

ハンドベル自動演奏のための 3次元操作インタフェースの開発

近畿職業能力開発大学校 電子情報技術科 奥田佳史

Development of 3D Interface for Hand-Bells Performance Robot

Yoshichika OKUDA

要約

電子情報技術科総合制作課題テーマとして「ハンドベル自動演奏のための 3次元操作インタフェースの開発」の立案及び指導を行った。センサとして非接触型 3次元ポインティングデバイスを採用し、指の位置情報からリアルタイムにハンドベル自動演奏装置を制御するマン=マシンインタフェースの構築を目指した。マルチコア機能とマルチタスク同期機能により並列処理を実現し、指揮に従い演奏中の曲のテンポの揺らぎにより演奏能力を向上させた。開発した装置は近畿ポリテックビジョン 2016 作品展示部門において、来場者投票により優秀展示作品賞を受賞した。

I はじめに

ハンドベル自動演奏装置は平成 26 年度の総合制作課題⁽¹⁾であり、midi 形式の楽譜データを解析し、合計 27 個のハンドベルで演奏する装置である。演奏時には midi データに含まれるテンポ情報をそのまま利用していたために演奏のテンポは毎回固定されたままであり、微妙なテンポの変化による楽想を表現することができなかった。そこで、ハンドベル自動演奏装置にテンポ等を指示するマン=マシンインタフェースを装備し、操作者と対話的に演奏を変化させる機能を追加し、より表現力ある演奏装置の開発を目指した。

指揮者のように指先の合図により、演奏の開始・終了及びテンポの揺らぎ等の音楽のもつニュアンスを再生できる装置の開発を目指した。以後、システムの操作者をここでは指揮者と呼ぶことにする。動作指示やテンポを調整するためのインタフェースとして革新的な 3次元非接触型ポインティングデバイスである Leap Motion コントローラ⁽²⁾を採用した。このインタフェースにより、指揮者は楽曲のテンポを自由にコントロールすることができる。図 1 に指先で演奏中のシステム全体の外観を

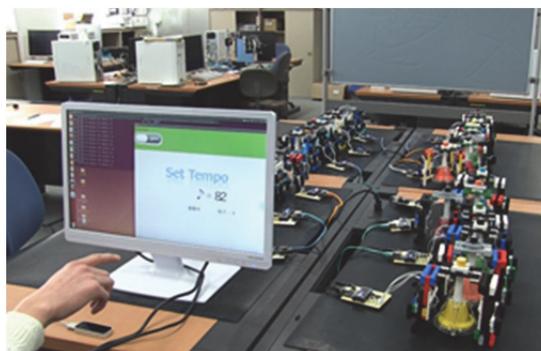


図1 指先で操作中のハンドベル自動演奏装置

を示す。

II システム概要

1 システム制御フロー

図 2 にシステムの制御フローを示す。指揮者の指の動きを Leap Motion コントローラは指の位置座標としてクライアントに最大 100FPS(Frame Per Second)で送信する。クライアントは指揮者の指のタップ間隔をテンポ情報に変換し、midi データに基づいたタイミングでハンドベルを発音させる機能をもつソレノイド制御サーバにイベントを送信する。ここで「タップ」とは拍の区切

りを検出するために画面上に作られた仮想的な壁を叩くことを言う。ソレノイドはイベント発生に従い遅延なくハンドベルを叩き発音を完了する。

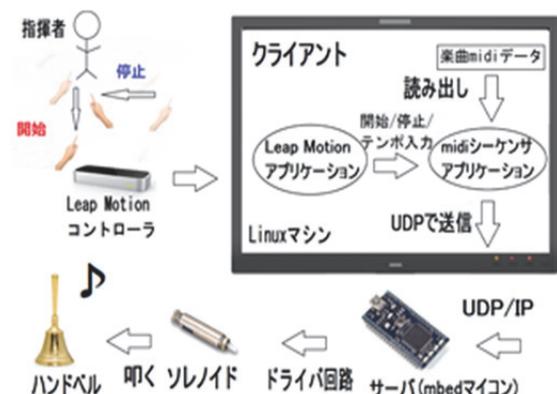


図2 システム制御フロー

2 非接触型 3次元ポインティングデバイス

Leap Motion コントローラは2012年にLeap Motion社から販売された手のジェスチャーによりコンピュータの操作を行うことのできるデバイスである。Leap Motion コントローラの実物は手のひらにのるほどの大きさとなる。2基の赤外線カメラと赤外線照射LEDから構成されており、赤外線LEDに照らされた手や指を2基の赤外線カメラで撮影し、画像解析によって対象物の位置の視差を利用して対象物までの距離を計算する。これにより指の位置を1/100 mm単位で検出し、3D空間での手や指の位置を割り出す。検知できる範囲は半径50cm程度、中心角110度の空間で、手、指、ペンのようなポイントを指し示すツールを認識する。

3 Interaction Box

Leap Motion コントローラはその中心から、垂直方向にx軸、水平方向にy軸、奥行き方向にz軸の検出範囲を持つが、この中にInteraction Boxと呼ばれる直方体の仮想検出環境を持つことができる。図3にInteraction Boxの座標系を示す。

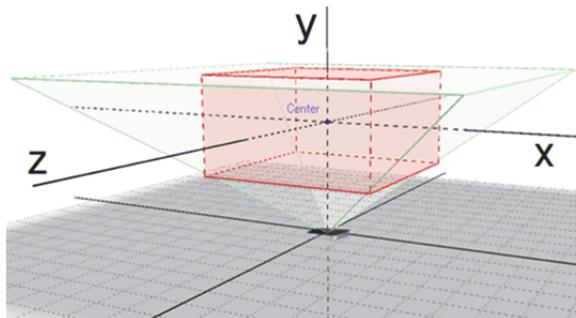


図3 Interaction Boxの座標系

III 仕様

本システムのマン=マシンインタフェースで実装する動作仕様を表1に示す。

表1 動作仕様

意図	ユーザの動作
曲開始・再開	指先で拍のテンポを入力する
曲一時停止	Leap Motion コントローラの上に手をパーにして持って行き、手のひらを下にしてグーにすることで曲を一時停止させることができる
曲目選択ボタンを押す/曲ディレクトリを移動する	人差し指を選択ボタンの上に持っていて、Leap Motion コントローラより奥に押し込む(以下、この動作をタッチと呼ぶ)

IV 開発環境・動作環境

Leap Motion コントローラ動作推奨環境及び開発の容易性を考慮し本システムでは以下の開発環境・動作環境を採用した。

CPU : i5-4460 3.2GHz 4Core

OS : Ubuntu 14.04LTS

SDK : Leap Motion SDK 2.3.1

開発言語 : C++

画像ライブラリ :

SDL(Simple DirectMedia Layer) 2.0⁽³⁾

V モジュール構成

図4に本システムのモジュール構成を示す。

1 入力モジュール

入力モジュールには Leap Motion を使用している。

2 音源制御モジュール

音源制御モジュールは UI(User Interface)プロセスと、midi シーケンサプロセスから構成されている。UI プロセスでは Leap Motion から指の位置情報を入力させるためのユーザインタフェースを提供し、指の位置情報から開始・停止・テンポ変更等の動作の判別を行う。midi シーケンサプロセスでは midi データの解析を行い、音源モジュールと通信を行って 27 個のハンドベルのうちどのハンドベルを鳴らすかという指示を出す。midi シーケンスプロセスから UI プロセスへの通信は UDP(User Datagram Protocol)を用いた同期 I/O プロセス間通信を採用した。一方、UI プロセスから midi シーケンサプロセスへの動作指示にはシグナルハンドラによる割り込み処理を伴う非同期 I/O プロセス間通信により実現している。

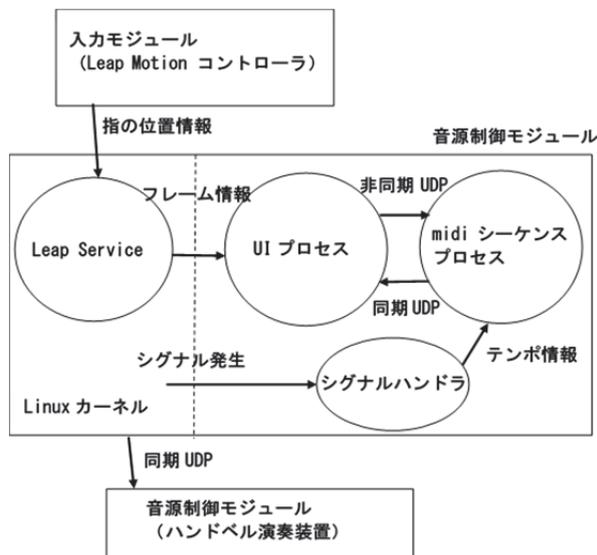


図4 モジュール構成図

3 音源モジュール

平成 26 年度の総合制作課題⁽¹⁾のサーバ機能をそのまま利用した。

VI 音源制御モジュール

1 タスク構成

1-1 UI プロセス

SDL(Simple Direct Media Layer)により、曲目選択画面と曲制御画面を提供する。Leap Service とやり取りをし、指の位置やテンポの取得、動作指示を行う。指のポインタは「Interaction Box」の検出環境内に右手人差し指が存在するとき、InteractionBox クラスの normalizePoint() メソッドで Interaction Box 内の座標を求め、その数値に応じて描画を行う。曲目選択画面上には、曲名のカードとそれらの表示場所を移動させるためのボタンが配置されている。ボタンや曲名のカードを押したいときは、ボタン、カード上に指のポインタをあわせ、押し込むことで選択できる。これは、Leap Motion コントローラより指が奥にあるとき、Pointable クラスの touchDistance() メソッドにより、タッチ状態であると判別されるためである。またテンポは、過去 10 回のテンポの単純移動平均を取ることで、指揮のテンポを滑らかに変化させることを実現している。

1-2 midi シーケンサプロセス

midi 形式の音楽データを先頭から順番に処理するプログラムである。UI 部から送られてきたテンポ情報と midi データを統合し、音源モジュールを制御する。

2 非同期 I/O プロセス間通信

UI プロセスから midi シーケンサプロセスへの通信はソケットを用いた非同期 I/O プロセス間通信によって可能となった。UI プロセスは画面表示のための描画ループを作る必要があり、一方 midi シーケンサプロセスでは音符の時間管理のために sleep 関数を使用している。同期 I/O プロセス間通信では、midi シーケンサプロセスが sleep しているとその間、UI プロセスからのデータ受信に遅延が発生し、リアルタイムに指揮情報の取得及びその結果を画面に表示することができないからである。プロセス間通信のシーケンス図を図5に示す。

CPU がマルチコアプロセッサでない場合は、マルチタスクにしても OS は 1 CPU に時分割で資源を割り当てるだけで、SDL の負荷に応じて midi プロセスの指定した sleep 時間は保障されなくなる。つまり描画のために曲のテンポが間延びする現象が発生し、初心者のように演奏時に楽譜を追いかけるだけのようなたどたどしい演奏になってしまう。

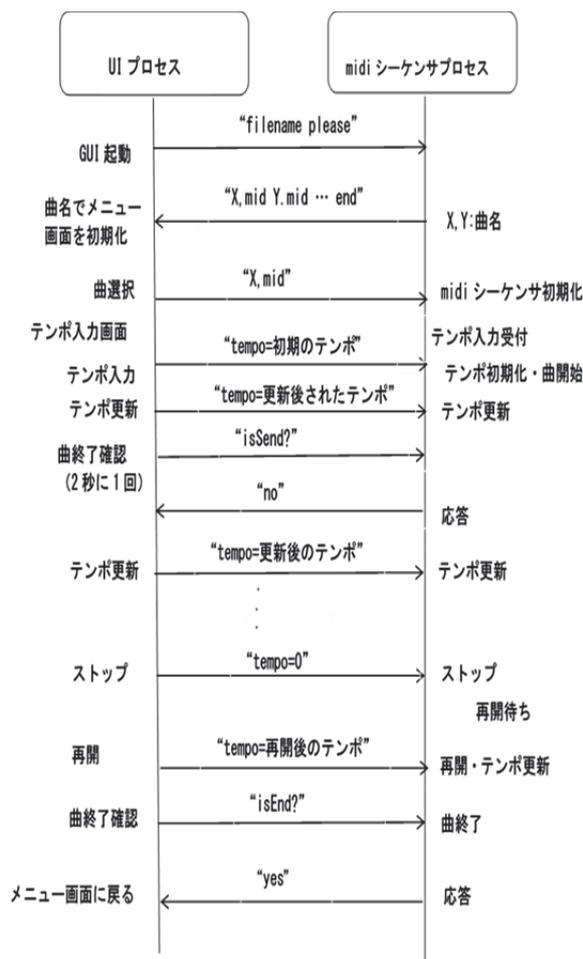


図 5 プロセス間通信シーケンス

よって本システムでは各プロセスに独立した CPU コアを割当てる必要がありマルチコアプロセッサは必須要件となる。sleep 関数と描画を両立させるために、midi シーケンサプロセスと UI プロセスを独立した 2 プロセスで構成し、UDP 通信することでプロセス間通信を行った。midi シーケンサプロセスは非同期 I/O プロセス間通信によって UI プロセスからのデータ受信があればシングルハンドラを実行する。受信がなければ midi データの解析、sleep を続行する。

VII 操作系

指揮者が操作を行う曲目選択画面の構成を図 6 に示す。

1 曲目選択

曲目選択画面で Leap Motion コントローラに右手を認識させると青色のポインタが表れる。ポインタは Leap

Motion コントローラの真上より奥に押し込むと赤色のポインタに変わり、この動作をタッチと呼ぶ。

水色のカードのアイコンは曲のカードを表し、このカードの上にポインタを持って行き、タッチすることで曲が選択でき、曲制御画面に移行する。黄色いアイコンは曲のディレクトリで、このアイコンにタッチするとディレクトリを階層的にする。次の曲目、前の曲目選択ボタンをタッチすると、次の 5 曲、前の 5 曲が表示される。



図 6 曲目選択画面

2 演奏中の状態遷移

曲目選択から始まるシステム状態遷移図を図 7 に示す。状態遷移のイベントを①～⑦に示す。

- ①曲名カード：選択するとその曲の演奏制御画面へ
- ②指のポインタ：指の位置を示すポインタ
- ③前の曲目選択ボタン：前の 5 曲表示
- ④次の曲目選択ボタン：次の 5 曲表示
- ⑤曲名：現在演奏中の曲名
- ⑥表示切替えボタン：二拍子と最適拍子を切り替え
- ⑦menu ボタン：演奏を中断し menu に戻る

3 曲制御

図 8 に曲制御画面を示す。曲制御画面では曲の開始、一時停止、テンポ変更ができる。まず青いポインタを出現させる。menu に戻るボタンをタッチすると曲目選択画面に戻る。

拍子切り替えボタンを押すと、四拍子、三拍子入力画面から二拍子入力画面に切り替わる。velocity0 の on/off ボタンは、タッチすると ON/OFF を切り替える。このボタンは midi データに含まれる velocity = 0 のノートオンイベントを鳴らすかどうかの切り替えである。鳴らすなら ON、鳴らさないなら OFF になっている。

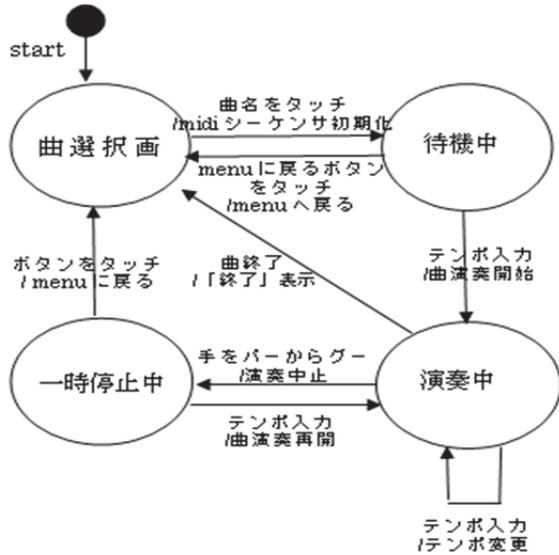


図7 システム状態遷移図

黄色い星は待機中と一時停止中、演奏中で指を認識していない時に表示され、待機中と一時停止中は曲の元のテンポを表す速さで動き、演奏中は入力されたテンポを表す速さで動く。テンポの入力方法は Tap 領域に指を入れることで Tap 領域が移動する。追いかける速さによって曲のテンポが入力される。テンポが入力されると曲が開始される。



図8 曲制御画面

① 四拍子の場合 四拍子入力画面の遷移を図9に示す。図9において(1)の状態が何も入力されていない状態で、指で Tap 領域を追いかけると(2)→(3)→(4)→(5)→(2)→(3)→(4)→(5)…と状態遷移していく。状態遷移に依存してタップ位置が移動する。この動作を繰り返すことで曲のテンポが入力できる。また、途中でテンポ入力をやめると最後に入力されたテンポで演奏が続く。テン

ポ入力をやめるには、Leap Motion コントローラの認識範囲から指をはずせばよい。

② 二拍子の場合 二拍子入力画面の遷移を図10に示す。二拍子入力モードでは画面最下部に固定されたタップ領域をタップすることで状態が遷移する。最初は(1)から始まり(2)→(3)→(2)→(3)→(2)→(3)…と状態遷移する。二拍子入力画面でも、途中でテンポ入力をやめた場合、最後に入力されたテンポを保持して曲が演奏される。

4 曲のテンポ計算方法

曲のテンポは過去10回のタップ時間差の単純移動平均を取っている。移動平均を求めるのは、bpm(beat per minutes)の敏感な変化により、テンポ変化がダイレクトに伝わってしまい曲が不安定に聞こえるのを防ぐためである。試行結果により、回数10が一番安定して曲のテンポ変化を自然に伝えているという主観に基づいて決定した。図11にテンポ取得のための時間測定ポイント及び計算式を示す。

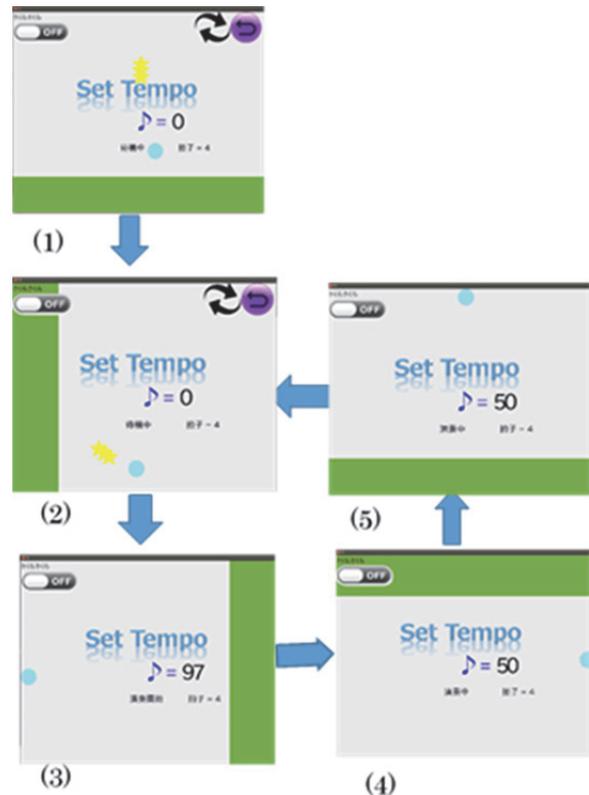


図9 四拍子入力画面の遷移

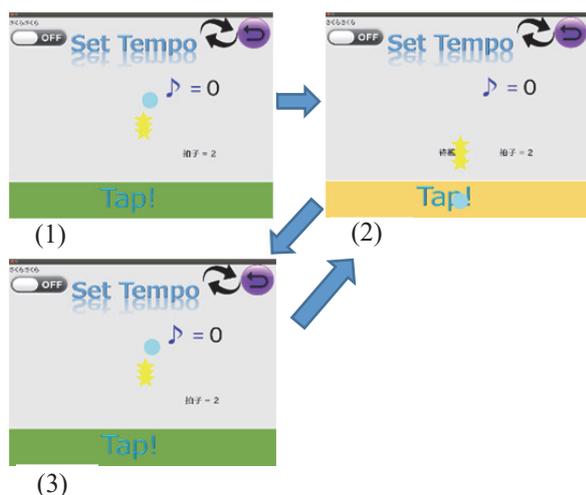


図 10 二拍子入力画面の遷移

の学習項目について開発を通して深く理解できる課題となった。楽曲演奏の2大要素であるテンポとダイナミクスのうちテンポの変化機能は完成した。今後は、もう一方のダイナミクスは演奏中の音の強弱をリアルタイムに変化させ実現していく予定である。

本テーマを熱心に取り組み担当してくれた現生産電子情報システム技術科1年生の塚本隆三君と尾崎真喜一君に感謝します。また本課題の成果として近畿ポリテクビジョン2016作品展示部門において、来場者投票による優秀展示作品賞の受賞として認められた。

[参考文献]

- (1) 近畿能開大ジャーナル, 第23巻, “ハンドベル自動演奏装置の製作”, pp.51-54, 2015. (近畿能開大ジャーナル)
- (2) Leap Motion1 開発者向けホームページ、
<https://developer.leapmotion.com/>
- (3) Simple DirectMedia Layer 開発元ホームページ、
<https://www.libsdl.org>

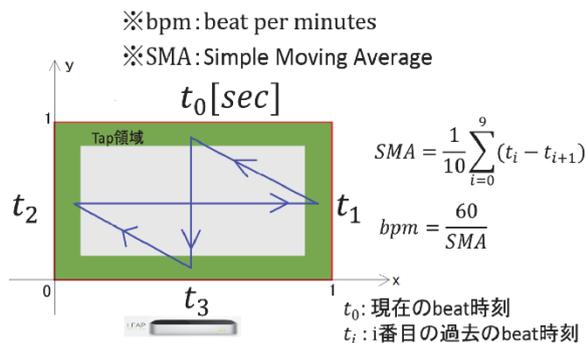


図 11 テンポの取得

VIII 動作検証

UIプロセスをフル稼働させ、1000 ms ごとに音を送り、パケットの送信間隔を測定した結果、送信間隔は平均1000.148 msであり、人間が遅れを感知できる限界であるとされる20 msと比較して、無視できる誤差であることが確認できた。また、曲の拍と指先のタップのタイミングが完全に一致していることを確認した。

IX おわりに

マン=マシンインタフェースを装備することによって、人と対話的に演奏をリアルタイムに制御できるシステムを総合制作課題として立案した。マルチタスク、非同期 I/O に関する知識、C++によるライブラリ操作等