味線からの演奏入力を受けて自動採譜装置が MIDI データの出力を行う。
- (2)MIDI データをテキストデータ変換処理部が受け取り、{} で囲まれている文字列部分の翻訳を行う。
- (3)テキストデータの翻訳が終了したら、拡張子を.ly に変換し、LilyPond に渡し、点字記号データベースに登録してある画像パスを参照し、点字記号の配置を行う。
- (4)点字楽譜の表示を pdf 形式で行う。

図 13 の点字楽譜は以上の流れを受けて出力された。今

回は音符を表す記号の出力のみに留まったが、MIDI テキストデータの翻訳のルール追加と共に、休符や臨時記号の表現等も実装していく予定である。

8. まとめ

本研究では、自動採譜装置によって録音した音源を点字楽譜に変換する手法を考案した、また、Python スクリプトを利用した点字記号配置によって点字楽譜を表現するという手法の提案と、テキストデータ変換を経た西洋楽譜の出力を可能にした。今回は採譜した楽曲の MIDI データを適切に変換し、点字楽譜化を行うことができたが、前後の音の高低差表現の問題や、拍子への対応ができていないという課題が残った。今後は、自動採譜装置から得た MIDI データをテキストデータへ変換する処理の精度向上、変換したテキストデータを用いて点字記号の出力を行う処理部の充実、点字記号データベースから適切な点字記号の選出を行う処理の実現、適切な拍子に対応した譜面作成処理部の実装を行っていく予定である。

参考文献

- [1] 小坂谷壽一, “伝統音楽(津軽三味線)保存用自動採譜装置に関する研究”, IEICE 信学技報, EA-2008-142(2009)
- [2] 文部科学省 点字楽譜の手引 2009.
- [3] LilyPond Music notation for everyone
<http://lilypond.org/>
- [4] 実用 abc 譜
http://www.asahi-net.or.jp/~mf4n-nmr/abc_notation.html