

# リズムパターンのインタラクティブなグルーブ可視化

神 展彦<sup>†</sup>      芳賀 直樹<sup>‡</sup>      藤代 一成<sup>†</sup>

慶應義塾大学大学院 理工学研究科<sup>†</sup> 野村総合研究所<sup>‡</sup>

## 1 背景と目的

グルーブ (Groove) とは、演奏における音のうねり、リズムのノリや一体感などを表す音楽特徴のひとつである。グルーブは、音楽の他の要素である音量や音程などの単純なものに比べてより人間の感情に訴えかける、よい演奏に必要な不可欠な要素のひとつである。音を形状や色にマッピングして音楽データの構造や特徴を表す画像を生成する音楽可視化は、ユーザのさまざまな音楽活動を支援する。そのなかでもグルーブ可視化は、音楽的センスや聴力の個人差によらないグルーブの習得や音楽経験に制限されない音楽の感動の共有など、さまざまな応用につながる。

Ohyaらは、グルーブと拍のずれの関係を可視化した[1]。しかし、その可視化結果はユーザの情報可視化に関する知識を前提としており、直感的な結果の読取りが困難である。本研究の先行研究である SeeGroove[2]では、3次元図形を用いて直感的なグルーブ可視化を行った。しかし、SeeGrooveでは図形の制御変数が1つであり、可視化できるリズムパターンの多様性が大きく制限されていた。また、入力はMIDIファイルのみでありインタラクティブ性はなかった。

そこで本研究では、MIDI信号として入力された音楽をインタラクティブに図形に変換し、音楽知識に乏しいユーザのグルーブ習得を支援するための直感的なリズム可視化システム SeeGroove2 を提案する。

## 2 SeeGroove2 の概要

グルーブの差異や特徴を図形として可視化することで、グルーブを直感的に捉えることが可能なるが[2]、本システムではさらにインタラクティブな可視化を目指す。大まかなシステムのプロセスループを図1に示す。

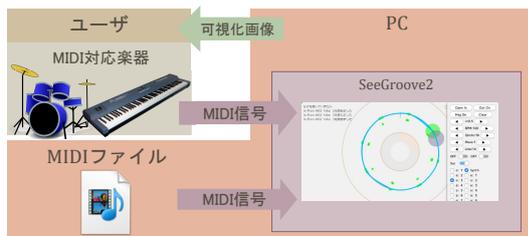


図1: プロセスループ。演奏しながら結果が可視化され、即時的に演奏のグルーブを確認することができる。

Interactive Groove Visualization of Rhythm Patterns  
Nobuhiko Jin<sup>†</sup>, Issei Fujishiro<sup>†</sup>, Naoki Haga<sup>‡</sup>  
Graduate School of Science and Technology, Keio University<sup>†</sup>  
Nomura Research Institute, Ltd.<sup>‡</sup>

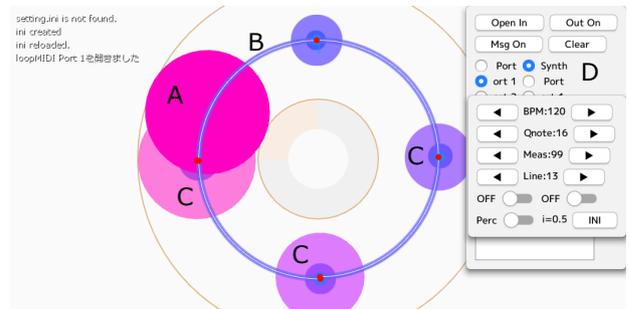


図2: SeeGroove2のUI。A: 現在位置。B: グルーヴの形。C: 演奏された音。D: 設定パネル。

SeeGroove2は、MIDI (Musical Instrument Digital Interface) と、その録音ファイルであるMIDIファイルの入力に対応する。MIDIは波形に比べ発音のタイミングが明確であるため情報を直接扱うことができる。電子ドラムやキーボード、電子クラリネットなどのMIDI規格に準拠した楽器が本システムの入力になり得る。

### 2.1 SeeGroove2のUI

ドラムをはじめとしたリズム楽器は、数小節から数十小節にわたり同じようなリズムを繰り返して刻む。この繰り返しを直感的に表現するために、SeeGroove2はリズムが「回る」ように円状に表現する。

図2に SeeGroove2 の UI を示す。A は現在の位置を示し、この円はテンポに合わせて周回する。その一周分の軌跡 (B) が演奏されたリズムのグルーブとして示される。演奏された音は円 (C) として残る。設定パネル (D) では、テンポや分析の基準となる音符の指定や、MIDI ポートの選択を行う。

### 2.2 図形の描画

SeeGroove2では、定量的な描画を行う。発音信号を受信すると、まず受信時刻から音符の座標点  $P_i(r_i, \theta_i)$  を決定する。 $\theta_i$  は1小節あるいは任意の小節をかけて  $2\pi$  進む。 $r_i$  は式 (1) で表される。

$$r_i = r_{BASE} + r_{di}f \quad (1)$$

ここで、 $r_{BASE}$  は基準半径、 $r_d$  は基準となる分音符に最も近い拍からの時間的割合、 $f$  は表現のための係数である。図3に8分の8拍子で8分音符を8つ演奏したときの座標を示す。式 (1) により、拍に対して早く演奏された音符はAのように内側に引き付けられ近道をしたような表現になる。また、拍に対して遅く演奏された音符はBのように外側に寄り道をしたような表現になる。次に、点を曲線で結び、グルーブの形を描画する。現在、

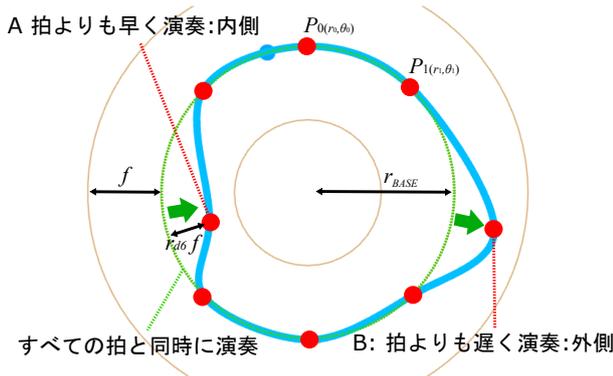


図3: 演奏されるタイミングと座標.

SeeGroove2 では一周のうちに演奏される音符の数に十分な 256 点を扱うが、これらの点  $P_0 \sim P_{255}$  を直角座標系に変換し、各点を 3 次スプライン補間曲線で補間することでグルーブの形として表示する。直感的には、内側に凹んだ部分では「早出し」のグルーブ、外側に膨らんだ部分は「タメ」のあるグルーブとなる。また、演奏された音 (図2の C) は時間の経過により大きさを小さくし、色を赤から青色に変化させ、音の余韻を表現する。

### 2.3 インタラクティブ性

音楽において、インタラクティブ性は重要である。演奏しながらフィードバックを得られることで、即時的な確認が可能になり、グルーブの表出スキルの効率的な学習が可能になる。演奏における入力処理の遅延は致命的であり、時間分解能を確保するため 1,920 Hz 程度の処理が求められる。また、描画は 60 FPS で行うことが望ましい。

入力処理を図4のように座標計算や描画処理から完全に分離させることで、時間分解能を確保する。まず、MIDI 信号を受信した際に受信時刻の記録など最低限の低負荷な処理を行い、スタックに格納する。その直後に外部音源に対して MIDI 出力を行い、音を鳴らす。そして、これとは異なるフローで MIDI 信号をスタックから取り出し、座標計算や補間曲線の再計算、円の縮小と色変化、描画を行う。計算や描画の高負荷フローに時間がかかったとしても、低負荷フローでフェッチされた情報をもとに処理するために、インタラクティブ性だけでは再現性も確保され、高速動作した場合と同一の図形を描画することができる。多数の座標計算は複数スレッ

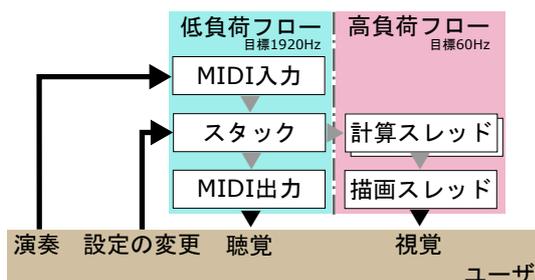


図4: フローの分離。描画処理の頻度や重さにかかわらずインタラクティブ性と再現性を保つ。

ドによって高速化し、描画部分では描画用 API として Siv3D[3] を導入し、円と補間曲線を計算資源に優れたグラフィックスハードウェアを用いて高速に描画する。

### 3 可視化結果

3 種類のリズムパターンを打ち込み、SeeGroove2 をラップトップパソコン (CPU: Core i5 2410M, メモリ: 4 GB, 内蔵グラフィックス: Intel HD Graphics 3000) 上で動作させ可視化した結果を図5に示す。テクノのリズム (a) では、すべての音符が正確なタイミングで演奏されるために真円で表現された。ジャズのリズム (b) では、特有のスイングで基準のタイミングより遅く演奏される音符があるため外に膨らんだ形として表現された。ウィナワルツのリズム (c) では、2 拍目が早く演奏される独特の特徴が表現されている。システムは 60 FPS で動作した。

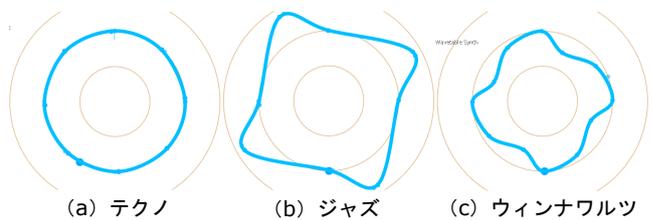


図5: さまざまなリズムパターンの可視化結果。(a) 正確なリズムのため真円。(b) スイングのため膨らんだ形。(c) 独特なリズムにより凹んだ形。

### 4 結論と今後の課題

演奏された音符それぞれに対して点をプロットし、滑らかに補間することにより、リズムパターンを多様な形に表現することができた。また、フロー分離によってインタラクティブなグルーブの可視化を行うことができた。

本システムでは主にタイミングに注目したが、音量やテンポの変化、また複数人の演奏などグルーブの他の要素についても検討していくことで、さらに有効な支援が可能になると考えられる。

### 謝辞

本研究の一部は、平成 27 年度科研費新学術領域計画 研究 25120014 の支援により実施された。

### 参考文献

- [1] Y. Ohya, K. Nakamura, and T. Tokunaga: "Extraction of Groove Feelings from Drum Data using Non-Negative Matrix Factorization," in *Proceedings of the Joint 6th International Conference on Soft Computing and Intelligent Systems, and the 13th International Symposium on Advanced Intelligent Systems*, pp. 125-130, 2012.
- [2] I. Fujishiro, N. Haga, and M. Nakayama: "SeeGroove: Supporting groove learning through visualization," in *Proceedings of Cyberworlds2015*, pp. 189-192, 2015.
- [3] C++ ライブラリ Siv3D <http://play-siv3d.hateblo.jp/> (最終アクセス日: 2016/01/06).