

音楽表現を拡張するジェスチャ操作楽器の開発

紫村勇綺^{†1} 宮脇健三郎^{†2}

概要: 音楽表現に使用される楽器は、時代と共に新しい物が開発されている。しかし、これらの楽器にはその構造や発音方法から音楽的表現に奏法上の制限が存在する。シンセサイザー等の電子楽器はそれら制限を一部克服しているが既存の楽器をシミュレートしているためすべての制限を超えるものではない。そこで本研究ではセンサで指先の位置を検出し、それに応じた音程・音量を出力する直観的奏法が可能な楽器を開発する。具体的にはユーザーがセンサに手をかざすと画面上に手のモデルが投影される。ユーザーはその画面上のモデルを見ながら手指を移動させ、オブジェクトに触れるとそれに応じた音が発音されるものである。

Development of Gesture Instruments Extend Music Expression

YUKI SHIMURA^{†1} KENZABURO MIYAWAKI^{†2}

Abstract: New musical instruments are being developed with the times. But these instruments have problems in rendition style. Although electronic musical instruments such as synthesizers overcome some of these limitations, they are not beyond all limits to simulate existing instruments. In this research, we develop an intuitive rendition style instrument that detects the position of the fingertip with a sensor and outputs the pitch and volume accordingly. When the user holds his hand over the sensor, the model of the hand is projected on the screen. User moves the fingers while watching the model on the screen, and when a user touches the object, a sound corresponding to it is sounded.

1. はじめに

音楽表現に使用される楽器は、時代と共に新しい物が開発されている。ギターのような発弦楽器やバイオリンのような擦弦楽器、ピアノの様な鍵盤楽器、トロンボーン等の管楽器等様々な構造の楽器が開発された。また 1900 年代にテルミンが開発されてからはシンセサイザー等の多種多様な電子楽器が開発された。

しかし、アコースティックやエレクトリックに関わらず、これらの既存の楽器にはその構造、発音方法から、音楽的表現に奏法上の制限が存在する。具体的には、トロンボーンのような管楽器は単一の音しか発音出来ず、和音を演奏する事は不可能である。また、ギターやピアノ等の打弦、発弦によって発音する楽器は、サステイン（音の長さ）を完全に制御することは出来ない。ピアノは複雑な和音に対応できるが、演奏のし易さは演奏者の手の大きさにより左右される。キーボードやシンセサイザーなどはこれらの問題の一部を解決しているが、インターフェースがピアノ、吹奏楽器等、既存の楽器をシミュレートしているため同じ問題が存在している。

そこで、本研究ではセンサで指先の位置を検出し、それに応じた音程、音量を出力する直観的奏法の楽器を開発する。具体的には、ユーザーがセンサに手をかざすと画面上

に手のモデルが投影される。ユーザーはその画面上のモデルを見ながら手指を移動させ、オブジェクトに触れるとそれに応じた音が発音される。

本研究のコンセプトは、新たな操作方法の楽器を開発することである。今回開発する楽器は他の楽器と合奏できるような、音楽的アンサンブルの一員としての位置づけを想定している。目標は、演奏者が演奏する曲に合わせて使いやすいうようにカスタマイズできる物を開発することである。また新たな奏法の楽器の開発によってユーザーが新たな音楽的表現を創造できること、それに対応できる物にすることを目指している。コンセプトユーザーはあくまで音楽的知識のある程度備えた人物である。

2. 関連研究

石川らは 3 次元位置検出センサを用いてのジェスチャ制御による直感的操作が可能な楽器「PlaytheDE」を開発した [1]。これは gitEye3D という簡易モーションキャプチャシステムを用いて球状のマーカーの 3 次元位置を計測し、それに応じた音高、音量を出力する物である。音の出力には MIDI 規格を使用しており、半音単位での音程操作が可能であり、この点は本研究と同様である。クロマチックな音程の他に、ブルースに用いる特別なスケールのみに対応し

^{†1} 大阪工業大学大学院 情報科学研究科
Graduate School of Information Science and Technology
Osaka Institute of Technology

^{†2} 大阪工業大学 情報科学部
Faculty of Information Science and Technology, Osaka Institute of
Technology

たパターンも演奏可能である。この関連研究がマーカーの操作による制御であるのに対し、本研究ではユーザーの手指による直接的な制御である。また、文献[1]はシステムの開発目的として、音楽や楽器の初心者が簡単に演奏できる楽器を目指していることに対し、本研究は音楽家が本格的な演奏会で利用できる楽器の開発を目的としている。他の類似システムの事例としては Leap Motion [https://www.leapmotion.com/]を MIDI コントローラとする AeroMIDI[2]がある。本研究と同じセンサを使用しているが、これは MIDI 対応のソフトウェアのパラメータやエフェクトを操作するものであり、音符の再生に特化したものではないため、楽曲の演奏そのものには適してはいないと考えられる。

3. 提案手法

提案システムではゲームエンジン Unity[3]を用い、スクリーン上の仮想 3 次元空間に発音オブジェクトを配置する。次に手指のモーションキャプチャセンサ(Leap Motion)が手指の位置を検出し、その手指の位置にオブジェクトを追従させる。発音オブジェクトと手指オブジェクトが交差した際に、それに応じて MIDI[4]データを用いて発音する。

3.1 操作方法

図 1, 2 に本楽器を実際に操作するユーザーとそのディスプレイに映る画面を示す。画面左側の円柱オブジェクトは触れると音を発する発音オブジェクトであり同時に音色の変更も行うことが可能である。右側に映っている黄色の矩形オブジェクトは発している音の音高・音量を操作するオブジェクトである。このオブジェクトは図 3 に示すように触れる位置によって音量・音高を操作する。

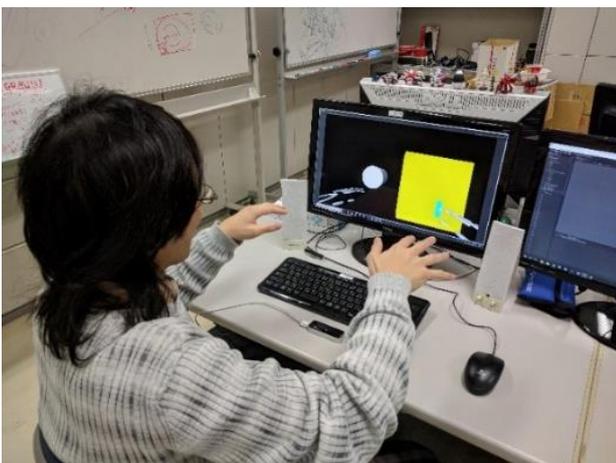


図 1 本楽器を操作するユーザーの様子

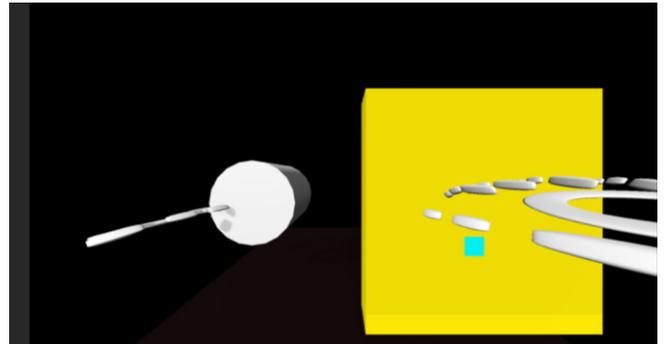


図 2 操作時の画面

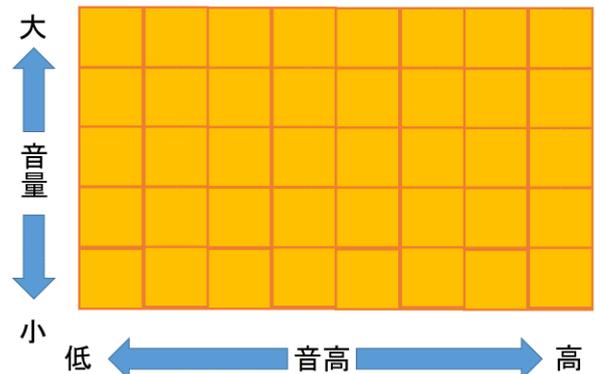


図 3 音高・音量を操作するオブジェクト

これらのオブジェクトはユーザーが予め自由に配置することが可能である。音色変更のオブジェクトを複数配置すれば演奏中に様々な音色に変更することが可能となる。また、発音オブジェクトを重ねることによって和音の生成が可能となる。

3.2 発音オブジェクトのパラメータ

発音オブジェクトには発音する音色や音高の変化幅など詳細なパラメータが設定可能である。例を図 4 に示す。これらの数字はすべて MIDI データの各種マップに対応している。Note は最初に発音する音高で 72 は C5 (“ド”の音)を表し、Vel は MIDI パラメータの velocity であり、発音時の音量に相当する。Inst は音色(1 はピアノ)をそれぞれ表している。これらのパラメータを変更することで目的の音を生成する。

Master		Main Camera
Note	72	
Vel	0	
Inst	1	

図 4 発音オブジェクトのパラメータ

4. システム構成

ゲームエンジン Unity を用い、スクリーン上の仮想 3 次元空間に発音オブジェクトを配置する。次に手指のモーションキャプチャセンサ(Leap Motion)が手指の位置を検出し、

その手指の位置にオブジェクトを追従させる。発音オブジェクトと手指オブジェクトが交差した際に、それに応じた発音を行う。

4.1 ハードウェアの構成

このシステムを実現するために使用する開発環境を、図5に示す。Leap MotionはPCにUSB接続している。

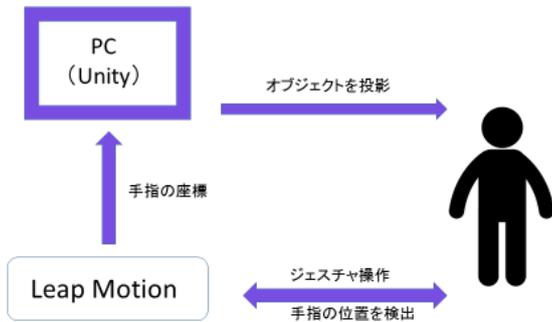


図5 ハードウェアの構成

4.2 ソフトウェアの構成

提案システムを実現するため、ゲーム開発用プラットフォームであるUnityを使用した。開発にUnityを使用した理由は、オブジェクト間のコリジョン判定が容易に実装できる点と、Leap Motionに対応したSDKが公開されており、Leap Motionとの連携が容易であるためである。

また、提案システムでは、音楽的音声再生の際にMIDIデータを使用している。MIDIデータはPC上で音声を発音するための規格であり、「発音」、「発音する音の高さ」、「発音する音の音量」などの情報を持ち、それらを操作できる[4]。MIDIデータに応じた音声に指定した音源から発音することで、音楽的音声の再生が可能である。

表1 音色マップ

No.	instrument
1~8	Piano
9~16	Chromatic Percussion
17~24	Organ
25~32	Guitar
33~40	Bass
41~48	Strings
49~56	Ensemble
57~64	Brass
65~72	Reed
73~80	Pipe
81~88	Synth Lead
89~96	Synth Pad
97~104	Synth Effects
105~112	Ethnic
113~120	Percussive
121~128	Sound effects

このシステムでは、互換性の観点からMicrosoft Windowsに標準搭載されているMSGs (Microsoft GS Wavetable SW Synth)という音源を使用している。この音源はGM規格に

準拠している。そのため本システムの生成できる音色は表1の128種類である。

4.3 UnityにおけるMIDIデータの操作

本研究では、前述の通りMIDIデータを使用して音の出力を行っている。今回は文献[5]を参考に、スクリプト内でWindows APIからMIDIデータを操作するため、winmmライブラリ[6]を使用している。音高の設定には、半音以下のインターバルを表現するために元々の音高設定ではなくMIDIコマンドのピッチベンドを用いている。ピッチベンドは本来弦楽器のバンドアップやスライド等の一時的な状態変更を行う機能であるが、本システムでは目的である無段階の音高変化を表現するために使用している。これにより最大1/8192音の細かい音高変化が可能である。音量の変化にはMIDIコマンドのボリュームチェンジを使用している。このボリュームは発音時に決まるVelocityとは異なり、発音後も音量を操作できる。ボリュームチェンジでは音量を128段階で設定できる。

5. 動作確認

開発した楽器を実際に使用し、既存の楽器の奏法上の制限を超えた表現が可能であるかを調べた。具体的には、複雑な和音の演奏、複雑なフレーズの演奏が可能かを確認した。

5.1 複雑な和音の演奏

ピアノやギター等は複数の音を発する和音の演奏が可能であるが、鍵盤や弦の構造から制限が存在し、演奏不可能な和音が存在する。そこで、複数の発音オブジェクト全く同じ位置に並べ演奏したところ、作成した楽器でのみ演奏可能であることが確認できた。

5.2 複雑なフレーズの演奏

楽曲は単音の組み合わせであるメロディで構成されているが、音高順に並んでいる鍵盤では演奏が困難な場合がある。そこで、演奏する曲中で使用する音を演奏する順に並べたところ、どのようなメロディであっても同様に演奏できることが確認できた。

6. おわりに

楽器はアナログやデジタルを問わず、全く発音方法や奏法の異なる物が世界で考案、開発されている。その技術が現代の音楽的表現にマッチした物もあれば、一見有用性に乏しい物も存在する。例えば、エレキギターの音を加工するエフェクターの中に、オーバードライブと呼ばれる種類のエフェクトが存在する。これはアンプの音量を上げすぎた

ことによる音質の劣化であったが、その音色は現代音楽のスタンダードとなっている。このように、新たな楽器により新たな音楽が生まれる。逆に言えば音楽は楽器の奏法内でしか生まれないと考える。

本研究で開発した楽器は複雑な和音や複雑な音量調整等が可能である。これらの機能を用いることで新たなスタイルの楽曲が創造されるきっかけになると考えている。

今後は、演奏中の音色切り替えの拡張やオブジェクト配置の簡略化など、インターフェースの改良と共に、エフェクトの追加などの機能追加をする予定である。

謝辞 本研究は JSPS 科研費 17K01160 の助成を受けたものです。

参考文献

- [1] 石川修, 平井重行, 金森務, 片寄晴弘, 井口征士. ジェスチャによる楽器 PlaytheDE. 情報処理学会インタラクシオン 2000, 2000, p.161-162.
- [2] “AeroMIDI HP”. <http://www.aeromidi.net/index.php>
- [3] “Unity 公式 HP”. <https://unity3d.com/jp/>
- [4] “MIDI 企画委員会 HP”.
<http://amei.or.jp/midistandardcommittee/index.html>
- [5] 井上謙次. Windows API による MIDI プログラミング. 2005,
http://www.deqnotes.net/midi/winapi_midiprogram/winapi_midiprogram.pdf
- [6] “Musical Instrument Digital Interface (MIDI)”
[https://msdn.microsoft.com/ja-jp/library/windows/desktop/dd743619\(v=vs.85\).aspx](https://msdn.microsoft.com/ja-jp/library/windows/desktop/dd743619(v=vs.85).aspx)