

楽曲の特徴を考慮したコンピュータによるオーケストラ指揮の実現 Implementation of Conductor by a Computer Considering the Characteristics of the Music

笹川 悠馬[†]

Yuma Sasagawa

萩原 威志[‡]

Takeshi Hagiwara

1 はじめに

大人で演奏を行うとき、その演奏の完成度には指揮者の技量が大きく影響する。指揮者には多くの能力が求められる。指揮の変化は繊細なものであるため十分な知識・技術・経験がなければ演奏者に指示をうまく伝えることができず、表現力が乏しい演奏になってしまう。また指揮者の力量が足りなすぎるとテンポが不安定になったり奏者のタイミングが揃わなくなってしまう、最悪の場合では演奏が途中で止まってしまうこともある。このような問題は特に初心者たちによる演奏団体や指揮の経験者がいない演奏団体で起きることが多い。演奏者のレベルに関わらず、よりよい演奏をするためには良い指揮の存在は必要不可欠である。

本研究ではコンピュータが指揮のアニメーションを出力するシステムを実装した。本システムでは楽曲の特徴を分析することで、奏者の能力を引き出すことができる指揮を出力することができる。また出力がコンピュータによって作られるアニメーションであるためテンポが乱れたりせず常に安定した指揮を提供できる。本システムを用いることで指揮者がいない状態でも大人数の演奏を手軽に行うことができ、学校の吹奏楽部やアマチュア演奏団体の練習など様々なシーンで活用することができる。

本システムは性質上、音楽にあまり詳しくない人間が使うことが想定される。そのため本研究では音楽に詳しくない人でも手軽に使えることを前提とし、簡単な入力のみでより楽曲の特徴を考慮した指揮を出力することを目的とした。楽曲の特徴を考慮した指揮をコンピュータで実現する際、大きな問題となるのが楽譜に書いていない情報をどのように考慮させるかである。指揮をする際拍子やテンポだけではなく楽曲の雰囲気(元気、静かなど)を考慮させなければならない。これは楽譜に書かれていることもあるが書かれていないことの方が多い。楽曲の雰囲気がかかれていない場合、指揮者は自分の感性でそれを考慮するが、コンピュータには感性がない。そのためコンピュータに指揮をさせる場合、どのようにして楽曲の雰囲気を考慮させるかが重要になる。また目立たせたいパートやそのパートの特徴によっても指揮の動作は変化する。特別なパートを主張させたい場合指揮者はそのパートのリズム、アーティキュレーションなどといった特徴を考慮した指揮をする必要がある。しかしどのパートを注目するかは指揮者によっても違うため楽譜に書いてあるのみの情報ではパートの特徴をうまく考慮させることができない。

本論文ではこれらの問題について考慮した指揮システムのモデルの構築及びそのシステム化を取り扱う。楽曲の雰囲気は楽曲のテンポ・強弱・楽器情報など楽譜から読み取れる情報を数値化し、それらを分析することで考慮させる。また注目したいパートの特徴はユーザーから楽曲の分割点とその分割点まで注目したいパートの入力を求めることで考慮させる。以下、第 2 章では本研究に関連する研究や技術について整理する。第 3 章では本研究の提案手法について詳しく説

明する。第 4 章では実装したシステムが入力した楽譜データの特徴によってどのように変化するかを考察・評価し、第 5 章で本論文の総括をする。

2 関連研究・技術

2.1 関連研究

これまでも楽曲と指揮を結びつける様々な研究がなされてきた。Max Mathews の Radio Baton[1] は両手に持った 2 本のバトンで平面センサ上を叩いたりこすったりすることで MIDI のコントロール信号を操作し楽曲の演奏を行う演奏インタフェースであり指揮システムの先駆といわれている。それに続き宇佐らの Conducting Simulator[2]、Nakra らによる UBS Virtual Maestro[3]、馬場らによる Virtual Philharmony[4] など様々な指揮システムが開発された。これらはセンサを用いて指揮者の動きをとらえ、そこから拍点を抽出してテンポを割り出しそれに合わせて楽曲を再生するというものであり指揮そのものを出力するというものではなかった。

また楽曲の感情を考慮する研究もある。森口らは MIDI 形式の音楽データに感情表現を付与するシステム MOR2ART[5] を開発した。Antonio Pedro Oliveira らは希望された感情の音楽を生成するシステムの提案をした [6]。これらのシステムは楽曲を生成するという目的で研究されてきた。本研究では楽曲の感情を用いて指揮を出力することが目的で、この出力結果は楽曲を演奏する演奏者に影響を与えるという点が特徴である。

2.2 指揮法

指揮法とは指揮者が管弦楽団や吹奏楽団、合唱団などを指揮するための技法である。指揮法には各拍子における図形の書き方や指揮の運動の種類といった基本的な技法から、複雑にリズムが組み合わさっているときの振り方やテンポの変化の表し方といった応用的運動まで様々な技法を網羅されている。指揮者の動きに絶対的な規則や法則は無いという考えもあるが、一般的にこのような技法を習得することで演奏者に伝わりやすい指揮が実現できると考えられている。本研究では指揮法に示されている技法を適切に表現できている指揮が楽曲の特徴を考慮した指揮であると考え、できるだけ少ない入力でも楽曲の特徴を考慮した指揮を出力することを目的とする。

2.3 MusicXML

本研究では入力に使う楽譜データとして MusicXML を用いる。MusicXML は楽譜表記のためのオープンなフォーマットファイルであり Recodare 社によって開発された。17 世紀から西洋で一般的に利用されている音楽表記を採用してお

[†]新潟大学大学院自然科学研究科

[‡]新潟大学工学部情報工学科

り Finale や Sibelius など多くの楽譜作成ソフトで作成することができる [7].

3 提案手法

指揮出力システムの仕様

この章では提案システムの仕様を説明する. 本システムは指揮のアニメーションを出力する. このシステムを用いることで指揮の経験者がいない演奏団体に対しても指揮を提供でき, 大人数での演奏をより手軽にすることができる. 本システムは性質上, 音楽にあまり詳しくない人間が使うケースが想定されるので難しい入力の要求は避けるべきである. そのため本システムは入力を MusicXML ファイル, 楽曲の分割点, その分割点までに注目したいパートのみとした. これらの入力は音楽に関する知識の有無に関係なく入力できるため本システムは音楽に詳しくない人でも手軽に扱うことができる. 出力はできるだけ楽曲の特徴を考慮したアニメーションを出力するべきである. 本システムでは楽曲のテンポ・拍子・音量・雰囲気・注目するパートの特徴を考慮した指揮のアニメーションを出力する. 指揮のアニメーションは HTML5 と CSS3 を用いて出力する. システムは次のような流れで処理をする.

1. ユーザーに MusicXML ファイルと楽曲の分割点, その分割点までに注目したいパートの入力を求める.
2. 入力された楽曲の小節番号, テンポ, 拍子, 各パートの楽器, 音符の長さ, 強弱を抽出する.
3. 楽曲の拍子, テンポから指揮の基本図形と動作速度を決定する. これらは 1 小節毎に決定する.
4. 楽曲の強弱と楽器情報から音量を計算し, その音量に応じて指揮の大きさを決定する. 音量は 1 小節毎に計算する.
5. 楽譜に書いてある情報から楽曲の雰囲気を決める. 楽曲の雰囲気は 1 小節毎に判定され, その結果に応じて指揮の図形や動作の加速・減速度に修正を加える.
6. ユーザーが入力した楽曲の分割点とその分割点までに注目したいパートの情報をもとに指揮の図形や加速・減速度に修正を加える. この修正は 1 拍毎に行う.
7. 指揮のアニメーションを出力する. アニメーションは HTML5 と CSS3 を用いて出力する.

以下, 各ステップを詳細に説明する.

Step:1 MusicXML の入力

本システムは指揮を出力したい楽曲の楽譜データ, 楽曲の分割点, その分割点までに注目したいパートの入力を求める. 楽譜データは MusicXML ファイルのみ扱える. 楽曲の分割点は小節単位で入力する. 楽曲の分割点と注目したいパートの情報はそのパートの特徴を考慮した指揮を出力する際に使用する. 詳細は Step:6 で説明する.

Step:2 楽曲情報の抽出

本研究では楽曲の情報を抽出するために CrestMuse Toolkit [8] を用いる. CrestMuse Toolkit は音楽情報処理システムを効率的に扱うための Java ライブラリである. MusicXML からの情報抽出だけでなく音楽データのリアルタイム処理, 確率推論など様々な機能を提供している. 本システ

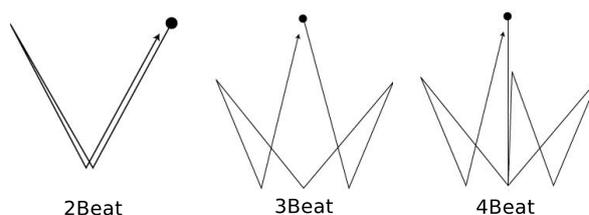


図 1: 各拍子における指揮の基本図形

表 1: 本システムにおける各速度記号とテンポの対応

速度記号	意味	テンポ
Largo	幅広く緩やかに	40
Larghetto	やや遅く	60
Adagio	ゆっくりと	66
Andante	歩くような速さで	76
Moderato	中庸な速度で	108
Allegro	快速に	120
Vivace	活発に	144
Presto	急速に	160
Prestissimo	Prest より速い	200

ムでは MusicXML から楽曲の小節番号, テンポ, 拍子, 各パートの楽器, 音符の長さ, 強弱を抽出するために使用する. テンポと拍子は指揮の基本的な図形と運動の速さを決めるために使用する. それ以外の情報は楽曲の音量, 雰囲気を考慮した指揮を出力するために使用する.

Step:3 楽曲のテンポ・拍子を考慮

指揮をする上でもっとも基本となるのは指揮の図形(運動の軌跡)と速度である. 演奏者は指揮の図形から楽曲が何拍子で現在何拍目かを判断し, 指揮の速度から演奏の速度や演奏を始めるタイミングを予測する. 指揮の基本図形は楽曲の拍子で決める. 拍子は MusicXML ファイルの beat タグと beat-type タグの値を用いて考慮させ, 1 小節毎に確認する. 小節内に拍子に関する情報がない場合は直前の小節と同じ拍子であると考慮させる. 各拍子の基本運動の図形を図 1 に示す.

指揮の速度は楽曲のテンポで決める. 楽曲のテンポは MusicXML ファイルの tempo タグの値で判定し, 1 小節毎に確認する. tempo タグが存在しない場合速度記号を参照する. 速度記号は word タグから抽出する. 表 1 は word タグの中身とテンポの対応表である. 小節内にテンポに関する情報がない場合, 拍子を決めるときと同様に, 直前のテンポと同じテンポにする.

Step:4 楽曲の音量を考慮

指揮を出力するにおいて動作の大きさも重要な要素である. 指揮者は楽曲の強さを強調するために大きな動作で指揮を振ることや, 静かさを表現するためにとても小さな動作で指揮を振ることがある. 本システムでは楽曲の音量によって

表 2: 各強弱記号と評価値の対応

強弱記号	評価値
p またはそれ以下	1
mp	2
mf	3
f	4
ff またはそれ以上	5

動作の大きさを決める。音量は楽曲の強弱とその時に演奏をしている楽器の数や特徴を用いて1小節毎に評価する。

楽曲の強弱は強弱記号を用いて評価する。強弱記号は MusicXML の dynamics タグから抽出することができる。各強弱記号と評価値の対応を表2に示す。強弱が弱いほど評価値は低くなり、強弱が強いほど評価値は高くなる。

楽曲の音量は強弱だけで決めることができない。例えば強弱記号が同じ *f* であってもフルートの *f* とトランペットの *f* を比べると音量はトランペットの *f* の方が大きい。音量を決めるためには楽曲の強弱だけでなく楽器の特徴も考慮して計算しなければならない。楽器別音量の違いを図2に示す。この値は文献 [11] を参考にした。強弱が *p* 以下の時はどの楽器であっても音量は同じであると考え、*f* の時は木管楽器：サクソ：弦楽器：ホルン：金管楽器 = 1 : 1.5 : 2 : 2 : 4 となる。

本システムでは各楽器の音量を先程の強弱記号の評価値を用いて次の式で表した。

$$\begin{aligned} wwValue &= ww \\ hornValue &= horn * (1 + DynamicsValue/4) \\ saxValue &= sax * (1 + DynamicsValue/8) \\ stringValue &= string * (1 + DynamicsValue/4) \\ brassValue &= brass * (1 + DynamicsValue/4 * 3) \end{aligned}$$

$wwValue$ は木管楽器の音量を求める際の式で、 ww は木管楽器が演奏していれば 1、していなければ 0 となる。 $DynamicsValue$ は強弱の評価値である。同様に $hornValue$ はホルンの音量、 $saxValue$ はサクソの音量、 $stringValue$ は弦楽器の音量、 $brassValue$ は金管楽器の音量を求める際に用いる。楽器の情報は MusicXML の part-name タグから抽出することができる。

楽曲全体の音量は以下の式で求める。

$$Volume = nowVolume / maxVolume$$

$nowVolume$ は対応する小節における各楽器の音量の総和である。 $maxVolume$ は対応する楽曲の編成においてすべての楽器が *f* で演奏しているときの音量の和である。 $Volume$ の値が 1 より大きくなった場合は 1 とし、楽曲の音量は 0~1 の間で数値化される。

Step:5 楽曲の雰囲気考慮

同じテンポ・拍子の楽曲であれば常に指揮の動作が同じというわけではない。指揮者は「元気な曲」「静かな曲」などといった楽曲の雰囲気によって指揮の図形や運動を変化させる。例えば同じテンポと拍子である場合元気な曲ではエネルギー的なイメージを与えるために加速・減速が大きい指揮

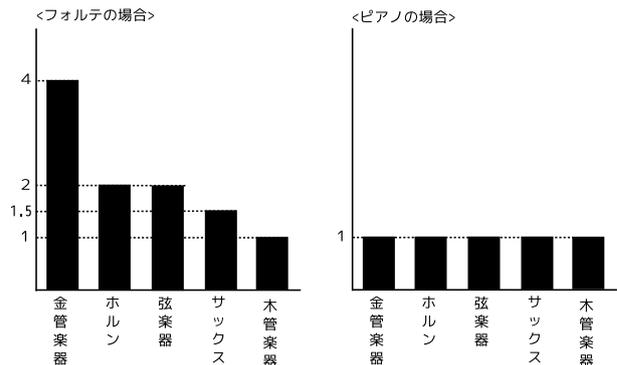


図 2: 楽器別音量

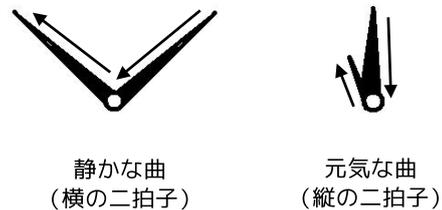


図 3: 楽曲の雰囲気による指揮の違いの一例

をすることが多いが、静かな曲では穏やかなイメージを伝えるために加速・減速がほとんどない緩やかな指揮をすることが多い。また元気な曲ではキレのある動きで表現するために縦長の図形を書いて指揮をするのに対し、静かな曲では柔らかさを表現するために横長の図形を描いて指揮をすることが多い(図3)。このように楽曲の特徴を考慮した指揮を実現するためには楽曲の雰囲気をどのように考慮するかが重要になる。

感情の表現方法に関して広く知れ渡っている方法の一つに感情を3次元で表現する方法がある[9]。この手法では感情は Valence と呼ばれる「肯定 (positive), 否定 (negative)」を表す次元, Arousal と呼ばれる「アクティブか, 落ち着いているか」を表す次元, Dominance と呼ばれる感情的な調整の度合いを示す次元から成り立つ。Dominance はしばしば無視され二次元で表現される(図4)。juslin ら [10] は Valence と Arousal を用いて、楽曲演奏における感情と音楽的特徴をまとめている(図5)。

本研究ではこの感情を二次元で表現する方法を用いて楽曲の雰囲気を考慮させる。また本研究では Valence については考慮せず Arousal の値にのみ着目する。Arousal の値は指揮の加速度や軌跡に大きく影響を与えるため楽曲の雰囲気を考慮した指揮を実現するために必要不可欠である。それに対して Valence の値は指揮の動作に直接影響を与えない。「肯定, 否定」という次元は音楽で表すと「明るい曲か, 暗い曲か」に相当する。指揮者は楽曲の明るさを自分の顔や左手(指揮棒をもっていない手)を用いて表現する。本システムでは指揮棒の軌跡を点で表したアニメーションで出力しており、Valence はそれらに直接影響を与えないため本研究では Arousal のみに着目する。

本システムでは Arousal の値を楽曲のテンポ、音の長さ、音量を用いて計算する。各パラメータは juslin らの研究結果と音楽理論をもとに 0~10 の間で設定した。テンポのは速

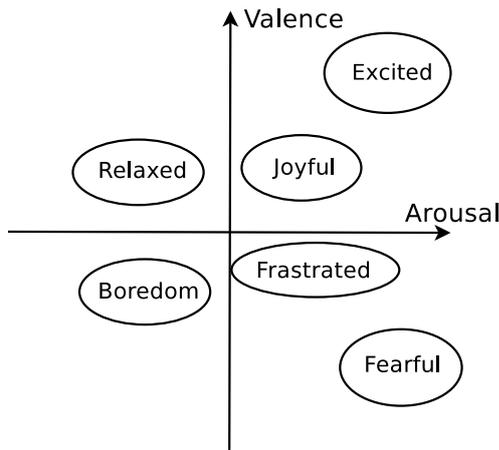


図 4: Valence-Arousal の概念

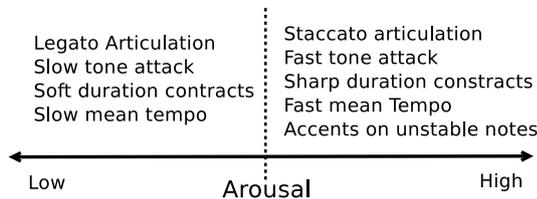


図 5: Juslin らによる Arousal の値と音楽的特徴のまとめ

いほどパラメータが高くなり、遅いほどパラメータが低くなるように設定する。各テンポとパラメータの値の対応は表3に示す。音の長さは短いほどパラメータが高くなり、長いほどパラメータは低くなる。音の長さのパラメータは以下の式で求め、値が 10 以上の場合になった場合は 10 と設定する。

$$DurationValue = ROUNDUP(4 \text{ 拍間にある音符の数} * 0.625)$$

音量は Step:4 で求めた音量を用いて以下の計算式を適用する。音量は大きいほどパラメータが高くなり、小さいほどパラメータは低くなる。

$$VolumeValue = ROUND(Volume * 10)$$

これら 3 つのパラメータを用いて以下の計算式で Arousal を計算する。

$$Arousal = (TempoValue * 0.03 + DurationValue * 0.02 + VolumeValue * 0.05) * 2 - 1$$

この重みの比率は実際に存在する曲の断片を約 200 集め、それらに様々な重みを適用し妥当と思われるものを採用した。妥当であるかは本論文の執筆者である笹川の主観で判断した。笹川は吹奏楽を中学時代から大学まで合計 10 年経験しており、担当楽器はフルートであった。また大学時代には 3 年ほど指揮者を経験している。楽曲の断片は複数の楽曲から様々な表情のものを 1 小節取りあげた。Arousal の値域は -1 ~ 1 である。

表 3: 各テンポと TempoValue の対応

テンポ	TempoValue
160 以上	10
144 ~ 159	9
132 ~ 143	8
120 ~ 131	7
108 ~ 119	6
92 ~ 107	5
88 ~ 91	4
76 ~ 87	3
60 ~ 75	2
54 ~ 59	1
53 以下	0

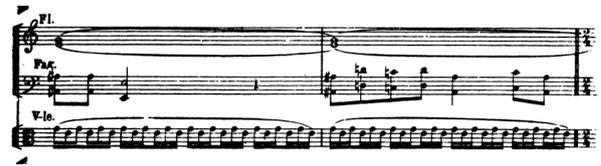


図 6: 組曲「展覧会の絵」より”バーバ・ヤガの小屋” (中間部)

本システムでは同じ拍子の曲でも Arousal の値に応じて三種類の振り方に分ける。Arousal の値が -1 ~ 0 の時は動きが緩やかな指揮を出力する。加速・減速があまりなく、指揮の軌跡が横長を描く特徴がある。Arousal の値が 0.5 ~ 1 の時はキレの良い指揮を出力する。加速・減速が大きく、指揮の軌跡が縦長に描かれる特徴がある。Arousal の値が 0 ~ 0.5 の場合、緩やかな指揮とキレのよい指揮の中間の指揮を出力する。

Step:6 目立たせたい部分の特徴

楽曲の雰囲気のみを考慮するだけでは楽曲の特徴を完全に考慮できなかったとは言えない。例えば図 6 はバーバヤガの小屋（展覧会の絵より）の中間部である。このフレーズでは静かに流れるような伴奏の上にファゴットが軽快なメロディーがある。楽曲の雰囲気のみを考慮するならばこの曲は「静かな曲 (=Arousal が低い曲)」と判断され、緩やかな指揮を出力する。しかしもしこの曲で軽快なファゴットを目立たせたい場合キレのあるはっきりした指揮を出力すべきである。このことから楽曲の雰囲気だけでなく、注目したい部分の特徴も指揮に影響を与えることが分かる。

このような問題を解決するために、本システムではユーザーに楽曲の分割点とその分割点までに注目したい部分を入力を求める。分割点は小節単位で入力する。システムは入力情報をもとに対応する部分の音符を一つずつ読み込み、その音符が (1) 何小節目の音符か、(2) どの部分の音か、(3) 何拍目の音か、(4) 音の長さはどれくらいか、(5) 休符かどうか、(6) 表拍か、裏拍か、(7) 強弱はついているか、(8) アーティキュレーションの指定があるかをチェックする。

すべての音符をチェックした後、以下のルールに従って指揮に修正を加える。

- 拍の頭がスタッカート、アクセント、ピチカートの場合は拍点を通常より下に打ち運動の加速・減速を大きくする。
- 各パーティションの境目は Arousal が低くても 1 拍目は拍点を通常より下に打ち運動の加速・減速を大きくする。
- クレッシェンドの頂点でかつ音量が f 以上の場合は運動の軌跡と加速・減速度を修正し通常より拍点をはっきり打つ。詳しい運動の軌跡や加速・減速については拍子、Arousal の値、その音符が何拍目かによって決める。
- 四分休符以上の長さの休符がある場合次にある音の 1 拍前の軌跡と加速・減速度を修正し、拍点を通常よりはっきり打つ。詳しい運動の軌跡や加速・減速はその時の拍子とその音符が何拍目かによって変化する。
- 八分休符と八分音符が連続する場合その拍の拍点を通常より下に打つ。
- 八分音符より短い音符が連続していかつアーティキュレーションがスラーでもテヌートでもない場合、Arousal が一段階上がった時の振り方になる。

これらのルールは 1 拍ごとに判定する。

Step:7 指揮アニメーションの出力

指揮の軌跡、速さ、加速・減速が決定した後、それらを適用したアニメーションを出力する。本システムでは HTML5 と CSS3 を用いて出力する。ユーザーは出力された HTML5 ファイルを開くことで指揮のアニメーションを見ることができる。指揮のアニメーションは指揮棒の軌跡を点を用いて二次元平面で表現する。出力されるアニメーションは Step:3 で抽出したテンポ情報を用いて運動の速さを考慮し、拍子情報を用いて指揮の基本的な軌跡を考慮する。また Step:4 で計算した音量の情報から指揮の運動の大きさを考慮する。Step:5 で計算した楽曲の Arousal や Step:6 で求めた目立たせたいパートの特徴は指揮の運動の加速度の決定や軌跡の修正に用いられている。

4 評価

4.1 動作例とその考察

この節では動作テストを行いその結果の報告及び考察をする。動作テストでは異なる特徴をもった同じ拍子・テンポの楽曲にシステムを適用し、出力された指揮のアニメーションについて考察する。

4.1.1 Arousal の値による出力結果の違い

図 7 は Arousal の値による指揮の変化を示している。Arousal の値は指揮の軌跡と加速度に影響を与える。二つの図はテンポが四分音符 120 で拍子が 2/4 拍子である楽曲に対する本システム指揮の軌跡であり、左は Arousal が 0.1 で右の Arousal は 0.3 である。楽曲のテンポはどちらも同じ値である。この 2 つを比べてみると Arousal が高い場合縦長の動作をしている。それに対し Arousal が低い場合は横長の動作になっていることが分かる。また本論文からでは読み取ることができないが同じテンポでも左に指揮に比べ

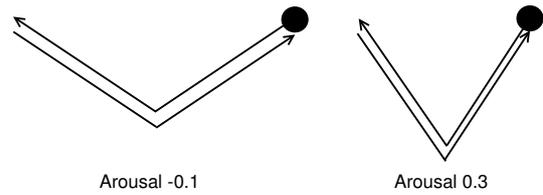


図 7: Arousal の値による出力の違い

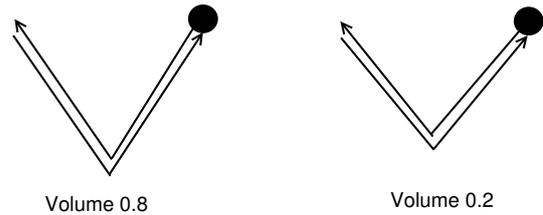


図 8: 音量による出力の違い

右の指揮の方が加速・減速の変化が大きくなっている。このことから楽曲の雰囲気や考慮して指揮が変化していることが確認できる。

4.1.2 音量の違いによる出力結果の違い

図 8 では音量の違いによる動作の変化を示している。音量は指揮の大きさに影響を与える。二つの図はどちらも四分音符 120 のテンポで拍子が 2/4 拍子である楽曲に対する本システムの出力結果であり、左は音量が 0.8 であり、右は音量が 0.2 の場合の出力結果である。動作の大きさは音量が大きい左の指揮は大きく、音量が小さい右の指揮は小さくなっている。この図では拍子や Arousal が同じである場合指揮の形は変化しないが、音量によって動作の大きさが変わっていることが確認できる。

4.1.3 異なるパートの特徴における出力結果の違い

図 9 では考慮したパートの特徴による違いを示している。パートの特徴は 1 拍毎に評価し指揮の軌跡と加速度に影響を与える。どちらの図も同じテンポ、Arousal である 2/4 拍子の指揮であるが左は 2 拍ともアーティキュレーションがテヌートである。この場合指揮の図形が横長で、加速・減速がゆるやかに出力される。これはテヌートで音を短くせず最後まで保つことを強調している。対する右の図は 1 拍目がスタッカートである場合である。この場合 1 拍目が縦長で 2 拍目は 1 拍目よりも打点が高くなっている。また 1 拍目の方が 2 拍目よりも加速・減速の変化が大きく出力される。これは 1 拍目のスタッカートをより強調するためである。このことから本システムでは注目したパートの特徴によって指揮の動作が変化することが確認できる。

4.2 改善点

前節では本システムが楽曲のテンポ・拍子・雰囲気・音量・パートの特徴を考慮した指揮を出力することを確認した。しかし本システムには改善すべき点もある。本研究では約 200 の既存の曲の断片を用いて Arousal の計算式を決めている。

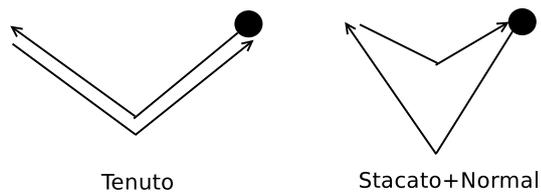


図 9: パートの特徴による出力の違い

現在数えきれないほどの楽曲が存在する中でこの実験データの数は決して多いとは言えない。そのためこれからより多くの楽曲に本システムの計算式を適用し、この計算が妥当であるか考察する必要がある。また Arousal の計算式のみではなく、本システムをたくさんの楽曲に適応し動作結果を考察する必要もある。

本システムで考慮できていない指揮の技法もいくつかある。例えば本システムでは指揮の分割やアウフタクトの合図などの応用技法を考慮されていない。これらの技法は演奏者のタイミングを合わせるためには重要な技法であるので今後考慮させていきたい。また本システムは1小節毎に拍子とテンポの情報を抽出している。そのため *rit.* や *accel.* など1小節の中でのテンポの変化に対応していない。これらの記号は実際の楽曲にもよく使われるため、考慮させなければならない。

アニメーションについても改善点がある。指揮者は指揮棒だけでなく顔の表情や左手を使って音楽を表現することがある。それに対し本システムで出力される指揮のアニメーションは指揮の軌跡のみである。またそのアニメーションも動きが滑らかでないため指揮者が実際に行う動作と比べて加速・減速が分かりづらくなっている。

5 まとめ

本研究ではコンピュータが指揮のアニメーションを出力するシステムを実装した。本システムは性質上、音楽の知識が深くない人が使うケースが想定されるため、できるだけ簡単な入力のみで出力を得られるように設計することを前提とし、その上でできるだけ楽曲の特徴を考慮した指揮を出力させることを目的とした。

楽曲のテンポ・拍子を参照することで指揮の基本的な図形や速さを考慮することができた。また楽器情報と強弱を分析することで楽曲の音量を計算し指揮の大きさを考慮することができた。楽曲の雰囲気は楽譜に書かれていない情報ではあるが、楽曲の特徴をパラメータ化し Arousal という値を求めることにより、雰囲気に応じた三種類の振り分けができるようになった。ユーザーに楽曲の分割点とその小節まで目立たせたいパートの入力を求め、その情報をからルール(規則)を用いて指揮の修正を加えることで、注目したいパートの特徴を考慮することができた。

同じ拍子・テンポで異なる特徴をもつ楽曲を用いて動作テストを行い、本システムでは楽曲のテンポ・拍子・雰囲気・音量・アーティキュレーション等パートの特徴を考慮した指揮を出力できることを確認した。

今後はより表現豊かな指揮を実現するためにシステム修正及び拡張していきたい。本システムでは出力されたアニメーションに問題があることや、拍の分割など考慮されていない指揮のテクニックがたくさんある。また現状では実験データが少ないためこれからサンプルを増やしていき現状のシステムをより考察する必要がある。今後はこれらの問題を解決さ

せることでより楽曲の特徴を考慮した指揮を実現させていきたい。

参考文献

- [1] Max V. Mathews. The Radio Baton and Conductor Program, or: Pitch, the Most Important and Least Expressive Part of Music. *Computer Music Journal* Vol. 15, No. 4, Dream Machines for Computer Music: In Honor of John R. Pierce's 80th Birthday (Winter, 1991), pp. 37-46.
- [2] 宇佐 聡史, 持田 康典. マルチモーダル指揮シミュレータ. *日本フアジイ学会誌* Vol.10, No.4, pp.707-716(1998).
- [3] Teresa Marrin Nakra, Yuri Ivanov, Paris Smaragdhis, Chris Ault. The UBS Virtual Maestro: an Interactive Conducting System. NIME09, June 3-6, 2009.
- [4] 馬場 隆, 橋田 光代, 片寄 晴弘. 指揮者のヒューリスティックスを盛り込んだ指揮システム”VirtualPhilharmony”. *情報処理学会シンポジウム論文集*, 巻: 2010, 号: 4, pp.169-172(2010).
- [5] 森口 徳崇, 三浦 雅展. MIDI 楽曲を対象としたリアルタイム感情付与システム: MOR2ART. *日本音響学会音楽音響研究会資料* MA2009-35, pp.21-26 (2009).
- [6] Antinio Pedro Olivaira, Amilcar Cardoso. A musical system for emotional expression. *Knowledge Based Syst*, Vol. 23, No. 8, pp.901-913 (2010).
- [7] MusicXML, <http://www.musicxml.com/>
- [8] CrestMuse Toolkit, <http://cmx.sourceforge.jp/>
- [9] E. Daly, W. Lancee, J. Polivy. A conical model for the taxonomy of emotional experience. *Journal of Personality and Social Psychology* 45 (1983) 443-457.
- [10] P. Juslin, Communicating emotion in music performance: a review and a theoretical framework, *Music and Emotion: Theory and Research* (2001) 309-337.
- [11] 原田宏美, DTM で学ぶオーケストレーション入門, 音楽之友社.