

## 指揮者の動きに同期した演奏アニメーションの生成

## Generation of brass band animation synchronized with the movement of conductor's hand

小林 克樹<sup>†</sup>      村木 祐太<sup>†</sup>      西尾 孝治<sup>†</sup>      小堀 研一<sup>†</sup>  
 Katsuki Kobayashi   Yuta Muraki   Koji Nishio   Kobori Kenichi

## 1. はじめに

近年, PlayStationVR や HTC Vive, Oculus Rift などの VR デバイスが普及し, 手軽に仮想現実を体験することが可能になっている.

ところで, 音楽データの種類である MIDI データを入力とし, 楽器演奏アニメーションの自動生成を行う研究<sup>[1]</sup>がある. しかし, この研究では音源に同期したアニメーションを生成することを目的としているため, ユーザがアニメーションに対してアクションを起こすことができない. そこで現実のユーザの指揮者の動きと同期した楽器演奏アニメーションの生成を行うシステムを提案する. 本システムでは, HTC Vive を用いて実装を行う. ユーザの指揮者の動きを HTC Vive コントローラ(以下コントローラ)により取得し, 音楽のテンポ推定を行う. 推定したテンポを用いてアニメーションおよび音楽の再生速度を変更する. また, 仮想空間内に CG アニメーションの生成を行うことにより, ユーザに対して没入感を与えることが期待できる.

## 2. 提案システム

本システムは Unity を用いて実装を行った. 本システムのフローチャートを図1に示す.

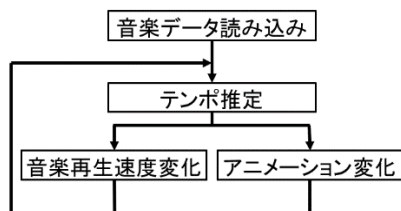


図1 提案システムのフローチャート

ユーザは HTC Vive を頭部に装着し, 右手にコントローラを所持し, 指揮者の動作を行う. HTC Vive を使用した理由は以下の3点である.

- 位置情報を取得できる点
- 速度ベクトルを取得できる点
- 仮想空間上にコントローラを表示できる点

これらの利点から HTC Vive が本システムに適していると考えた. 本システムではコントローラを指揮棒に見立てて仮想空間上に表示する.

## 2.1 音楽データ読み込み

まず, 音楽データの読み込みを行う. 本システムでは音楽データとして音楽データの一種である MIDI データを使用する. MIDI データには楽器の種類やテンポなどの情報が含まれており, これらの情報を使用することで演奏アニメーションの多様性を増すことが期待できる. また, MIDI データの読み込みには Unity 内で使用することのできるアセットの一種である「Midi Tool Kit Pro<sup>[2]</sup>」を用いた. このアセットは MIDI データを音源として再生することができる点や, 前述した MIDI データに含まれる情報を解析することができる点から本研究に適している.

MIDI データから取得した楽器の種類の情報をもとに演奏者の表示を行う. なお, 現在対応している楽器はフルート, ギター, クラリネット, トランペット, ドラム, ピアノの6種類である.

## 2.2 テンポ推定

次に, ユーザの指揮者の動きからテンポの推定を行う. 今回は最も曲数が多い4拍子の曲に注目し, ユーザの動作を定義した. ユーザの動作を図2に示す.

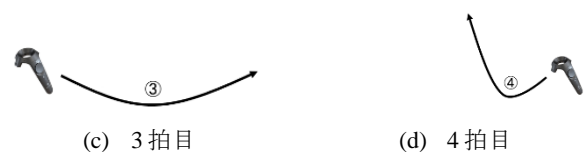
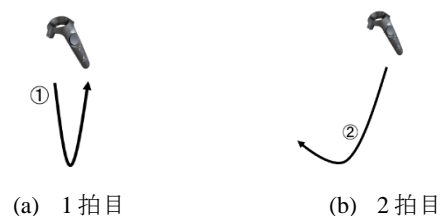


図2 ユーザの動作

同図に示すそれぞれの動作にかかった時間を一拍の実時間として取得する. ここで, 一拍の実時間とテンポの関係を式(1)に示す.

$$\text{一拍の実時間(s)} = 60(\text{s}) / \text{テンポ} \quad (1)$$

同式より, 以下の式(2)を得ることができる.

$$\text{テンポ} = 60(\text{s}) / \text{一拍の実時間(s)} \quad (2)$$

式(2)より、一拍の実時間を取得することでテンポの推定を行うことが可能であることがわかる。

一拍の実時間を取得するためには、図 2 に示した動作の開始地点を取得する必要がある。それぞれの動作に注目すると、開始地点の速度ベクトルとその次のフレームの速度ベクトルを比較した時、縦方向もしくは横方向の向きが反対になっているという特徴がある。これによりそれぞれの動作の開始位置を取得することができる。ただし、以下に記す地点は開始位置として取得しない。

- 速度ベクトルの向きが下から上に変化した地点。
- 開始位置として取得してから0.25秒経過していない地点。

前者の条件を設けた理由は以下の通りである。図 2(a)に示す 1 拍目の動作に注目する。この動作において最も下に移動した地点の速度ベクトルとその次のフレームの速度ベクトルを比較した時、下方向から上方向に変化していることがわかる。しかし、この地点は動作の開始位置として不適切である。この地点が動作の開始位置として取得されることを防ぐためである。

後者の条件を設けた理由として、わずかな振動によってその位置が動作の開始位置として取得されることを防ぐためである。また、0.25 秒とした理由として、式(2)より一拍の実時間を 0.25 秒とした場合テンポは 240 と推定される。そのため、指揮動作をこれ以上早くすることは現実的ではないと考えたからである。

### 2.3 音楽再生速度およびアニメーションの変更

最後に、推定したテンポを用いて音楽再生速度およびアニメーションの変更を行う。再生速度の倍率はユーザの動きから推定したテンポを MIDI データから取得したテンポで割ることによって算出する。

変化したアニメーションは HTC Vive を用いて仮想空間上に投影される。仮想空間上に投影されている様子を図 3 に示す。



図 3 本システムのイメージ

同図右上部分に存在する白い棒は指揮棒に見立てたコントローラである。これらの処理を MIDI データの再生が終了するまで繰り返し行う。

### 3. 実験と考察

ユーザの動きに応じて音楽再生速度およびアニメーションの変化が正しく行われているかを確認するために実験を行った。実験ではユーザの動きを、MIDI データから取得したテンポより早くした場合と遅くした場合、それぞれに対して実験を行った。

実験結果より、ユーザの動きに応じて音楽再生速度およびアニメーションの変化が正しく行われていることを確認することができた。また、1 小節分の動作に対して 4 拍分意図した位置でそれぞれの動作の開始位置が取得されていることも確認することができた。これらのことからユーザの動きから正しくテンポの推定を行うことができていくことがわかる。

その一方で、アニメーションが単調である点から違和感が生じていることがある。この問題に関して、現在 MIDI データに含まれるテンポ以外の情報を使用していないことが原因としてあげられる。また、指揮者の動作においても同様にテンポ以外の情報を使用していないことも原因として考えられる。そのため、テンポ以外の情報をアニメーション制御に用いることにより改善することができると考えられる。改善案を以下に示す。

- 指揮棒の振る大きさによって音の強弱やアニメーションの大きさを変更する。
- 1 拍目の動作が行われた時に、ドラムのアニメーションを大きくする。
- 指揮棒の動きに応じて演奏者が指揮者を見るアニメーションを追加する。

また、今回は 4 拍子の曲に注目したが、MIDI データには 4 拍子以外の曲も存在する。そのため、4 拍子以外の曲でのユーザの動作の定義を行う必要があるとともに、4 拍子以外の曲に関する実験を行う必要があると考えられる。そして、アンケートを用いて本システムの評価を行うことを考えている。

### 4. おわりに

本研究では HTC Vive およびコントローラを用いて、ユーザの指揮者の動きからそのテンポを推定し、推定したテンポを用いて音楽再生速度およびアニメーションの変更を行うシステムの開発を行った。今後の課題として、テンポ以外の情報を用いたアニメーションの改善や 4 拍子以外の曲に関する実験があげられる。また、アンケートを用いた評価実験も行う必要がある。

#### 参考文献

- [1] 武内航, 堀井絵里, 藤代一成. "吹奏アニメーションにおける音源同期型姿勢制御." 情報処理学会第 80 回全国大会講演論文集 2018.1 (2018): pp.315-316.
- [2] Midi Tool Kit Pro - Asset Store  
<[https://assetstore.unity.com/packages/tools/audio/midi-tool-kit-pro-115331?aid=1011Gbg&utm\\_source=aff](https://assetstore.unity.com/packages/tools/audio/midi-tool-kit-pro-115331?aid=1011Gbg&utm_source=aff)>

† 大阪工業大学 情報科学研究科  
〒573-0196 大阪府枚方市北山 1-79-1  
TEL.072-866-5301  
Email: kobayashi\_katuki@ggl.is.oit.ac.jp