筑波大学大学院博士課程

システム情報工学研究科修士論文

既存物体に対する能動的音響計測を用いたプロトタイピング手法

大野 誠

修士（工学）

(コンピュータサイエンス専攻)

指導教員　志築 文太郎

2015 年 3 月

概要

ものづくりにおいて、早期の段階でユーザのフィードバックを得るために、アイディアを最低限形にするプロトタイピングが行われることがある。プロトタイピングの形態は開発するシステムや製品によって異なる。本研究では電子工作の領域で扱われる、フィジカルなインタフェースのプロトタイピングに焦点を当てる。電子工作を伴うフィジカルなインタフェースのプロトタイプの作成は、電気回路等の専門知識を要するため、一般のユーザには困難である。本研究では、既存物体に対して能動的音響計測を行うことにより、フィジカルなインタフェースのプロトタイピングを容易に行う手法を示す。物体の音響特性は、その物体に対して触れる、把持する、圧力を加える、構造を変化させることによって変化が生じる。本研究では、そのような音響特性の変化をユーザがプロトタイプに対して行う操作によるものとしてとらえ、音響特性からプロトタイプの操作を認識する技術およびそれを応用したプロトタイピングツールキットを開発する。提案する手法（以下本手法とする）では能動的音響計測によって既存物体の音響特性を取得し、コンピュータ上で機械学習を適用することによってその音響特性から既存物体に対する操作を認識する。これにより、ユーザは自分の使いたい操作を既存物体に対して実演することによって、容易にフィジカルなインタフェースを構築することが可能となる。この際にユーザに求められるハードウェア構築の作業は、能動的音響計測のための 1 組の圧電素子を取り付けるだけである。本研究では、開発したツールキットを用いて実際にいくつかのプロトタイプを作成することにより、本手法の適用可能性を示した。また操作の認識精度を評価する実験を行い、その結果から本手法の有用性を示した。

目 次

第 1 章　序論 **1**

1.1 プロトタイピング 1
1.2 能動的音響計測 3
1.3 本研究の目的 3
1.4 本論文の構成 3

第 2 章　関連研究 **4**

2.1 プロトタイピング手法に関する研究 4
2.1.1 専門知識を持つユーザを対象とした研究 4
2.1.2 専門知識を持たないユーザを対象とした研究 5
2.2 既存物体をインタフェース化する研究 5
2.3 音響計測に基づいたセンシング手法に関する研究 6
2.3.1 受動的音響計測に基いたセンシング手法に関する研究 6
2.3.2 能動的音響計測に基いたセンシング手法に関する研究 6
2.4 本研究の位置づけ 7

第 3 章　能動的音響計測を用いたセンシング技術 **8**

3.1 センシング原理 8
3.2 プロトタイピング手法への応用方法 9

第 4 章　既存物体に対する能動的音響計測を用いたプロトタイピングツールキット **10**

4.1 ツールキットの構成 10
4.2 ハードウェア 10
4.2.1 インタフェースモジュール 10
専用インタフェースモジュール 10
オーディオインタフェース 12
4.2.2 圧電センサ 13
4.3 ソフトウェア 13
4.3.1 認識ソフトウェア 13
離散値入力型 14
連続値入力型 15
4.3.2 外部プラグイン 15

Scratch 用プラグイン 15
Max/MSP・PureData 用プラグイン 16
4.4 使用例 17
4.4.1 ハードウェアの取り付け 17
4.4.2 操作の学習 19
4.4.3 アプリケーションとの関連付け 19

第 5 章 実装 **21**
5.1 ハードウェア 21
専用インタフェースモジュール 21
5.1.1 圧電センサ 22
アクリル版 22
レゴブロック版 26
5.2 ソフトウェア 26
5.2.1 認識ソフトウェア 26
音響特性データの取得（専用インタフェースモジュールを使用する場合） 27
音響特性データの取得（オーディオインタフェースを使用する場合） 27
離散値入力の学習 28
連続値入力の学習 28
5.2.2 Scratch 用プラグイン 29
JSON ファイル 29
中間ソフトウェア 30
5.2.3 Max/MSP・PureData 用プラグイン 30

第 6 章 適用例 **32**
6.1 タッチ入力インタフェース 32
6.1.1 タッチ認識技術に関する研究 32
6.1.2 適用例 32
タッチに反応する玩具 32
楽器 33
ゲームコントローラ 33
6.2 把持認識インタフェース 35
6.2.1 把持認識技術に関する研究 35
6.2.2 適用例 35
携帯電話のケース 35
複数機能マウス 36
6.3 圧力入力インタフェース 39
6.3.1 圧力認識技術に関する研究 39
6.3.2 適用例 39

手のオブジェ . . . 39
せん断力入力 . . . 39
6.4 ブロック玩具を用いたインタフェース . . . 41
6.4.1 ブロック玩具を用いた研究 . . . 41
6.4.2 適用例 . . . 41
連続入力系 . . . 41
6.4.3 連続値入力 . . . 42
6.4.4 離散値入力 . . . 43

第 7 章 評価実験 **45**
7.1 離散値入力学習に関する実験 . . . 45
7.1.1 被験者 . . . 45
7.1.2 実験手順 . . . 45
7.1.3 実験結果 . . . 46
Per-User Classifiers . . . 46
General User Classifiers . . . 46
7.2 連続値入力学習に関する実験 . . . 48
7.2.1 被験者 . . . 48
7.2.2 圧力センサを用いた学習システム . . . 48
7.2.3 実験手順 . . . 48
7.2.4 認識対象 . . . 49
7.2.5 実験結果 . . . 49

第 8 章 議論 **52**
8.1 制約 . . . 52
8.1.1 センサの取付位置 . . . 52
8.1.2 物体の構造 . . . 52
8.1.3 学習の習熟 . . . 52
8.2 今後の展望 . . . 53
8.2.1 センサ取付方法の検討 . . . 53
8.2.2 ツールキットとしての使いやすさの評価 . . . 53
8.2.3 3D プリンタで出力された物体への応用 . . . 53

第 9 章 結論 **54**

謝辞 **55**

参考文献 **56**

付 録 A 評価実験に使用した書類 **64**

図目次

1.1 携帯情報端末用ソフトウェアのペーパープロトタイピング。 2
1.2 圧力検出可能なタッチパッドのプロトタイピング。 2

3.1 セラミックのお椀の音響特性（周波数スペクトル）。 8

4.1 本ツールキットの構成。 11
4.2 専用インタフェースモジュール（USB 接続型）。 12
4.3 専用インタフェースモジュール（Bluetooth 接続型、1 回の充電につき約 1.5 時間駆動する）。 12
4.4 本研究で使用したオーディオインタフェース。 13
4.5 圧電センサ（アクリル版）。 14
4.6 圧電センサ（レゴブロック版）。 14
4.7 離散値入力型認識ソフトウェア。 15
4.8 連続値入力型認識ソフトウェア。 16
4.9 Scratch 用プラグインの使用例。 17
4.10 Max プラグインの使用例。 18
4.11 本ツールキットを用いたプロトタイピングの流れ。a) ハードウェアの取り付け、b) 操作の学習、c) アプリケーションとの関連付け。 18
4.12 適用可能な物体の制約（材質）。 18
4.13 適用可能な物体の制約（大きさ）。 19
4.14 音楽プレイヤの操作のための 3 種類のタッチジェスチャ。 19
4.15 音楽プレイヤーの Max パッチ。 20

5.1 専用インタフェースモジュールのブロック図。 22
5.2 専用インタフェースモジュールの回路図（USB 接続版）。 23
5.3 専用インタフェースモジュールの回路図（Bluetooth 接続版）。 24
5.4 専用インタフェースモジュールの基板（USB 接続版）。 25
5.5 専用インタフェースモジュールの基板（Bluetooth 接続版）。 25
5.6 一般的な包絡線検波回路。 26
5.7 3 次 Shellkey ローパスフィルタ。 26
5.8 レゴブロック版圧電センサの作成手順。 27
5.9 Scratch プラグインの JSON コード。 29

6.1 タッチに反応する玩具。 33
6.2 楽器のプロトタイプ。 34
6.3 使用したキャラクタの玩具。 34
6.4 お椀を用いたゲームコントローラ。 34
6.5 本手法適用した携帯情報端末ケースの作成。 36
6.6 携帯情報端末の 6 種類の把持姿勢。 37
6.7 本手法を適用した複数機能マウスのプロトタイピング。 37
6.8 複数機能マウスの把持方法と対応するツール。 No Touch) マウスを使用していない状態。Move) カーソルを移動させるツール。マウスのサイドを持つことによって機能する。 Pick) ドラッグを行うツール。マウスの持ち手に力を加えることによって機能する。なにもないところをドラッグすると選択範囲が表示されそれを再びドラッグすることによって選択範囲を移動させることができる。Paint) ペイントを行うツール。線を書くことができる。マウスの中央の溝に指を引っ掛けることによって機能する。 Erase) 線を消すためのツール。マウス全体を把持することによって起動する。Loupe) ズームを行うツール。マウスカーソルの周囲のキャンバスを拡大することができる。マウスの凸型部分の先端把持することによって起動する。 38
6.9 手のオブジェ。 40
6.10 せん断力入力のためのケース。 40
6.11 せん断力入力の様子。 40
6.12 連続入力系プロトタイプ：a) レバー、b) ペダル、c) スライダー。 42
6.13 離散入力系プロトタイプ：a) スイッチ、b) ゲームコントローラ、c) インタラクティブアニマル。 43
6.14 動物型ブロックとタッチの認識。 44
6.15 動物型ブロックの識別。 44

7.1 Per-user classifier における認識率。 47
7.2 General classifier における認識率。 47
7.3 圧力センサを用いた学習システムの構成。 49
7.4 認識対象：a) 木の机、b) プラスチックの玩具（頭）、c) プラスチックの玩具（右手）、d) アクリルのオブジェ。赤丸は圧力センサの取り付け位置および実験時にタッチした位置を示す。 50
7.5 評価結果：a) 木の机、b) プラスチックの玩具（頭）、c) プラスチックの玩具（右手）、d) アクリルのオブジェ。 51

第1章　序論

本研究では、能動的音響計測をプロトタイピングに応用することにより、既存物体を用いて容易にフィジカルなインタフェースを構築する手法を示す。本章では、まず初めに背景として、既存のプロトタイピング手法とその課題について述べ、その課題に対して能動的音響計測がどのような点において利用可能であるかを述べる。次に、本研究の目的について述べ、最後に本論文の構成を述べる。

1.1　プロトタイピング

何らかのシステムや製品を開発する際に、早期の段階でユーザのフィードバックを得るために、アイディアを最低限形にする試作が行われることがある。これはプロトタイピングと呼ばれ、作成するシステムや製品に応じて様々な形態をとる。例えば、ソフトウェア開発においては、開発するソフトウェアのユーザインタフェースの使用感を確かめるために、ペーパープロトタイピングが行われる。ペーパープロトタイピングにおいては、紙やペン等を使用してユーザインタフェースのスケッチを作成し、システムの操作に応じてスケッチを切り替えることによってそのシステムの使用感を確認する。例として、図に携帯情報端末用のソフトウェアに関するペーパープロトタイピングの様子を示す。この図では、ソフトウェアのユーザインタフェースのスケッチと携帯情報端末の画面枠を出力した紙を用意し、スケッチ上のボタンの配置や画面遷移に関する使用感を調査している。これには特別な専門技術を必要としないため、誰でも容易にプロトタイピングを行うことができる。

一方で、ハードウェア開発においてはフィジカルなインタフェースのプロトタイピングが必要となる場合がある。フィジカルなインタフェースのプロトタイピングは、一般的にマイクロコントローラ、センサ、アクチュエータ、ブレッドボード等を使用した電子工作として行われる。図にプロトタイピングの例として、圧力検出可能なタッチパッドを試作している様子を示す。この例では大量の光学式センサを使用し、複数のマイクロコントローラを用いて複雑な制御を行っている。これには電気回路やプログラミング等の専門知識が求められるため、一般のユーザにとっては複雑な作業となる。本研究では、このようなフィジカルなインタフェースのプロトタイピングに焦点を当てる。

フィジカルなインタフェースのプロトタイピングを容易にするためのツールキットも開発されている。Makey Makey [SRS12] は、そのような目的で作られたプロトタイピングツールキットであり、本体に備わる端子と既存の導電性の物体をワニ口クリップで接続するだけで、その物体をタッチ入力インタフェースとして扱うことができる。例えば、8 つのバナナを Makey

Makey の本体に接続するだけで、ドレミファソラシドの音階を演奏できる楽器のプロトタイプを作成可能である。これは既存物体の電気的な状態を静電容量計測を用いてセンシングすることによって実現している。これによって、ユーザは電気回路を意識することなく、既存物体を用いてフィジカルなインタフェースのプロトタイピングが可能となる。しかしながら、ここで使用可能な既存物体は導電性の物体に限定されており、検出可能な操作はタッチ操作に限定されるという制約がある。

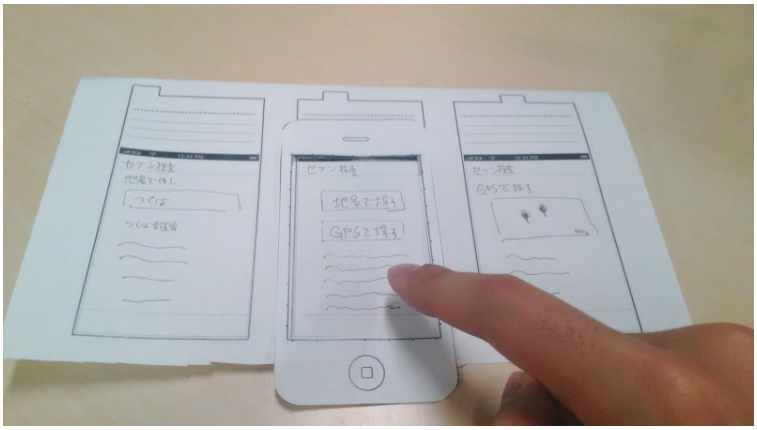

図 1.1: 携帯情報端末用ソフトウェアのペーパープロトタイピング。

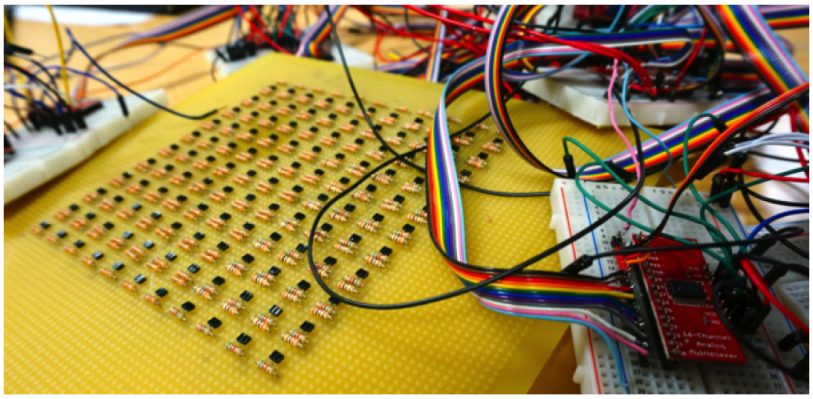

図 1.2: 圧力検出可能なタッチパッドのプロトタイピング。

1.2 能動的音響計測

物体の状態をセンシングする手法の一つとして能動的音響計測と呼ばれる手法がある。これは物体の音響特性を計測する手法であり、インパルスハンマー、加振器、振動子等を用いて物体を振動させその応答を加速度センサやコンタクトマイク等を通して取得することによって行われる。能動的音響計測は従来テニスラケットや梁等の構造物において、理想とする振動状態を実現するために用いられる。これは物体の音響特性がその物体の構造や形状によって変化することを利用している。例えばテニスラケットの場合、打球時に手に振動が伝わりにくいラケットを実現するために、能動的音響計測の結果に基いてその形状の微調整が行われる。これは実際の計測以外にシミュレーションも組み合わせて行われる。一方で、物体の音響特性は物体の形状や構造だけでなく、その境界条件によって変化することが分かっている [SR99]。例えば、テニスラケットの振動においても、ラケットを紐で吊り下げた自由支持状態と、手でグリップを握った支持状態では音響特性が異なる。すなわち、ユーザが物体に対して、触れる、把持する、力を込める等の操作を行うとき、音響特性に変化が現れることを意味する。これを利用して物体の音響特性とユーザの操作のパターンの対応さえとれれば、物体の音響特性からユーザの操作を認識することが可能である。

本研究ではこのことに着目し、それをプロトタイピングへと応用する。音響特性の計測は振動を伝達する物体であれば、非導電性、導電性問わず、様々な物体に適用可能である。したがって、本手法は Makey Makey と同様に既存物体を用いてフィジカルなインタフェースを構築する手法でありながら、より広い適用範囲をもつ手法として期待される。

1.3 本研究の目的

本研究の目的は能動的音響計測をプロトタイピングに応用することによって、専門知識を持たない一般のユーザが既存物体を用いて容易にフィジカルなインタフェースのプロトタイピングを行えるようにすることである。また、本手法が導電性、非導電性を問わず多くの既存物体対して適用可能であり、かつ様々な種類の操作を認識できることを示すことも目的とする。

1.4 本論文の構成

第 1 章以降の本論文の構成は次のとおりである。第 2 章では、関連研究を紹介することにより、本研究の位置づけを述べる。第 3 章では、本研究の提案手法中で使用する能動的音響計測の詳細とプロトタイピングへの応用を述べる。第 4 章では、提案手法に基づいて実装したプロトタイピングツールキットを、第 5 章ではその実装を述べる。第 6 章ではそのツールキットを用いていくつかのプロトタイプを作成することにより、提案手法の適用例を述べる。第 7 章では、提案手法の操作認識精度に関する実験について述べる。第 8 章では、本研究における課題および今後の展望を述べる。最後に第 9 章にて結論を述べる。

第2章　関連研究

本章では本研究に関連する研究について述べる。本研究では、能動的音響計測に基づいたセンシング手法を用いて既存物体をインタフェース化することにより、プロトタイピングを行う手法を提案している。よって、本研究の関連研究はプロトタイピング手法に関する研究、既存物体をインタフェース化する研究、音響計測に基づいたセンシング手法に関する研究に分類される。本章では、それらの研究について述べた後、最後に本手法の位置づけについて述べる。

2.1　プロトタイピング手法に関する研究

これまでに多くの研究者によって、様々なプロトタイピング手法の研究がなされてきた。これらの研究は電気回路やプログラミング等の専門知識を持つユーザを対象としたより広い応用範囲をもつ研究と、限定された応用範囲でありながら、専門知識を持たないユーザを対象としている研究に分類される。

2.1.1　専門知識を持つユーザを対象とした研究

Greenberg らの Phidgets [GF01] はパッケージングされたセンサやアクチュエータ群とそれをプログラムするためのソフトウェア API によって構成されるプロトタイピングツールキットである。これにより、ユーザはフィジカルなインタフェースの構築する際に、その実装よりもアイディアの実現に集中することができる。Arduino [ard] や Mbed [mbe] は多数のアナログやデジタル IO を持つマイコンによるプロトタイピングプラットフォームである。ユーザはそれらのマイコンとオンラインやオフラインの IDE を用いてハードウェアのプロトタイピングを行うことができる。Hartman らの d.tools [HKB+06] はプロトタイプの設計、テスト、解析機能をビジュアルプログラミング環境に統合したプロトタイピング環境である。ユーザはプロトタイプの動作をソフトウェア上の状態遷移図に設計可能である、これにより容易に試行錯誤を行うことができる。Villar らの.NET Gadgeteer [VSH+12] は Microsoft の.Net Micro Framework によって動作する d.tools と同様のパッケージモジュールである。Kawahara らの AgiC [KHC+13] は銀ナノ粒子を含ませたインクを用いることによって様々な回路を家庭用のプリンタで出力することを可能にした。上述したツールによるプロトタイピングでは様々なセンサを用いたハードウェアの設計およびプログラムの記述が必要となるため、電気回路やプログラミング等の専門知識を求められる。

2.1.2 専門知識を持たないユーザを対象とした研究

一方で、専門知識を持たない多くのユーザが使えることを目的として開発されたプロトタイピングツールもいくつか存在する。序論で述べた Silver らの Makey Makey [SRS12] もこの分類に該当する。Ayah の littleBits [Bde09] は磁石によって容易に接続可能な電子部品モジュール群によるプロトタイピングツールであり、センサやアクチュエータだけでなく論理モジュール等も備わっているため、それらのモジュールを単に接続していくだけでソフトウェア上のプログラミングなしにプロトタイプを構築可能である。Sony の Mesh [mes] は Bluetooth によって接続された小型のモジュール群（センサ、LED、モータ等）を用いたオープンプラットフォームのプロトタイピングツールである。センサ状態の取得やアクチュエータの操作は Bluetooth 通信を介してコンピュータ上で可能となっている。ユーザはモジュール群を物体の好きな位置に取り付け、挙動をヴィジュアルプログラミング環境を用いてプログラムすることによって、インタラクティブなプロトタイプを作成可能となる。Mitchel らの Scratch [RMMH$^+$09] はブロックを組むような感覚でプログラミングを可能とするヴィジュアルプログラミング言語であり、フィジカルなインタフェースを構築するための拡張ハードウェア（PicoBoard）も存在する。Hartmann らの Exemplar [HAMK07] はセンサベースのインタラクションを設計する際に、センサによる入力を実演し、そのデータを編集することによって反復的なテストを可能としている。Fourney らの PICL [FT12] も同様に実演によって、アクションを起こすセンサ入力のしきい値を設定しているが、こちらはハードウェア上の機械学習を用いている。Hudson らの BOXES [HM06] は画鋲とボール紙を用いてフィジカルなボタンを持つデバイスのプロトタイピングを可能にしている。Valkyrie らの Midas [SZH12] はビニールカッターによって生成された導電パターンを用いてタッチ入力を扱うデバイスのプロトタイピングを可能にしている。

2.2 既存物体をインタフェース化する研究

フィジカルなインタフェースをデバイスとして新たに作成するのではなく、既存物体を活かしながらインタフェースとして拡張しようという研究がなされている。Sato らの Touchè [SPH12] および Ivan らの BOTANICUS INTERACTICUS [PSLS12] は静電容量式のタッチ認識技術を電流の周波数を掃引することによって拡張し、1 つの静電容量センサで多様なタッチを検出することを可能にした。これによりドアノブのような導電性の日常品において、手で触れる、握る等のジェスチャを認識可能なインタフェースを構築することが可能となった。Sugiura らはフォトリフレクタとサーボモータによって、ぬいぐるみ [SLO$^+$12] やクッション [SKW$^+$11, LSFI13] のような既存の柔軟物を入出力インタフェースとして拡張するデバイスを開発した。Harrison らの Scratch Input は壁や机等を引っ掻いた時に生じる筆記音をコンタクトマイクで検出することにより、それらを入力インタフェースとして用いる応用を示している [HH08]。また、宮田らはドアや引き出しのような動きが生じる家具に加速度センサモジュールを取り付けることによってそれらの操作を検出し、インタフェース化している [宮田 13]。前節で述べた Makey Makey [SRS12] および Sony の Mesh [mes] もこの分類にも該当する。

2.3 音響計測に基づいたセンシング手法に関する研究

音響計測に基づいたセンシング手法に関する研究も多くなされている。これらの手法は受動的な手法と能動的な手法に分類される。

2.3.1 受動的音響計測に基いたセンシング手法に関する研究

受動的な手法はユーザの操作によって生じる音を録音し、解析することに基づく。The Sound of One Hand [AHT02] は骨伝導音を解析することによって、叩く、擦る、弾く等のユーザの指先のジェスチャを認識している。Stane [MSWHQ08] は、そのザラザラした面を引っ掻いたり、擦ることによって生じる音を識別し、入力インタフェースとして応用している。2.2 節で述べた Scrach Input [HH08] はこれを壁や机等の既存物体上における入力に応用したものである。Acoustic Barcodes [HXH12] はザラザラしたテクスチャを持つバーコードであり、ユーザがそれを爪や携帯情報端末で引っ掻くことによって、そのバーコードのテクスチャに応じた音が生じる。コンピュータ上でこの音を解析することにより、ID を認識する。Paradiso らは複数のコンタクトマイクを用いて音の到達時間差を計算し、ガラスシート上においてタップされた位置を推定している [PLCH02]。Pham らも単一のコンタクトマイクと Time Reversal を用いて、立体物においてタップされた位置を推定している [PJY+07]。Skinput [HTM10] は生体音響センサを腕に取り付けることにより、人間の肌上におけるタップされた位置を推定している。TapSense [HSH11] および Pedro らの研究 [LJJ11] では携帯情報端末やテーブルトップ上のタップ音の周波数特性を解析することにより、どのような物体によってタップされたか、もしくは指のどの部位でタップされたか等の推定を行っている。MicPen [HBW12] はゴム状のペン先を持つペンの筆記音の強さを測定することにより、筆圧検出を行うスタイラスである。

2.3.2 能動的音響計測に基いたセンシング手法に関する研究

能動的な手法はスピーカを用いて音を出力し、その応答をマイクで取得し解析を行うことに基づく。潜水艦や漁船に搭載されるアクティブ・ソナー [SW59] は典型的な例である。アクティブ・ソナーはパルス音を出力し、その応答が帰ってくる時間を測定することにより、周囲の障害物とその距離を検出している。表面弾性波タッチスクリーン [BF87] はもう 1 つの代表例である。これは 1 枚のガラスシート、トランスミッタ、レシーバ、反射材によって構成される。トランスミッタが弾性体の表面を移動する表面弾性波を発生させると、それらはスクリーンの端に取り付けられた反射材によって受信機の方に反射され、検出される。この時、タッチスクリーンを指で触れると弾性波が吸収されるので、その情報に基いてタッチ位置を推定する。近年の研究においてその他の能動的な手法に関する研究もいくつかなされている。SoundWave [GMPT12] はドップラー効果を応用して、空中のハンドジェスチャを検出している。Takemura らは、骨伝導音の伝達特性の変化を 1 つのスピーカと 2 つのマイクで計測することにより、肘の角度を推定できるウェアラブルセンサシステムを開発した [TITO11]。Collins は 1 つのトランスミッタと 2 つのレシーバを用いてガラス平面の周波数応答を解析す

ることにより、そのガラス平面上におけるタッチ位置を推定した [Col09]。この研究は音響波からタッチ位置を推定する点で表面弾性波タッチスクリーンとに似ているが、周波数特性からタッチ位置を推定するため、単純なシステム構成によって実現可能である。

能動的手法はシステム自身が音を発生させユーザの操作によって生じる応答の変化を検出しているため、タップのようなユーザが意図的に音を発生させるような操作以外にも、軽く触れたり、強く押し込んだり等の操作を認識できる点が利点としてあげられる。

2.4 本研究の位置づけ

本研究では 2.1.2 節で示した研究のように、電気回路やプログラミング等の専門知識なしでフィジカルなインタフェースを構築するプロトタイピング手法を提案する。本手法によって作成するプロトタイプにおける入力の検出は、MESH や littleBits のように複数種類のセンサ群を用いるのではなく、1 組の圧電素子を用いて行う。また、既存物体をインタフェース化するという点で Makey Makey 等の研究と類似しているが、本手法は能動的音響計測に基づくものである。これにより、導電性、非導電性問わず様々な物体に適用可能であり、様々な操作を認識可能としている。音響計測技術に関する研究の中では Colins の研究が最も近い。しかしながら、本研究は能動的音響計測を、プロトタイピングに応用するという点で異なる。

第3章　能動的音響計測を用いたセンシング技術

本章では本手法で使用する能動的音響計測を用いたセンシング技術について、その原理とプロトタイピングへの応用方法を述べる。

3.1　センシング原理

物体はその構造の特徴を表す指標のひとつとして音響特性を持つ。この特性は振動モード、固有振動数、モード減衰比によって表現される。振動モードは振動する際の物体の形状、固有振動数は最もよく振動する周波数、モード減衰比はその周波数から遠ざかった際にどの程度振幅が減衰するかを示している。これらの性質は周波数スペクトルとして表すことができる。例として図 A にセラミックのお椀の音響特性による周波数スペクトルを示す。

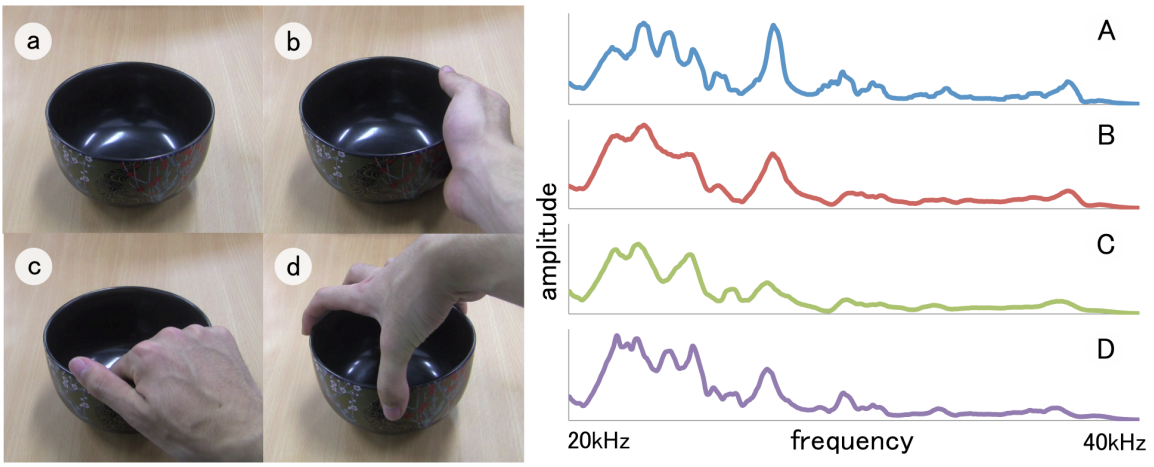

図 3.1: セラミックのお椀の音響特性（周波数スペクトル）。

物体の音響特性はその物体の形状、材質、境界条件に依存する。本研究ではこれらのうち、境界条件に着目する。境界条件は物体に対する荷重や拘束の条件によって定義される。物体に触れたり力をかけたりする場合、それらが境界条件の変化として働くため、音響特性に変化が生じる。この時の触れ方（触れる位置や強さ）によってそれぞれ物体に対する接触点や圧力が異なるため、それらは音響特性の変化として表れる。よって、物体の形状や材質に変化がない場合、物体の触れ方の変化は、その圧力の変化も含めて音響特性の変化として観測

可能である。図B、C、Dはそれぞれお椀に対してb、c、dのような触り方をした場合の周波数スペクトルを示す。各触れ方によって異なる周波数スペクトルを示していることが分かる。

同様にして、物体の形状が変化する場合においても音響特性は変化する。これは物体の形状が変化することによって振動モードが変化するためである。例えば、レゴブロックのようなブロック玩具を用いて形状を構成する場合、その形状によって得られる音響特性は異なる。

本研究ではこれらの現象を利用して、物体の音響特性を解析し、その解析結果から物体の触れ方やその強さ、形状の変化等を推測する。音響特性の解析には、実現象の観測に基づく周波数応答解析を用いる。これはアクチュエータによって物体を様々な周波数で振動させ、センサによって振動の周波数応答を取得する、実現象の観測に基づく手法である。

3.2 プロトタイピング手法への応用方法

前節の原理によって、物体の音響特性を能動的に観測することにより物体の様々な状態が分かることを示したが、これをプロトタイピングに応用するには、得られた音響特性の情報から、ユーザによって物体に対してどのような操作が行われたかという情報を引き出す必要がある。有限要素法等を用いて実際に物体にどのような荷重がかかっているかを解析する方法も存在するが、本手法では機械学習を用いる。機械学習ではユーザが作成するプロトタイプに対して使用する操作をコンピュータ上で定義し、その操作をプロトタイプ上で実演する。ここでユーザによる操作の定義を教師ラベル、実演する際に得られる音響特性を特徴量データとして機械学習を行う。これによって、ユーザが再びその操作を行った際にコンピュータ上でそれを認識することができる。

第4章　既存物体に対する能動的音響計測を用いたプロトタイピングツールキット

本章では、第 3 章で述べた提案手法に基いて作成したプロトタイピングツールキットについて述べる。

4.1　ツールキットの構成

本ツールキットは既存物体から音響特性を取得するハードウェアと、操作の認識とそれに対する応答を定義するソフトウェアによって構成される。本ツールキットの構成を図に示す。ユーザはハードウェアを既存物体に取り付けることによってプロトタイプを構成し、認識ソフトウェアを用いてそのプロトタイプへの操作を学習させる。これによってプロトタイプへの操作をコンピュータに認識させることができる。さらに認識結果はプロセス間通信を通して Max/MSP や Scratch 等のビジュアルプログラミング環境に送信され、ユーザはそれらのソフトウェア上でプロトタイプへの操作に対する振る舞い（音を鳴らす、ビジュアルエフェクトを表示する等）を定義する。これらの手順によってユーザはフィジカルなインタフェースのプロトタイピングを行う。

4.2　ハードウェア

ハードウェアは既存物体から音響特性を取得し、コンピュータに送信するために用いる。ハードウェアはインタフェースモジュールと圧電センサによって構成される。

4.2.1　インタフェースモジュール

インタフェースモジュールは音響特性をコンピュータに取り込むための信号の入出力を行う。本ツールキットではインタフェースモジュールとして、専用インタフェースモジュールとオーディオインタフェースのどちらかを使用する。

専用インタフェースモジュール

本ツールキットによるプロトタイピングを行うために最適化した、専用インタフェースモジュールを作成した。専用インタフェースモジュールには USB 接続型（図）と Bluetooth 接

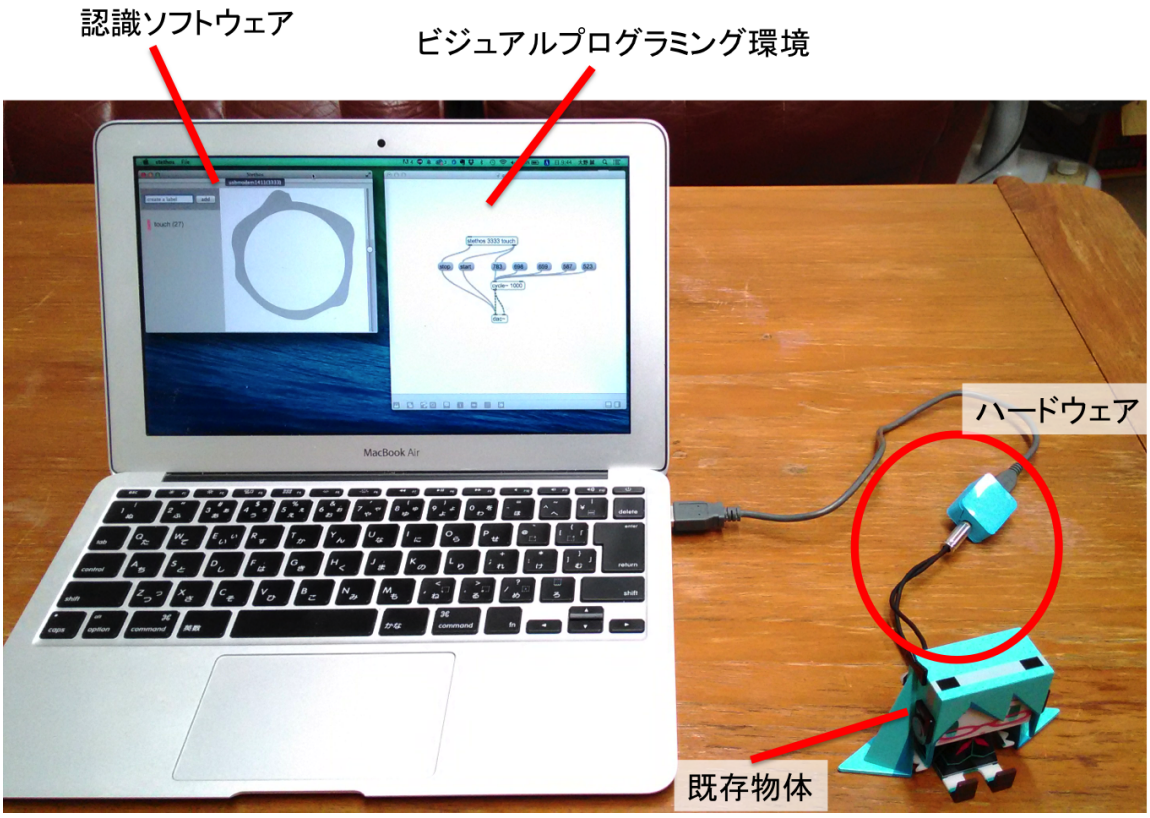

図 4.1: 本ツールキットの構成。

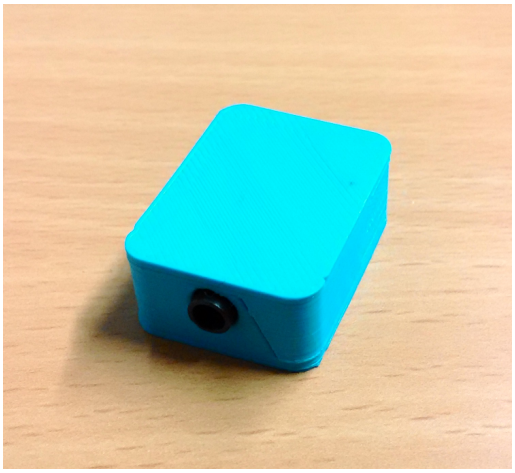

図 4.2: 専用インタフェースモジュール（USB 接続型）。

図 4.3: 専用インタフェースモジュール（Bluetooth 接続型、1 回の充電につき約 1.5 時間駆動する）。

続型（図）があり、プロトタイピングの用途に応じて使い分けることができる。初期段階のプロトタイピング等において、プロトタイプとコンピュータとの接続が有線でも支障ない場合には USB 接続型を用いる。一方でプロトタイプ作品の展示の際等、できるだけ配線を目立たなくさせたい場合には Bluetooth 接続型の使用が望ましい。

専用インタフェースモジュールは必要最低限の端子のみを備えているため、使用が簡単である。また音響特性の抽出をハードウェア内で行っているため、コンピュータの計算負荷が低く無線化も実現している。

オーディオインタフェース

本ツールキットでは、専用インタフェースモジュールの代替品としてオーディオインタフェースを用いることができる。オーディオインタフェースは、本来楽器の演奏の録音や高品質なオーディオの再生を用途としているため、不要な端子やツマミが多く、大きさも専用インタフェースモジュールに比べて大きい。また、オーディオインタフェースは信号の入出力のみを行うため、音響特性の抽出はコンピュータ内で行う必要があり、ある程度コンピュータの計算負荷を伴う。しかしながら、ノイズの少ない高品質な信号を得られるため、認識精度の評価等においてなるべく正確な認識を実現したい場合に有用となる。本研究では評価実験のために Native Instruments 社製のオーディオインタフェース、Audio Kontrol 1（図）を使用した。

4.2.2 圧電センサ

圧電センサは物体の音響特性を取得する際に、インタフェースモジュールからの信号を振動に、物体の振動応答を電気信号に変換するために使用する。

図 4.4: 本研究で使用したオーディオインタフェース。

圧電センサは 2 つの圧電素子を有しており、それらを物体に取り付けることによって音響特性の計測を行う。2 つの圧電センサのうち、片方は振動を出力するためのもの、もう片方は振動応答を取得するためのものである。圧電センサにはアクリル版（図）とレゴブロック版（図）の 2 種類が存在する。アクリル版は両面テープ（SPG-12）で貼り付けることによって様々な既存物体に取付可能である。一方レゴブロック版はレゴブロックに対しては両面テープを用いずに取り付け可能である。そのため、レゴブロックを用いてプロトタイピングを行う際にはレゴブロック版の圧電センサを使用する（6.4 節で詳述する）。圧電センサとインタフェースモジュールは 3.5mm プラグを介して接続される。

また、以上の 2 種類の圧電センサ以外にも、圧電素子を直接利用することにより使用用途に最適化させることも可能である。

4.3 ソフトウェア

ソフトウェアはユーザのプロトタイプに対する操作の学習・認識を行う認識ソフトウェアと外部プラグインによって構成される。

4.3.1 認識ソフトウェア

認識ソフトウェアはハードウェアから得られた音響特性を元にプロトタイプに対する操作の認識を行い、その結果を外部ソフトウェアのプラグインに送信する。認識可能な操作は物体に対するタッチや把持、押し込み等である。詳細は第 6 章の適用例にて示す。認識ソフトウェアには離散値入力型と連続値入力型があり、用途に応じて使い分ける。

図 4.5: 圧電センサ（アクリル版）。

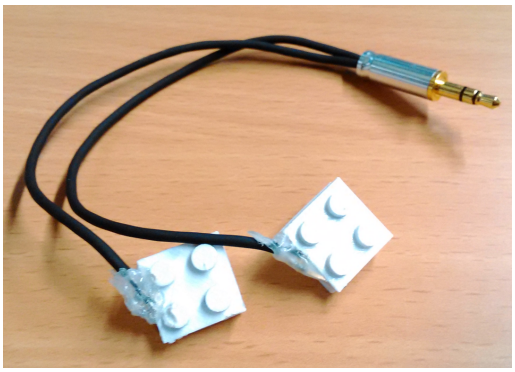

図 4.6: 圧電センサ（レゴブロック版）。

離散値入力型

離散的な動作による操作（例:ボタンを押す）を認識させる場合には離散値入力型の認識ソフトウェアを使用する。離散値入力型の認識ソフトウェアの GUI を図に示す。

GUI の左には現在認識可能な操作のラベルが表示されており、右にはハードウェアから取得した音響特性（円環状のもの）、操作の認識結果、学習の経過が表示される。

ユーザは GUI の左上のテキストボックスにプロトタイプの操作ラベルを入力することによって認識可能な操作を追加することができる。予めハードウェアを物体に取り付け、プロトタイプを構成した上で、操作ラベルのテキストを入力すると認識ソフトウェアは学習モードとなり、GUI の右にプロトタイプへの操作を学習するためのインストラクションを提示する。ユーザはこのインストラクションに従って、自分の使用したい操作を実演し、認識ソフトウェアにそれを学習させる。操作の学習は、1 つの操作につき 5 回の実演によって行われる。ユーザが操作の実演を行う際に、学習の開始と終了は音響特性の情報から自動的に行われる。認識ソフトウェアは操作の学習を終えると自動的に認識モードとなり、これまでに学習した操作を認識するようになる。

また、GUI の左に表示される各操作ラベルにマウスカーソルをあわせると、ポップアップメニューが表示される。このメニューから、各操作の認識精度を高めるために学習データ追加したり、上書きすることができる。またそのメニューには削除ボタンも備わっており、それを選択することによってその操作を認識対象から除くことができる。

連続値入力型

連続的な動作による操作 (例:スライダを動かす) を認識させる場合には連続値学習を行う。離散値入力型の認識ソフトウェアの GUI を図に示す。

画面右に表示される Train ボタンを押すことにより、認識ソフトウェアは学習モードとなり、

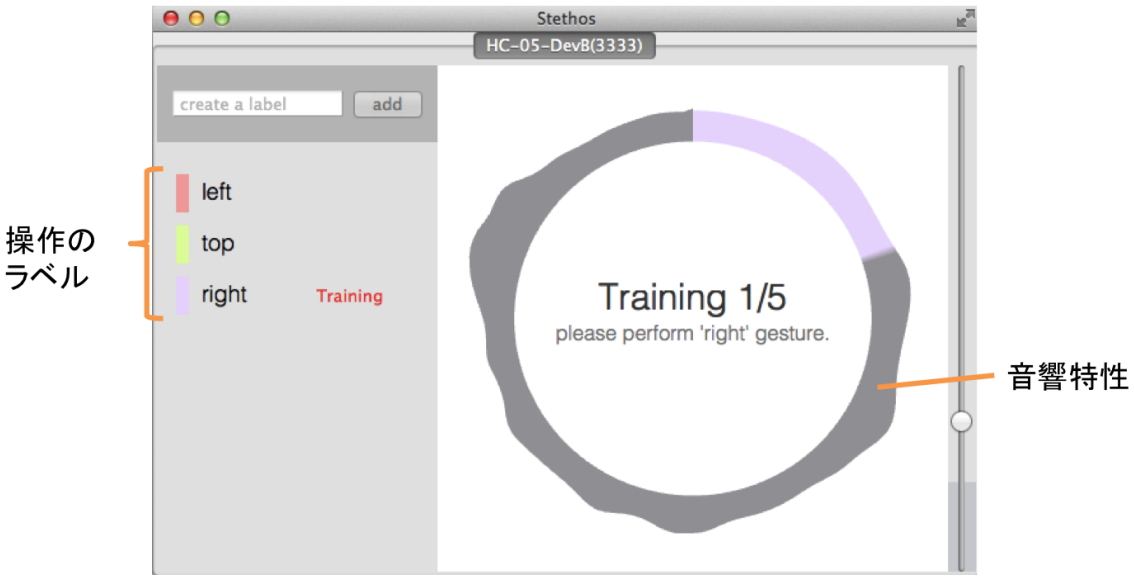

図 4.7: 離散値入力型認識ソフトウェア。

ユーザが学習させたい操作の実演を開始すると画面左にアニメーションガイド（青いバー）が表示される。このアニメーションガイドは捜査開始時点から時間とともに一定速度で上昇し、ユーザはそれに合わせて操作の経過を進める（例：スライダを動かす）。アニメーションガイドが上端まで達すると認識ソフトウェアは認識モードとなり、ユーザが学習させた操作を連続値として認識するようになる。

4.3.2 外部プラグイン

外部プラグインは認識ソフトウェアから送信された認識結果を受信し、外部のヴィジュアルプログラミング環境上でプロトタイプの操作に対する振る舞いを記述するためのプラグインである。

Scratch 用プラグイン

外部プラグインの 1 つとして、Scratch 用のプラグインを作成した。Scratch は構文の学習をせずともプログラムをレゴブロックのように組むことを可能とするビジュアルプログラミング言語である。今回、Scratch 用のプラグインを作成することによって、プログラミング初学者がフィジカルなインタフェースの作成を通して、楽しくプログラミングを学べるようになることを狙いとした。を Scratch はプログラムを記述する構成要素として様々な種類のブロックが存在する。今回のプラグインは Boolean Reporter Block として作成した。このプラグインは現在 Offline 版の Scratch 2.0 で動作するものとなっている。

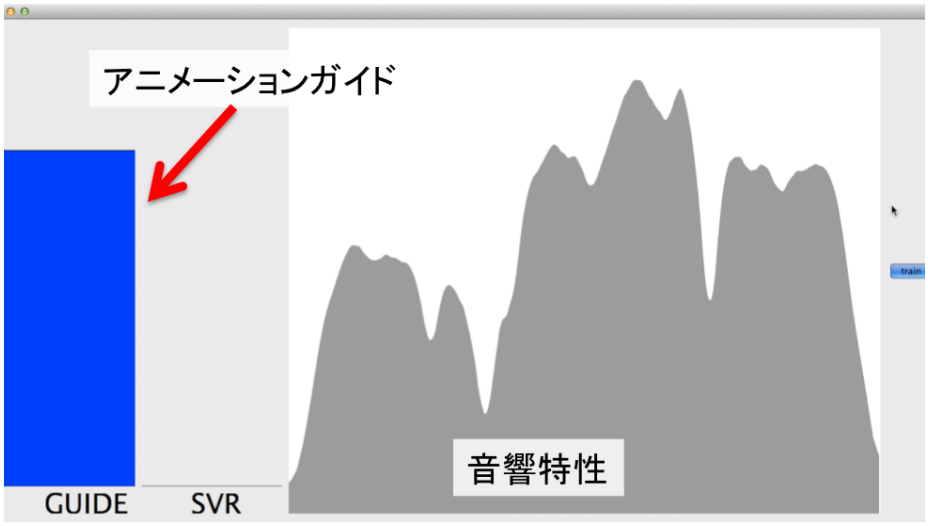

図 4.8: 連続値入力型認識ソフトウェア。

図にプラグインを使った Scratch プログラムの例を示す。黒い菱型のブロックが今回作成したプラグインであり、引数としてプロトタイプの操作ラベル名を受け取る。このプラグインは HTTP 通信によって認識ソフトウェアから操作の認識結果を受信している。受信した認識結果のラベル名が引数に記述したラベル名と一致していれば真、そうでなければ偽を返すものとして、Scratch プログラムの構成要素として用いることができる。

このプログラム例では、ラベル「left」の認識結果が認識ソフトウェアから発信されたときに、キャラクタが 10 歩進むものとなっている。

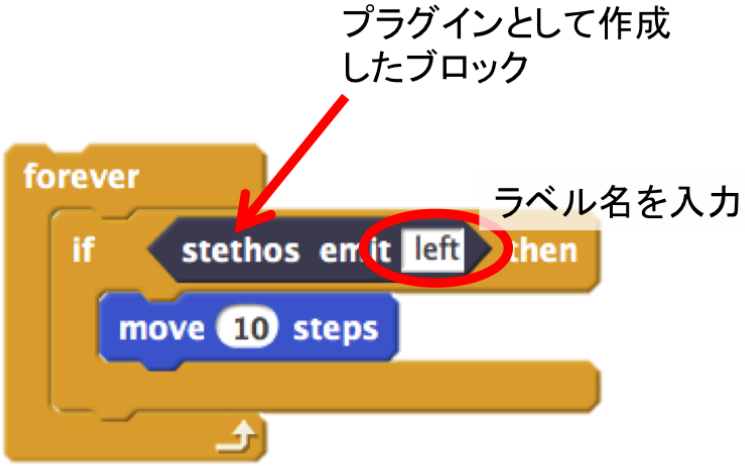

図 4.9: Scratch 用プラグインの使用例。

Max/MSP・PureData 用プラグイン

もう 1 つの外部プラグインとして、Max/MSP･PureData 用のプラグインを作成した。Max/MSP および PureData は音や映像を扱うビジュアルプログラミング環境であり、メディアアーティストによく使用される環境の一つとなっている。これらのソフトウェアに対応することによって、本ツールキットをメディアアーティストに使ってもらうことを狙いとした。

図に Max プラグインの使用例（3 種類のタッチ入力に対して異なる効果音を鳴らすパッチ）を示す。図の上部のオブジェクトがプラグインとして作成したオブジェクトである。このオブジェクトは認識ソフトウェアの認識結果を OSC 通信によって受信するため、その第 1 引数[1] には OSC のポート番号を入力する必要がある。1 つのハードウェアを用いる場合、OSC のポート番号は 3333 を使用する。複数のハードウェアを使用する場合、その数に従ってポート番号を 1 つずつ増加させる必要がある（3334、3335 等）。第 2 引数以降はプロトタイプの操作ラベル名を入力するものとなっている。図の例ではオブジェクトに 4 つの outlet があり、認識ソフトウェアの認識結果に対応した outlet から bang が送信されるものとなっている。これらの bang をトリガとして、Max/MSP 上で様々な振る舞いを記述することが可能である。

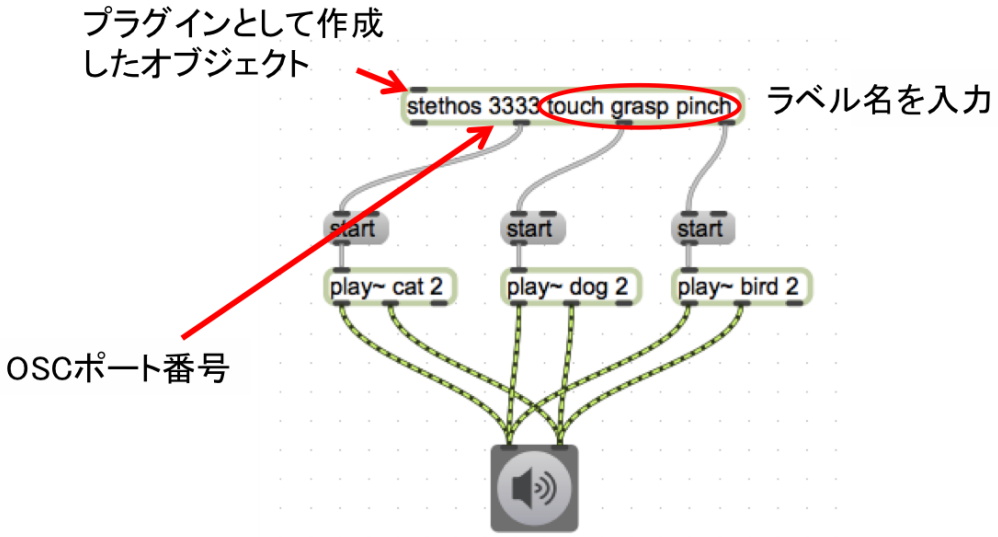

図 4.10: Max プラグインの使用例。

4.4 使用例

本節では本ツールキットを用いたプロトタイピング例を示す。この例では既存物体であるキャラクタ玩具を用いて、再生/一時停止、曲送り、曲戻し機能を持つ音楽プレイヤを作成す

[1] stethos はオブジェクト名であり、引数ではない。

る。全体の流れを図に示す。

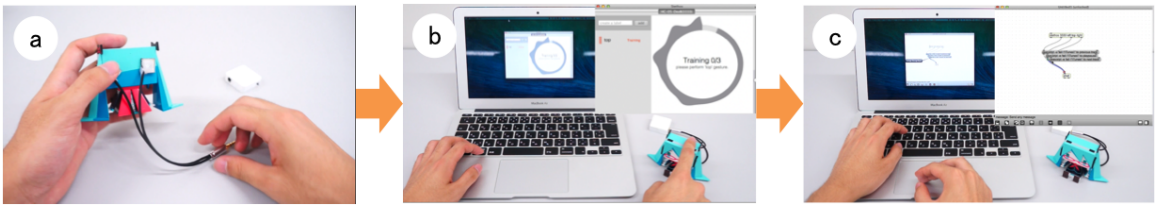

図 4.11: 本ツールキットを用いたプロトタイピングの流れ。a) ハードウェアの取り付け、b) 操作の学習、c) アプリケーションとの関連付け。

4.4.1 ハードウェアの取り付け

まずはじめにキャラクタ玩具にハードウェアを取り付ける（図 a)。今回、ユーザは両面テープ（3M SPG-12）を用いて圧電センサを玩具に貼り付けている。また、専用インタフェースモジュールと圧電センサ、コンピュータと専用インタフェースモジュールをそれぞれ接続する。今回の玩具はプラスチック製で高さ約 5cm のものを使用したが、本ツールキットがサポートする既存物体には、材質（図）や大きさ（図）についての制約があり、プラスチックや金属等の硬い材質かつ手のひら大の物体が適用対象として望ましい。

図 4.12: 適用可能な物体の制約（材質)。

4.4.2 操作の学習

ハードウェアの取り付けが完了したら、次にプロトタイプの操作の学習を行う（図 b)。この例では、離散値入力型の認識ソフトウェアを用いて、キャラクタに対する 3 種類のタッチ

図 4.13: 適用可能な物体の制約（大きさ）。

ジェスチャ（left, right, top）を学習させている。それぞれのジェスチャを図に示す。各ジェスチャを 5 回ずつ行うことによって、認識ソフトウェア上でそれらを認識できるようにする。

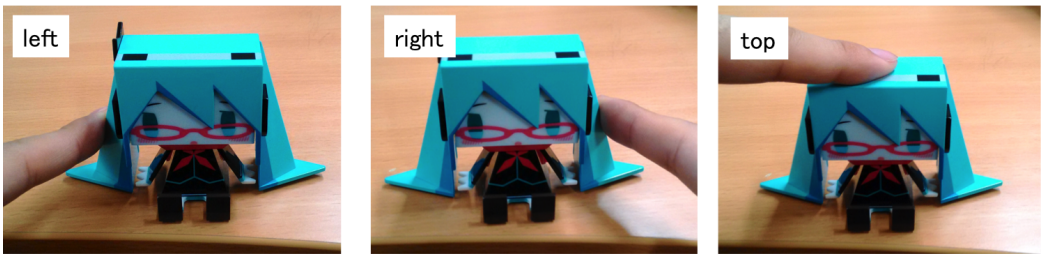

図 4.14: 音楽プレイヤの操作のための 3 種類のタッチジェスチャ。

4.4.3 アプリケーションとの関連付け

操作の学習を終えたら、次にアプリケーションとの関連付けを行う（図 c）。この例では Max 用プラグインを用いて音楽プレイヤのパッチを作成している。プラグインのオブジェクトの第 2 引数以降に前節で学習させた操作ラベル（left, right, top）を入力する。それらの outlet を iTunes を操作するためのスクリプトの message オブジェクトを経由して shell オブジェクトに接続する。今回は left の outlet を曲戻し、right の outlet を曲送り、top の outlet を再生/一時停止のメッセージを割り当てることによって音楽プレイヤを作成した。作成した Max/MSP のパッチを図に示す。

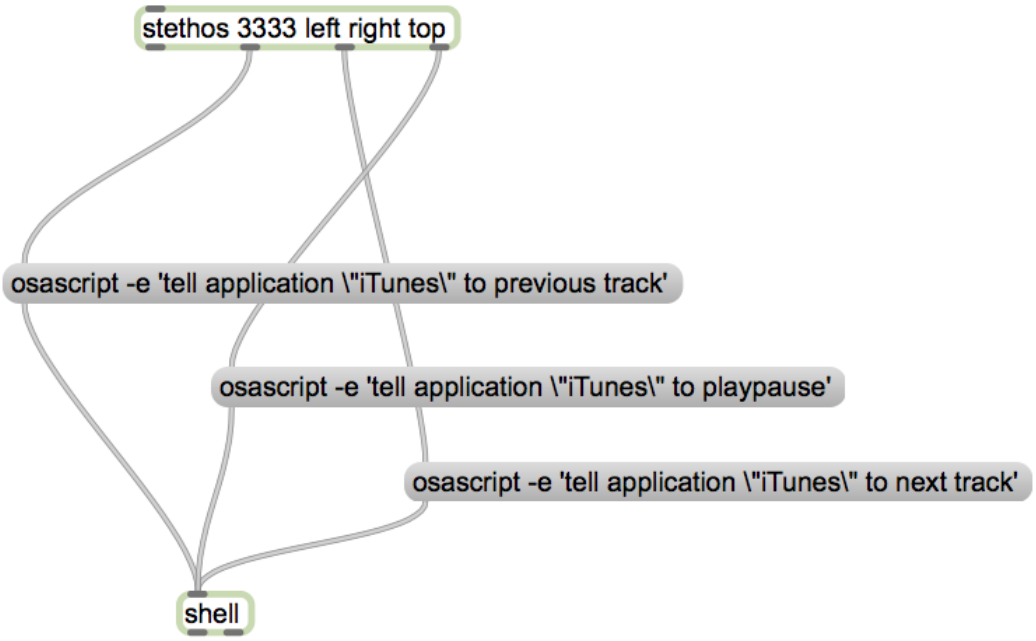

図 4.15: 音楽プレイヤーの Max パッチ。

第5章　実装

本章では、提案手法に基いて作成したプロトタイピングツールキットの実装方法を述べる。

5.1　ハードウェア

専用インタフェースモジュールと圧電センサの実装方法を以下に示す。

専用インタフェースモジュール

専用インタフェースモジュールのブロック図を図に回路図を図、に示す。専用インタフェースモジュールは mbed 互換のマイクロコントローラ（LPC11u24）によって制御される。

マイコンは波形生成 IC（AD5930）に制御信号を送ることによって 20kHz〜40kHz のスイープ信号を生成する。AD5930 は CTRL ピンに周波数インクリメント用のトリガ信号を送ることによって、生成する信号の周波数を予めた決められた量だけインクリメントさせることができる。今回は 150 回のインクリメントによって 20kHz から 40kHz へと変化するように設定した（1 インクリメントあたり約 133Hz 増加する）。LPC11u24 の動作周波数で AD5930 を制御した場合、約 5ms で 1 周期分のスイープ信号が生成されることになる。生成された信号はオペアンプ（AD8541）によって 3.3V の振幅の信号へと増幅され、出力側の圧電センサを振動させる。この時の振動は 20kHz〜40kHz の振動であるため、音声として人間の耳に聞こえることはない [Chr03]。

入力側の圧電センサは物体を介して伝わってきた振動応答を電気信号へと変換する。このときの信号は微小（数 mV）であるため、オペアンプ（AD8618）を用いて増幅する。このアンプのゲインはデジタルポテンショメータを通してマイコンから制御できるようになっている。また、このオペアンプは単電源で動作しているため、振動応答の波形の下半分はカットされるが包絡線を抽出する分には特に影響はない。増幅された信号は包絡線検波回路によって包絡線として検波される。基本的な包絡線検波回路はダイオード、コンデンサ、抵抗によって構成される（図）が、ノイズからの影響を受けにくくするためこの回路のコンデンサ部分を 3 次の Shellkey ローパスフィルタ（図）に置き換えた。検波された波形はマイコンの A/D コンバータによってデジタルデータとして取り込まれる。この時、検波された波形がクリッピングされているようであれば、波形が A/D コンバータの許容内におさまるようにオペアンプのゲインを調節する。周波数インクリメントをする度に波形のサンプリングを行うことにより、1 周期のスイープ信号につき 150 次元の音響特性データを取得することができる。この

データをシリアル通信（USB もしくは Bluetooth）を用いてコンピュータへと送信する。コンピュータへ送信する間隔は約 30Hz である。

以上の回路をブレッドボード上でテストし、動作を確認した後プリント基板上に実装した。実装した基板を図、図に示す。これらの基板を 3D プリンタで作成したケースに梱包することによって図および図のように仕上げた。

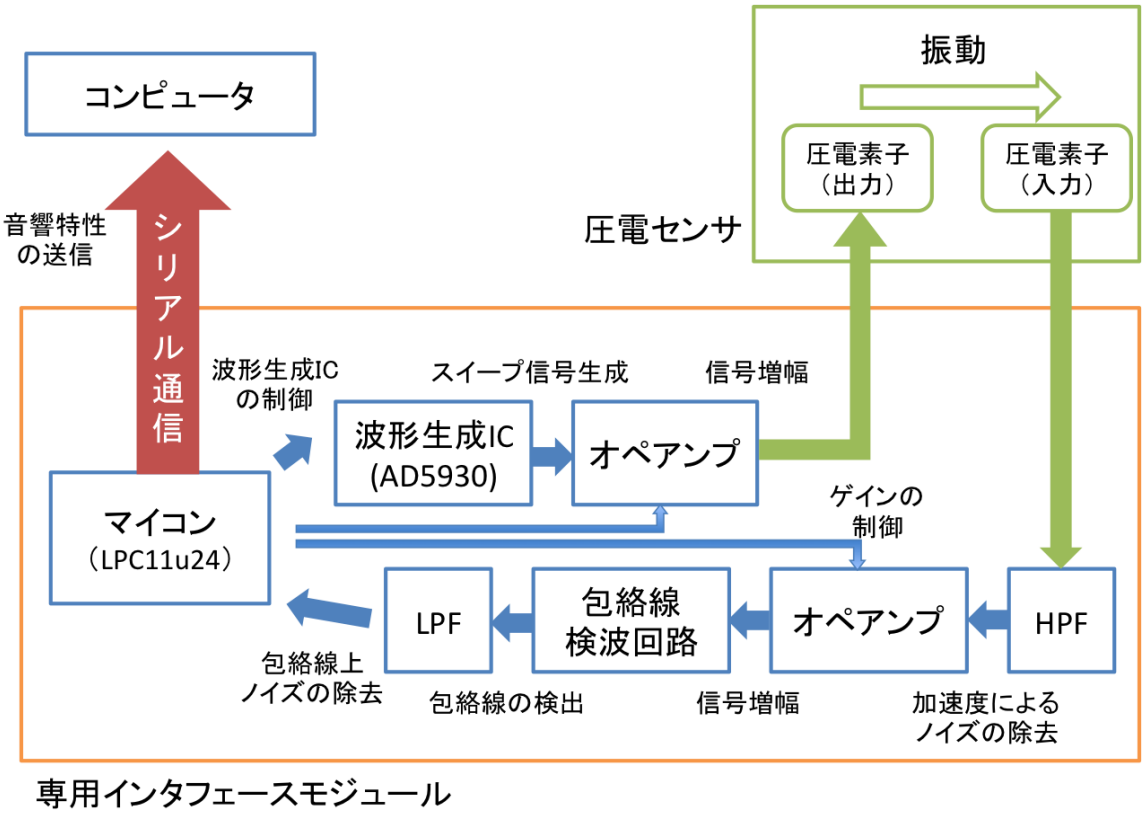

図 5.1: 専用インタフェースモジュールのブロック図。

5.1.1 圧電センサ

アクリル版

アクリル版の圧電センサは圧電素子（SRIVE K2512BP1、幅 25mm,、高さ 12mm、厚さ 0.23mm）を 2 つに切断し、2 つのアクリル板（幅 12mm、高さ 12mm、厚さ 3mm）に接着剤で接着することによって作成した。圧電素子を直接使用せずにアクリル版に貼り付けて使用することによって、圧電センサを物体に取り付けたりはずしたりする際に、アクリル部分を指で持つことによって圧電素子部分の損傷を軽微にするためである。また、各圧電素子の 2 つの電極と 4 極プラグの各端子をシールドケーブルによって配線した。

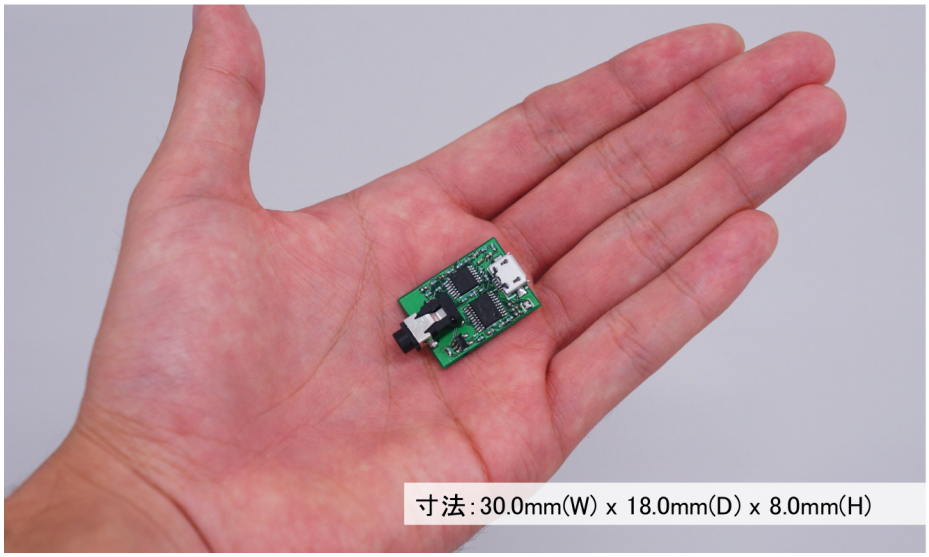

図 5.4: 専用インタフェースモジュールの基板（USB 接続版）。

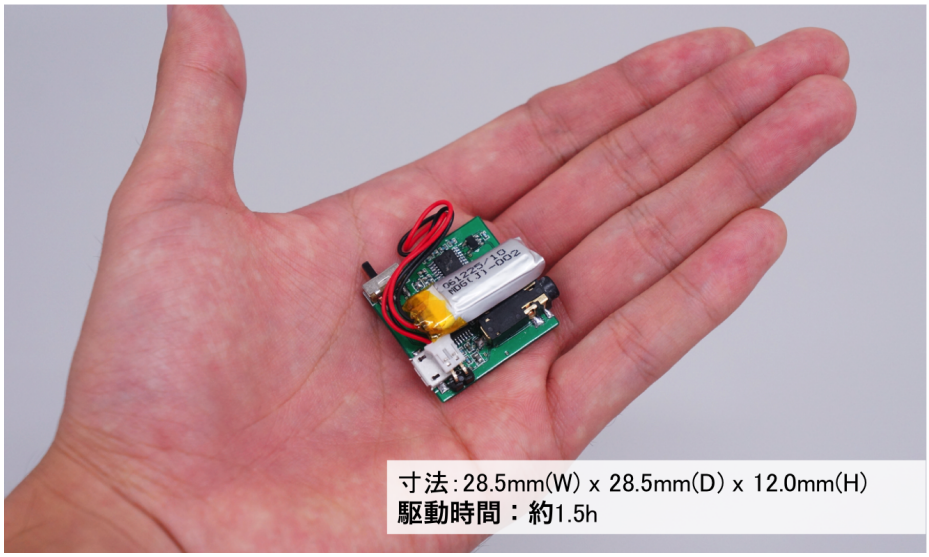

図 5.5: 専用インタフェースモジュールの基板（Bluetooth 接続版）。

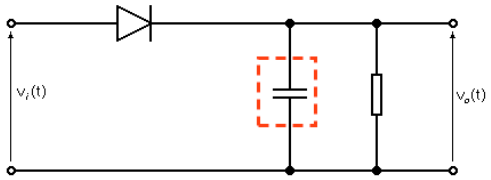

図 5.6: 一般的な包絡線検波回路。

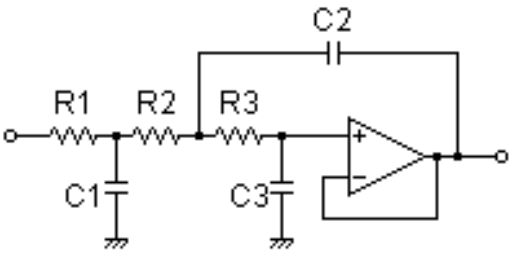

図 5.7: 3 次 Shellkey ローパスフィルタ。

レゴブロック版

レゴブロック版の圧電センサについては、3D プリンタで 2×2 のレゴブロックと同等のモデルを途中まで出力し、途中で一時停止した状態において 2 つに切断した圧電素子（SRIVE K2512BP1）を接着剤で出力物の上に固定し、再びプリントを再開することによって作成した（図）。このように 3D プリント中に他の素材や素子等を挿入する製造技術は [WBHP12] や [KR01] で使用されている。なお、この圧電センサは、レゴブロックだけでなく、そのサイズが 2 倍となるデュプロブロックにも同様に取付可能である。配線についてはアクリル板と同様である。

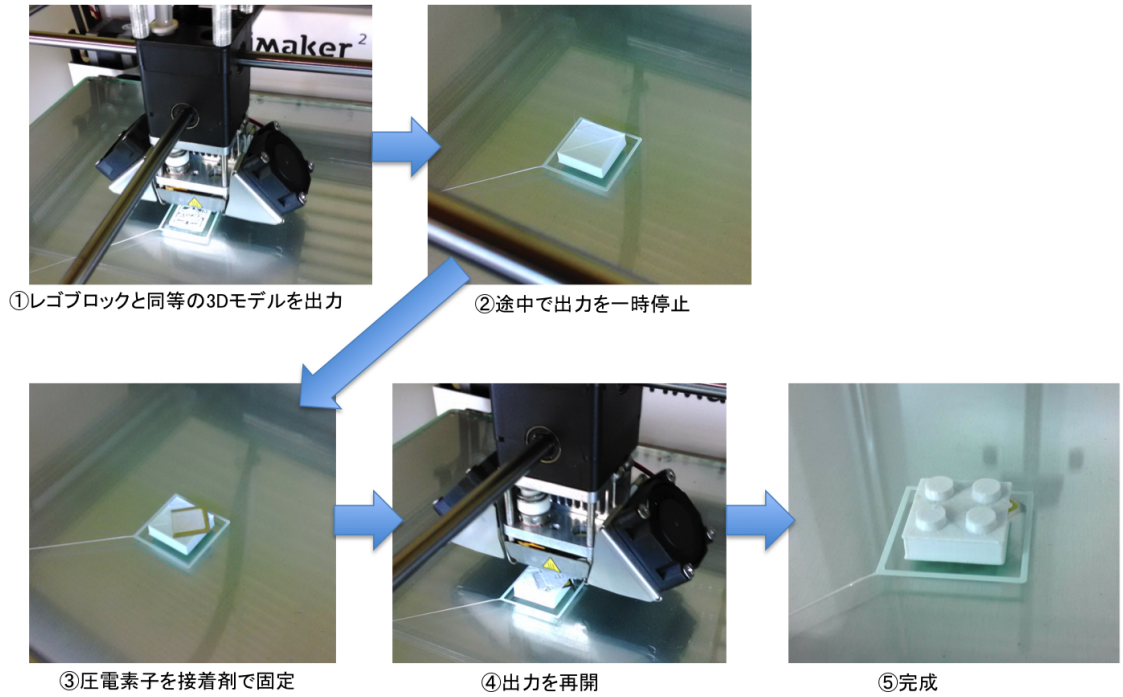

図 5.8: レゴブロック版圧電センサの作成手順。

5.2 ソフトウェア

認識ソフトウェアと外部プラグインの実装方法を示す。

5.2.1 認識ソフトウェア

認識ソフトウェアを Qt/C++[1] を用いて実装した。以下に認識ソフトウェアの主な機能の実装方法の詳細を示す。これらのうち、「音響特性データの取得」の項は離散値入力型ソフトウェアと連続値入力型認識ソフトウェアと共通の実装である。また、「離散値入力の学習」の項は離散値入力型認識ソフトウェアの学習部分の実装、「連続値入力の学習」の項は連続値入力型認識ソフトウェアの学習部分の実装をそれぞれ示している。

音響特性データの取得（専用インタフェースモジュールを使用する場合）

専用インタフェースモジュールを使用する場合、認識ソフトウェアはハードウェアとのシリアル通信によって音響特性データを直接取得する。今回シリアル通信を行うためのモジュールとして QtSerialPort を用いた。これを用いることにより、Bluetooth や USBSerial による通信を一つのモジュールで管理することが可能である。音響特性の 1 フレーム分のデータは長さ 150 の float 型配列として読み込まれる。

音響特性データの取得（オーディオインタフェースを使用する場合）

オーディオインタフェースを使用する場合、認識ソフトウェアは振動応答を取得するためのスイープ信号の生成も行う。スイープ信号の生成は、96kHz のサンプリングレートで 20ms 間に 20kHz から 40kHz までスイープする正弦曲線のスイープ信号を生成する。また、オーディオインタフェース経由で取り込まれる信号は振動応答であるため、これを音響特性に変換する必要がある。音響特性への変換は振動応答の波形にフーリエ変換をかけることによって行う。今回、振動応答を 96kHz のサンプリングレートで取り込み、8192 サンプルごとにフーリエ変換を行う。今回、フーリエ変換を行う際の窓関数としてハニング窓を使用した。これにより 4096 点の 0Hz から 48kHz の周波数スペクトルが抽出される。この中から 20kHz から 40kHz の部分のみを取り出し、ダウンサンプリングすることによって長さ 400 の float 型配列として音響特性を抽出する。

離散値入力の学習

離散値入力型認識ソフトウェアにおいて、離散値入力の学習は、まず学習用データの取得から行う。ユーザがテキストボックスに操作ラベル名を入力した際に、認識ソフトウェアは内部的にユーザのプロトタイプ操作待ち状態になり、その瞬間の音響特性データを「操作な

[1] http://qt-project.org/

し状態における音響特性データ」として記録する。その後、ハードウェアから逐次送られてくる音響特性データと「操作なし状態における音響特性データ」の比較を行う。もしそれらのデータ間において差が一定以上あるようであれば、それをユーザが入力した操作ラベルの操作によるものとみなす。仮にこの操作ラベルが「A」であるとした場合、この間の音響特性データを「A 操作状態における音響特性データ」として記録する。このデータのフレーム数が 10 フレームに達するまで記録を続ける。記録する音響特性データが 10 フレームに満たずして「操作なし状態における音響特性データ」に戻った場合は、それを誤操作によるキャンセルとみなし、それまでに記録した「A 操作状態における音響特性データ」を破棄する。10 フレームに達してから「操作なし状態における音響特性データ」に戻った場合にはそれを正常な学習用データとして保存する。これを 1 つの操作ラベルにつき 5 回繰り返す。5 回の音響特性データの取得が完了すると、「操作なし状態における音響特性データ」および「A 操作状態における音響特性データ」に固有の教師ラベルを振り、0〜1 の値をとるように正規化を行った後にそれらから SVM の学習モデルを構築する。今回 SVM の実装には LIBSVM [CL11] を用いた。SVM-type には C-SVC、kernel タイプには rbf カーネルを採用し、その他パラメータはデフォルトのものを用いた。学習モデルの構築を終えると認識ソフトウェアは認識モードになる。認識モードでは、ハードウェアから送られてきた音響特性データを学習モデルに与えることによって操作を認識する。

連続値入力の学習

連続値入力型認識ソフトウェアにおいて、連続値入力の学習を行う。まず学習用データの取得から行う。連続値入力型ソフトウェアにおいて、Train ボタンを押した時にユーザのプロトタイプ操作待ち状態になり、ユーザによるプロトタイプの操作の検出を行う。検出のアルゴリズムについては離散値入力型認識ソフトウェアと同様に、音響特性データの差分を求めることによって行う。もしプロトタイプへの操作を検出した場合は、教師ラベル用の float 型の変数の値を 33ms ごとに 0.03 ずつ増加させることによって、0〜1 の間で変化させる。同時に、この時の教師ラベルの値に基いてアニメーションガイドのバーの高さを変化させる。ユーザはこのバーの高さに応じてプロトタイプを操作するので、それにともなって音響特性が変化する。よって提示した教師ラベルの値に対する音響特性データを対として記録する。これによって教師ラベルの値が 1 に達するまでに合計 33（=1/0.03）の学習データが得られることになる。これらのデータを正規化した後に SVM の学習モデルを構築する。連続値入力の場合には SVM-type に ε -SVR を用いる。SVR は SVM を回帰に応用したもので、SVC とは異なり連続的な出力を推定することに特化している。kernel タイプには rbf カーネルを採用し、その他のパラメータはデフォルトのものを用いた。学習モデルの構築を終えると認識ソフトウェアは認識モードになり、ハードウェアから送られてきた音響特性データを学習モデルに与えることによって、ユーザの連続的な操作を連続値として推定する。

5.2.2 Scratch 用プラグイン

Scratch には Web ブラウザ上で起動するオンライン版と、デスクトップで起動するオフライン版が存在する。プラグインの実装方法はオンライン版とオフライン版によって異なる。オンライン版は JavaScript を用いるのに対し、オフライン版は HTTP 通信を使用する。今回作成したプラグインはオフライン版で動作するものである。オフライン版のプラグインを作成する際には、通信内容を定義する JSON 形式のファイルと、実際に通信を行う中間ソフトウェアが必要となる。これにより Scratch はプラグインの JSON ファイルの定義に従って中間ソフトウェアと通信を行い、中間ソフトウェアが扱う外部デバイスの入力等を Scratch プログラム上で扱うことが可能となる。

JSON ファイル

作成した JSON ファイルを図に示す。

```
1 {
2         "extensionName": "stethos3333",
3         "extensionPort": 3333,
4         "blockSpecs": [
5                 ["b", "stethos emit \%s ", "stethos_label", "
                     default"]
6         ]
7 }
```

図 5.9: Scratch プラグインの JSON コード。

このコードにおけるキー “extensionName”に対する値 “stethos3333”はプラグインの名前を示す。キー “extensionPort”に対する値 3333 は中間ソフトウェア HTTP 通信を行う際に使用するポート番号を示す。キー “blockSpecs”に対する値には Scratch ソフトウェア上で表示される拡張ブロックの仕様が記述される。拡張ブロックの仕様は配列として定義され各要素は以下のような意味を持つ。

第 1 引数 ブロックの種類。“b”は Boolean Reporter Block を表す。

第 2 引数 ブロックのフォーマット。ブロックに表示されるラベルやパラメータスロットを記述する。パラメータスロットは “%”で始まる語で定義される。“%s”では文字列型のパラメータスロットを指定している。

第 3 引数 中間ソフトウェアと通信する際に使用する変数名である。

第 4 引数 パラメータスロットの初期値を指定する。

このプラグインではパラメータとして認識ソフトウェアの認識結果のラベルを“stethos_label”という変数名で受け取るように記述されている。そして受け取った結果がパラメータスロットに記述されているものと等しければ真、そうでなければ偽を出力する。

中間ソフトウェア

Scratch と通信を行う中間ソフトウェアを認識ソフトウェア内のモジュールとして実装した。このモジュールは認識ソフトウェアの認識結果を Scratch と通信するためのフォーマットに整形して HTTP 通信にて送信する。

通信は Scratch がコマンドを中間ソフトウェアに送信し、中間ソフトウェアがそれに対する応答としてセンサの値やステータス情報を HTTP GET リクエストとして送り返すものとなっている。具体的な通信の手順は以下の通りである。

polling

Scratch が中間ソフトウェアに対して/poll という URL の HTTP GET リクエストを送信する。

コマンドの返信

中間ソフトウェアは/poll に対してコマンドを Scratch に返信する。コマンドのフォーマットは以下の様な URL リクエストの形式になっている。

/stethos_label/label

この URL のうち stethos_label 部分については JSON コードに記述した通信に使用するラベル名である。label 部分については認識ソフトウェアの認識結果のラベルを記述する部分となっている。

polling は約 30Hz の間隔で行われるため、その程度の時間分解能で応答する Scratch 上のアプリケーションを記述可能である。

5.2.3 Max/MSP・PureData 用プラグイン

Max/MSP・PureData のプラグインの実装には flext[2] というフレームワークを利用した。PureData は Max/MSP のオープンソース版として開発されたため、ビィジュアルプログラミング言語としての文法と挙動はほぼ同じである。しかしながら、プラグインを作成する場合には本来別々のコードを書く必要がある。この問題を解決するためのフレームワークが flext であり、これを用いることにより 1 つの C++コードで Max/MSP、PureData 双方のプラグインを生成可能である。

作成したプラグインは Max/MSP・PureData 上でオブジェクトとして生成される際に引数をチェックする。第 2 引数以降には認識結果の候補のラベルが入力されているので、それらのラ

[2]http://puredata.info/Members/thomas/flext

ベルを配列に保存しておき、それらのラベルに対応付けられた outlet を生成する。その後プラグインは OSC 通信を介して認識ソフトウェアから認識結果を受信する。OSC 通信を行うための C++ライブラリとして oscpkt を用いた。具体的な通信の手順は以下の通りである。

認識結果の送信

認識ソフトウェアは認識結果を受けて、以下のようなフォーマットの OSC メッセージを送信する。

/stethos label

このメッセージにおいて label 部分に認識結果を記述して送信する。

認識結果の受信

プラグインは OSC メッセージを受信し、アドレスが/stethos となっているものを抽出する。

次にプラグインは抽出したメッセージから認識結果を表す label をパースする。パースされた認識結果とプラグイン生成時に保存した認識結果の候補を照合し、いずれかにマッチするようであればそれに対応する outlet から bang メッセージを出力する。これらの処理をメッセージを受信するたびに行う。メッセージの受信間隔は約 30Hz である。

第6章　適用例

本章では、本研究で開発したプロトタイピングツールキットによってどのようなプロトタイプが作成可能であるかについて、4 種類の分類に基いて述べる。

6.1　タッチ入力インタフェース

物体に触れることによってその物体の音響特性は変化する。またそのときの触れる場所によって音響特性の変化の仕方は異なる。これらの変化を本手法を用いて学習することによってタッチ入力インタフェースのプロトタイピングを容易に行うことができる。本節ではそのような適用例について述べる。

6.1.1　タッチ認識技術に関する研究

近年のタッチ認識技術はタッチ認識をプロトタイプに用いるための敷居を下げ、そのセンシング能力も広がっている。FTIR [Han05] は Frustrated total internal reflection を用いてマルチタッチサーフェースを低コストに作成する手法である。OmniTouch [HBW11] はあらゆる面においてマルチタッチアプリケーションを利用可能にするウェアラブルなデプスセンシング・プロジェクションシステムである。これらのビジョンに基づくタッチ認識技術はタッチ位置を高精度に認識できるものの、ハードウェアの規模が大きく、また画像処理のための高速なプロセッサを必要とする。Touché [SPH12] や Capacitive Fingerprinting [HSP12] は周波数を掃引させた静電容量センシングにより、既存の導電性の物体に触れる際の把持姿勢の認識やユーザの識別が可能であることを明らかにした。しかしながら Touché は物体に導電性を要求するため、非導電性の物体上でそれらの認識を行うためには導電インクやテープを用いたコーティングを必要とする。

本手法で用いるタッチ認識技術は、簡単なハードウェア構成を用いて非導電性の物体を含む既存物体上においてで様々な種類のタッチジェスチャを認識することができる。

6.1.2　適用例

タッチに反応する玩具

100 円ショップで売られていたプラスチックのキャラクタ玩具に本手法適用して、その玩具に触れると触れ方によって異なる声を出すプロトタイプを作成した。図に示すような 4 種類の

タッチジェスチャに対して離散値入力学習を行い、Max/MSP を用いて効果音を割り当てた。

楽器

本手法を用いてピアノの音階を演奏することがきる楽器のプロトタイプを作成した（図）。図に示すようなキャラクタの玩具の台座が鍵盤の形をしていたため、それを適用対象として利用した。各鍵盤をタッチした状態において離散値入力学習を行い、Max/MSP を用いて音階を割り当てた。

ゲームコントローラ

本手法をセラミックのお椀に適用し、ゲームコントローラのプロトタイプを作成した（図）。お椀の左右に対するタッチを認識できるように離散値入力学習を行い、Scratch を用いてゲームアプリケーションを作成した。このゲームアプリケーションはキャラクタを左右に操作し、上から落ちてくるりんごを制限時間以内になるべく多く集めるゲームである。

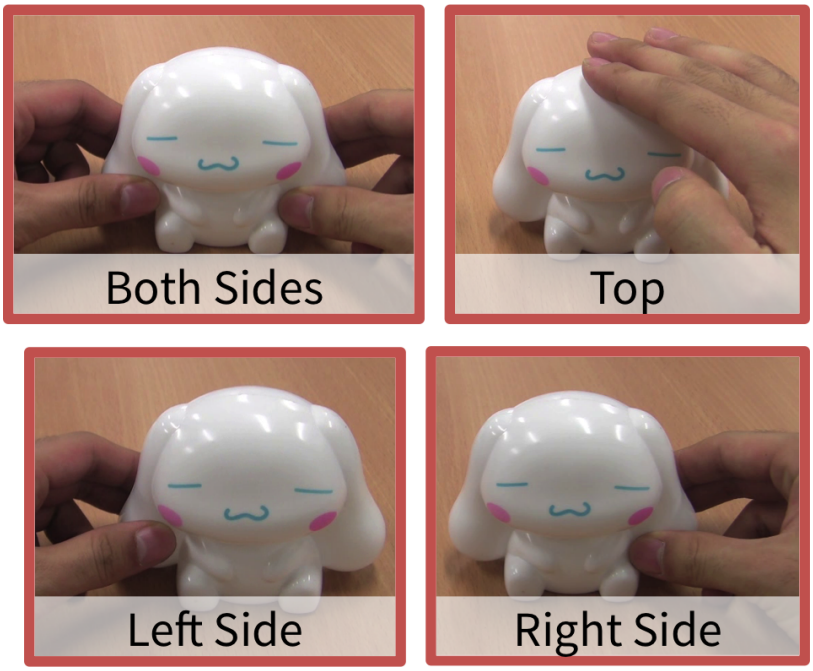

図 6.1: タッチに反応する玩具。

6.2 把持認識インタフェース

タッチより多くのコンテキストを含む把持を認識して HCI への応用を試みる研究が数多く行われている. 物体の音響特性は把持によっても変化するため、本手法を用いれば物体の把持

図 6.2: 楽器のプロトタイプ。

図 6.3: 使用したキャラクタの玩具。

図 6.4: お椀を用いたゲームコントローラ。

を入力として扱うようなインタフェースのプロトタイピングも可能である。

6.2.1 把持認識技術に関する研究

既存の把持認識を行う研究は大量のセンサや特殊なハードウェアを用いており複雑な構成となる。例えば、Kim らの研究では 64 個の静電容量センサを携帯情報端末に組み込むことにより、端末の把持姿勢を認識する [KCC+06]。FlyeEye [Wim10] は光ファイバの端を面に拡散して配置し、反対側の末端を収束させてカメラで取り込み、画像処理を行うことによって、把持状態を認識可能なデバイスのプロトタイピング手法を示している。Taylor らは 23〜72 の静電容量センサををデバイスの面に設置してどのようにフィジカルな物体の持ち方を操作に応用できるか探求した [TB09]。。HandSense [WB09] は携帯情報端末のサイドに 4 つの静電容量センサを設置し、端末がどのような状態にあるか（把持されている、ポケットに入っている）を検出している。また、MTPen [SBG+11] は 200 個の静電容量センサアレイを用いてペンの把持姿勢を認識している。FlexAura [LG12] は 16×24 個のフレキシブルなフォトリフレクタのアレイを用いてペンのグリップや状態を認識している。

これらの研究では大量のセンサや特殊なハードウェアを用いるのに対し、本手法を用いれば、1 組みの圧電素子を用いた簡単なハードウェア構成によって把持認識を扱うインタフェースのプロトタイピングが可能である。

6.2.2 適用例

携帯電話のケース

多くの研究者が把持状態を認識可能な携帯情報端末を開発し、それによってどのようにユーザインタフェースが改善されるかを探求している [CHLC12, GWP12, KCC+06, TB09, WB09]。ほとんどの研究者は端末の把持姿勢や把持力を認識するために複数のタッチセンサを用いている。GripSense [GWP12] という研究に関しては端末に予め内蔵されたセンサのみを用いて把持姿勢や把持力を認識可能であるが、把持姿勢を認識する際にユーザに画面のスワイプ操作を要求する。

本手法を用いれば、携帯情報端末の把持姿勢や把持力の認識を行うことも可能である。この適用例では iPhone4S 用のハードケースに 2 つの圧電素子を取り付けることにより、把持認識可能なケースを作成した（図）。2 つの素子とケーブルの厚さにより、ケースを本体に取付た際に不安定な構造になるため、6 つのプラスチックビーズをケース底面に取り付けることにより、底面の厚さを均一化した。以上の構成を用いて、把持姿勢と把持力の認識を行った。

近年の携帯情報端末のアプリケーションを使用する際に独特の把持姿勢を伴うものがある。例えば、電話をする間、ユーザの親指は端末の側面に沿い、他の 4 本指が反対側の側面に触れるような把持姿勢になる。カメラを使うときは端末の 4 隅を摘むような把持姿勢になる。したがって、もしこれらの把持姿勢を認識できれば、持ち方に応じてアプリケーションを自動的に切り替える等の応用が可能になる [KCC+06, WB09]。加えて、異なる把持姿勢において、

タッチの届く範囲等が異なるため、端末の操作性や使い方に影響する [AZ12]。もしデバイスがユーザの把持姿勢を認識できれば、その把持姿勢において最も使いやすくなるように UI を調節することが可能になる（例：キーのレイアウト [AZ12] や、スクリーンローテーション [CHLC12] の最適化）。本手法はそのようなインタラクションを探求したい研究者に対して、デバイスのプロトタイピングを行うための簡単なハードウェア構成を提供する。我々は 3 種類のアプリケーション（Call、SMS、Camera）と 3 種類の持ち手（Left handed、Right handed、Two-handed）の組み合わせから構成される 6 種類の把持姿勢（図）を認識するようにシステムを学習させた。

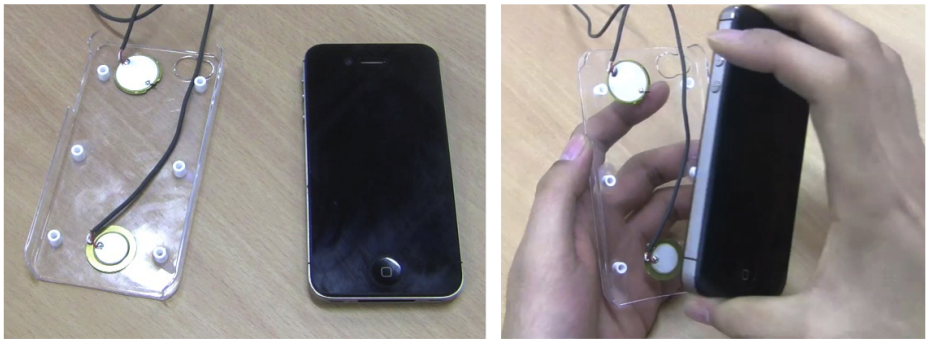

図 6.5: 本手法適用した携帯情報端末ケースの作成。

図 6.6: 携帯情報端末の 6 種類の把持姿勢。

複数機能マウス

デバイスの把持状態によって機能を選択する入力デバイスがいくつか提案されている [VC11, RS00]。この適用例ではデュプロブブロックに小型のトラックボール（Sanwa Supply 400-MA018）を組み込むことによりによりペイントアプリケーション向けにデザインされた複数機能マウスのプロトタイプ作成した（図）。マウスカーソルの動きは USB Overdrive[1] というソフトウェアを用いて水平方向に反転させている。このデバイスは 5 種類の入力ツールを把持状態によって選択することができる（図）。例えばユーザは先端の黄色いブロックをつかむことにより、ルーペツールが選択される。

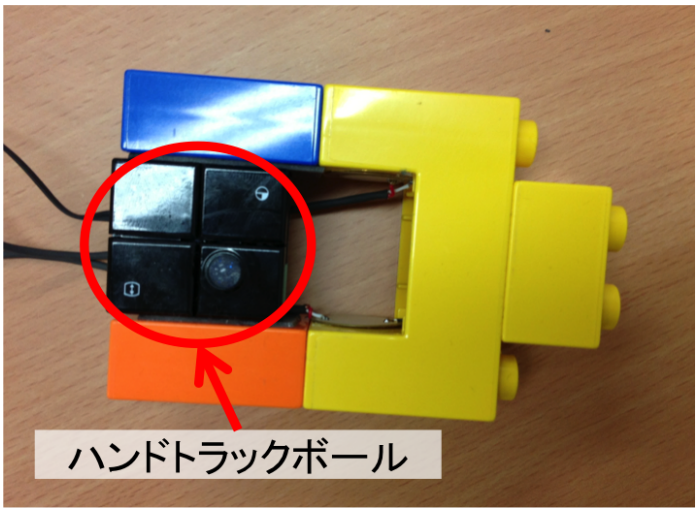

図 6.7: 本手法を適用した複数機能マウスのプロトタイピング。

[1] http://www.usboverdrive.com/USBOverdrive/Information.html

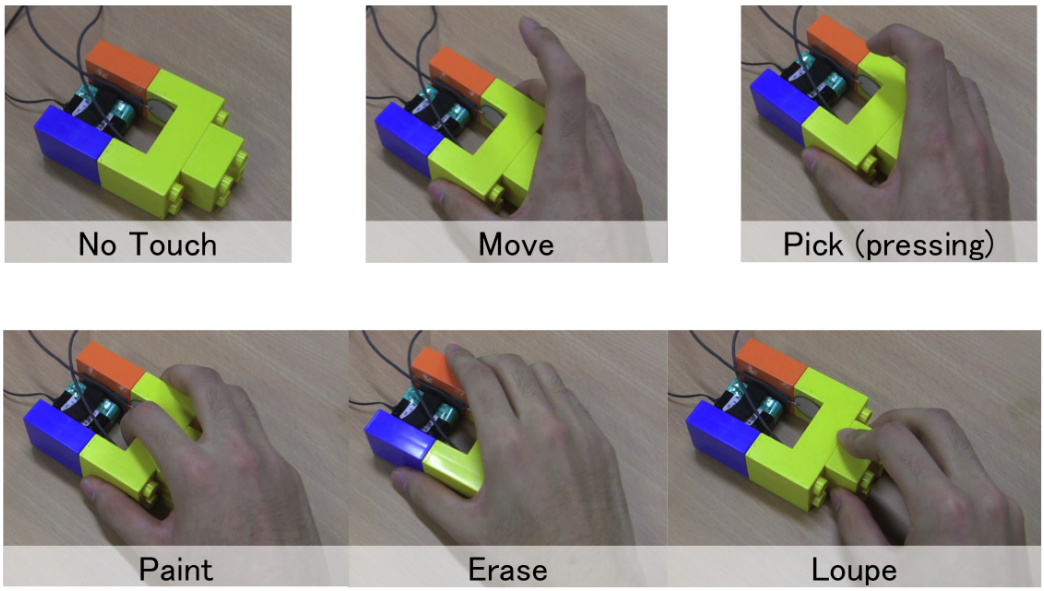

図 6.8: 複数機能マウスの把持方法と対応するツール。 No Touch) マウスを使用していない状態。Move) カーソルを移動させるツール。マウスのサイドを持つことによって機能する。 Pick) ドラッグを行うツール。マウスの持ち手に力を加えることによって機能する。なにもないところをドラッグすると選択範囲が表示されそれを再びドラッグすることによって選択範囲を移動させることができる。 Paint) ペイントを行うツール。線を書くことができる。マウスの中央の溝に指を引っ掛けることによって機能する。 Erase) 線を消すためのツール。マウス全体を把持することによって起動する。Loupe) ズームを行うツール。マウスカーソルの周囲のキャンバスを拡大することができる。マウスの凸型部分の先端把持することによって起動する。

6.3 圧力入力インタフェース

物体の音響特性は、タッチや把持を行う際に圧力を加えることによっても変化する。これを本手法における連続値入力学習を用いて学習させれば、そのような圧力を認識してプロトタイプの入力として扱うことが可能である。本節ではそのような適用例について述べる。

6.3.1 圧力認識技術に関する研究

タッチパッドやタッチスクリーンにおけるタッチインタラクションの語彙を拡張するために、圧力検出可能な面を作成する様々な試みがある。Resenberg らの UnMousePad [RP09] は Force Sensitive Resistance [eve82] の原理を応用した、マルチタッチ感圧パッドである。 Rendel らの PyzoFlex [RGH+12] は圧電効果と焦電効果を組み合わせた、ホバーと圧力を検出可能な入力デバイスである。Resonant Pressure Sensor [Bla73] は、圧力を加える事によって自身の共振周波数が変化する微細構造による圧力センサである。Kim らの RetroDepth [KID+14] はステレオカメラと再帰性反射材を用いて高精度な 3D シルエットトラッキングを可能にしており、それを用いて変形する物体上におけるタッチの圧力を検出している。

これらの研究のうち、Resonant Pressure Sensor [Bla73] は本手法と同様のセンシング原理を用いている。しかしながら、これはセンサに組み込まれた機構の音響特性を測定することによりセンサ自体に加わる圧力を測定している。一方、本研究は既存物体の音響特性を計測し、機械学習を適用することによってその物体に加わる圧力を推定しようとする点でこの研究とは異なる。

6.3.2 適用例

手のオブジェ

ネックレスを掛けるために販売されていた手の形をしたアクリルのオブジェに対して本手法を適用することによって手の形をした握力計のプロトタイプを作成した。(図左)。オブジェを握手するように握る際の圧力変化を連続値学習を用いて学習させ、オブジェを握る力を検出できるようにした（図右)。

せん断力入力

携帯情報端末のタッチ入力の語彙拡張のためにせん断力入力機能を追加する研究がなされている [HL11, LLL+12]。本手法を用いてこれらの簡易プロトタイプを作成した。それに際して、まず図に示すようなスマートフォンケースを作成した。これは通常より少し大きめのプラスチックケースを用いており、端末を手に持った状態で画面にせん断力を加えるとケースが少し歪むようになっている。また、端末の持ち手がケースの音響特性に影響を与えないようにするためにケースの底面にはゴムシートを接着してある。この状態でケースの下部に圧

図 6.9: 手のオブジェ。

図 6.10: せん断力入力のためのケース。

図 6.11: せん断力入力の様子。

電センサを取り付け、せん断力を図の赤矢印方向に加える動作についての連続値学習を行った。これによってスマートフォンのせん断力入力インタフェースの使用感を確かめることができた。

6.4 ブロック玩具を用いたインタフェース

レゴブロックやデュプロブロックのようなブロック玩具は容易に様々な構造を構築することが可能であり、プロトタイピングによく使用される [Cra03]。

一方で、物体の音響特性はレゴのような構造物の構造の変化によっても生じることが分かっている。本研究ではこのことに着目し、既存のブロック玩具をプロトタイプの構成要素として用いながら、インタラクティブなプロトタイプを作成可能とするプロトタイピング手法を示す。これはブロック玩具を用いて構成したプロトタイプ（タッチパッド、レバー、ペダル等）に本研究で開発したプロトタイピングツールキットを取り付けることによって行う。

6.4.1 ブロック玩具を用いた研究

ブロック玩具に着目した研究はこれまでにいくつか提案されている。Gupta らの DuploTrack [GFCC12] は深度カメラを用いてユーザが組み立てているデュプロブロックの状態を認識し、プロトタイプの組み立てプロセスを提示している。Muller らの faBrickation [MMG+14] は 3D プリンタとブロックを組み合わせることにより、高速なプロトタイプ作成の支援を行っている。椎尾らの StudI/O [椎尾 14] は投影を用いてブロック玩具の組み立てを支援している。これらの研究はブロック玩具を用いたプロトタイプの組み立て支援を対象としているのに対し、本研究はブロック玩具を用いて構成したプロトタイプにインタラクティブ性を持たせることを主眼としている。

本手法と同様の目的で作られたツールキットとして、Mindstorms [Min] がある。Mindstorms はモーター、センサ、ギア、ローラー等の部品を用いてより高度かつインタラクティブな作品を作成可能である。本手法は Mindstorms のセンサのサブセット的な立ち位置として機能する。Mindstorms に含まれるセンサ（温度センサやタッチセンサ等）は基本的に単一の機能のみを有するが、本手法におけるセンサは学習の仕方によって異なる働きをするため、より汎用的に使用することができる。また、将来的には Mindstorms と組み合わせてより高度なプロトタイプを作成可能になるものと考えられる。

6.4.2 適用例

本手法の適用例を以下に示す。これらの適用例の作成に際して、レゴブロック版の圧電センサ（図）を使用した。

連続入力系

連続入力系の学習を用いて作成したプロトタイプを図に示す。

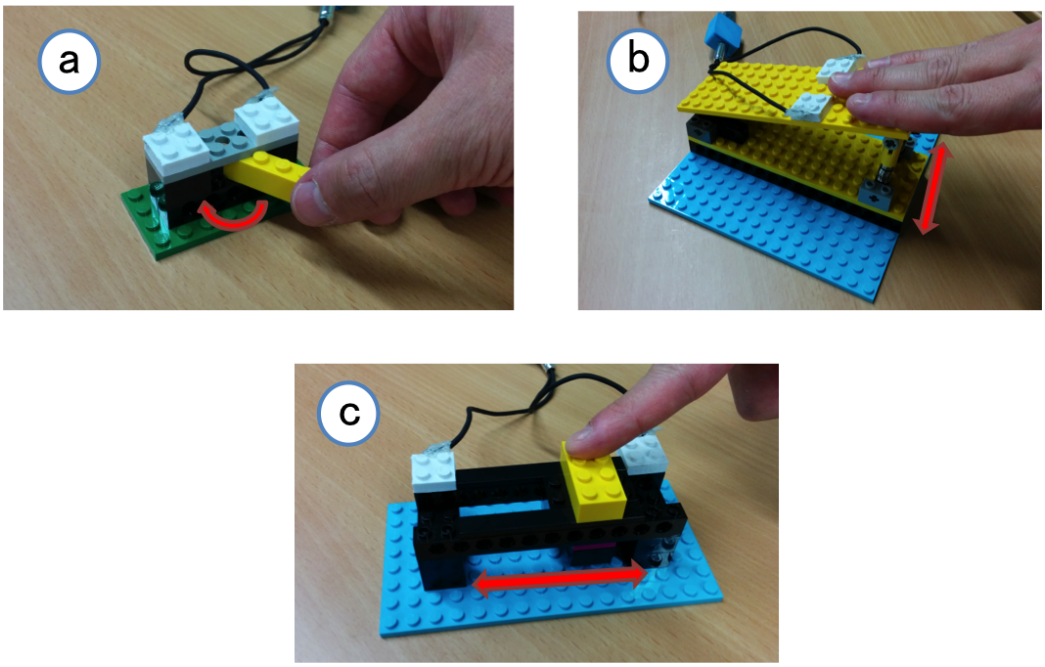

図 6.12: 連続入力系プロトタイプ：a) レバー、b) ペダル、c) スライダー。

6.4.3 連続値入力

連続値学習を適用した例を図に示す。

レバー

図 a のようにブロックを組み、黄色いブロックをレバーとして赤矢印方向に回転させる動作によって連続値学習を行い、レバーの回転角が認識されることを確かめた。本例はレーシングゲームやバランスゲームのハンドルコントローラに応用可能である。

ペダル

図 b のようにブロックを組み、ペダルを赤矢印方向に押し込む動作によって連続値学習を行い、ペダルの踏み具合が認識されることを確かめた。なお、このペダルにはバネが含まれており、ペダルを押し込まない時には自動的に元の高さに戻るようになっている。本例は、レーシングゲームのアクセルペダルや、Angry Birds のようなアクションゲームの入力インタフェースとして応用可能である。

スライダ

図 c のようにブロックを組み、黄色のブロックをつまみとして赤矢印方向にスライドさせる動作によって連続値学習を行い、スライダの位置が認識されることを確かめた。本例は音楽プレイヤのシークを行うための入力インタフェースとして応用可能である。

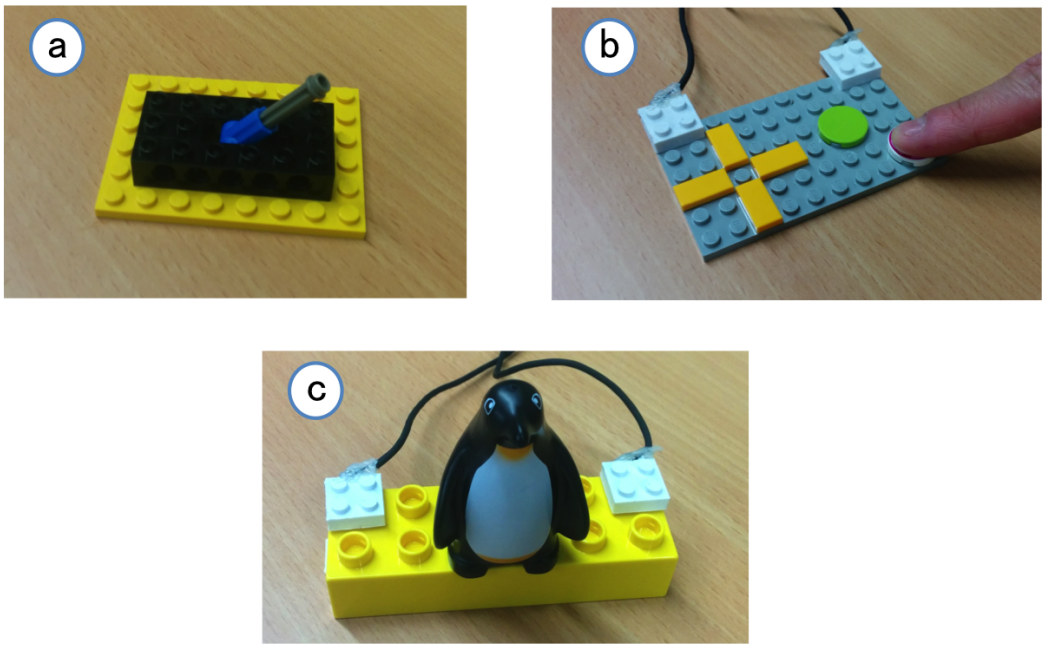

図 6.13: 離散入力系プロトタイプ：a) スイッチ、b) ゲームコントローラ、c) インタラクティブアニマル。

6.4.4 離散値入力

離散値入力の学習を適用した例を図に示す。

スイッチ

図 a のようにブロックを組み、スイッチを左右に倒した状態においてそれぞれ離散値学習を行い、2 状態の入力が可能なスイッチを作成した。

ゲームコントローラ

図 b のようにブロックを組み、4 つの黄色いタイルブロックと、2 つの丸いタイルブロックをそれぞれタッチした状態において離散値学習を行い、ゲームコントローラを作成した。4 つの黄色いタイルブロックを方向入力に、2 つの丸いタイルブロックをアクションボタンとして割り当てることによって、スーパーファミコンやゲームボーイのようなゲームのコントローラのプロトタイプを作成することが可能である。

インタラクティブアニマル

デュプロブロックと動物型のブロックを用いて、インタラクティブアニマルというプロトタイプを作成した（図 c）。これは黄色のベースブロック（図 a）に動物型のブロックが乗せられると、その種類を認識し、その動物に対応した効果音を再生する（図 b）。また、その状態で動物ブロックが触れられるとさらに異なる効果音を再生する（図 c）。図

のように、複数種類の動物型ブロックを識別することも可能であり、それぞれの動物に対応した異なる効果音が再生される。このプロトタイプは、以上の各状態における音響特性の離散値学習を行うことによって作成した。

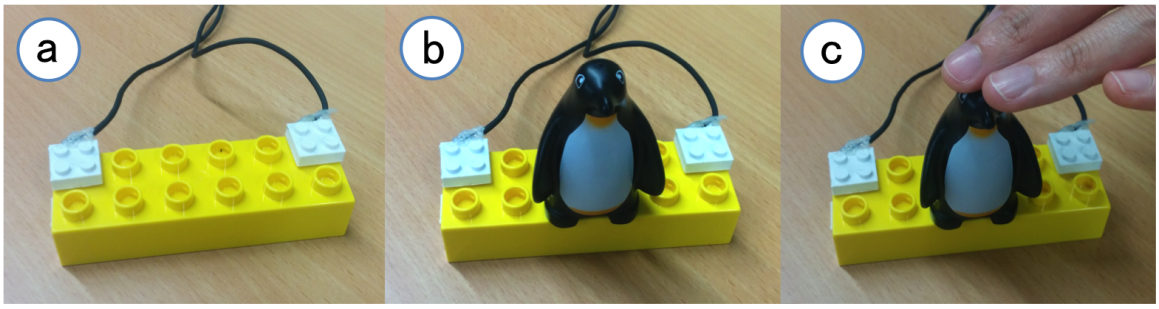

図 6.14: 動物型ブロックとタッチの認識。

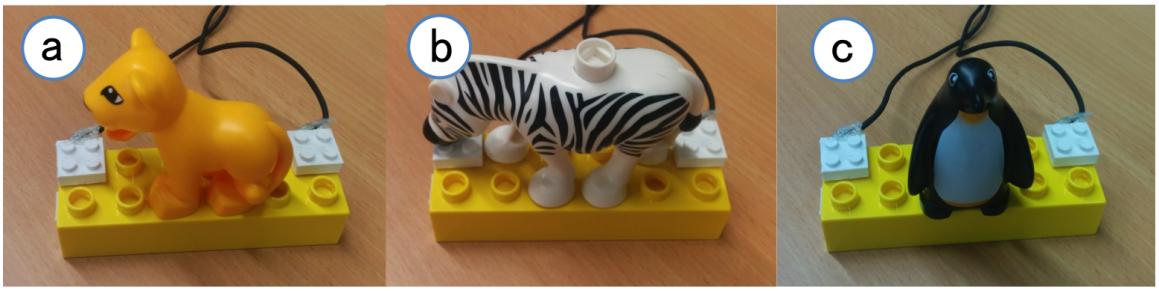

図 6.15: 動物型ブロックの識別。

第7章　評価実験

本章では本手法によって学習させた操作の認識性能に関する評価実験について述べる。

7.1　離散値入力学習に関する実験

離散値入力型の認識ソフトウェアを用いて学習させた操作に関して、操作の認識率と学習回数のトレードオフを調べる評価実験を行った。実験対象として、6.1.2 節で述べたプラスチック製の玩具における 5 種類（玩具に触れない状態も 1 種類とする）のタッチジェスチャ認識と、6.2.2 節で述べた携帯情報端末における 6 種類の把持状態認識を選択した。

7.1.1　被験者

この実験の被験者は 22 歳から 32 歳の 10 人（男性 7 人、女性 3 人）のボランティアである。すべての被験者はスマートフォンの利用者であり、1 週間から 3 年の利用歴があった。実験に際して使用した実験同意書を付録 A に添付する。

7.1.2　実験手順

実験では、各被験者が対象となる 2 つの適用例について独立に次のタスクを行う。

まず、各操作セット（すなわち、図の 5 種類のタッチジェスチャ（玩具に触れない状態も 1 種類とする）と図の 6 種類の把持姿勢）の中から 1 つの写真がランダムに選択され、被験者に 7 秒間提示され、被験者はそれに従って同様の操作を行う。操作を行う間、その姿勢を微動させるように指示した。これは、音響特性が物体に対する触れ方のわずかな変化に対して敏感に変化するため、それらの変動に対してロバストな識別結果を得るためである。最後の 1 秒間に共振スペクトルデータ（400 点の特徴ベクトル）を 40 個分取得する。写真が提示されてから 7 秒経過すると、その後異なる操作の写真が提示される。以上の手順はすべての操作の写真が提示されるまで繰り返される（以降これを 1 ラウンドとして扱う）。各被験者は 12 ラウンド分のジェスチャを行いう。これによって各被験者、各ジェスチャあたり 480 個のデータが得られることになる。

本実験からは、プラスチック製の玩具について、24000 個のデータ（10 被験者× 5 タッチジェスチャ× 12 ラウンド× 40 データ）を、携帯情報端末について、28800 個のデータ（10 被験者× 6 把持姿勢× 12 ラウンド× 40 データ）がそれぞれ得られた。

7.1.3 実験結果

以上の手順によって得られたデータを用いて、per-user classifier と general classifier からなる 2 種類の識別器の評価を行った。識別モデルの構築、評価には LIBSVM（RBF kernel, c=1.0, g=0.0025）を用いた。

Per-User Classifiers

各被験者のデータごとに識別モデルを構築し、個人利用環境で本手法を使用した際の認識率の評価を行った。各被験者のデータについて、11 ラウンド分のデータを訓練データとして利用し、残りの 1 ラウンド分のデータをテストデータとして認識率を求めた。これをすべてのラウンドの組み合わせについて行い（12 分割交差検定）、各被験者の結果から平均値を求めた。結果として、プラスチック製の玩具における 5 種類のタッチジェスチャの認識率は 99.6%（SD=0.36%）となった。一方で、携帯情報端末における 6 種類の把持姿勢の認識率は 86.3%（SD=4.2%）となった。

認識率と学習回数のトレードオフを調べるために、異なる学習回数で学習を行った際の認識率を求めた。そのために、各被験者のデータを 2,3,4,6 つのデータセットに分割し、それぞれのデータに対して 6,4,3,2 分割交差検定を行い、各被験者の結果から平均値を求めた（図）。結果として、プラスチック製の玩具については 1 ラウンド分の学習データにおいても 96.2%（SD=7.9%）の認識率が得られた。携帯情報端末については 1 ラウンド分の学習データにおいて 65.3%（SD=7.9%）の認識率にとどまった。また、5 ラウンド分の学習データを用いることにより、認識率は 82.3%（SD=3.0%）に改善した。

General User Classifiers

すべてのユーザのデータから識別モデルを構築し、複数人利用環境で本手法を使用した際の認識率の評価を行った。すべての被験者のデータから 9 人分の被験者のデータを訓練データとして使用し、残り 1 人分のデータをテストデータとして認識率を求めた。これをすべての被験者の組み合わせについて行い（10 分割交差検定）。結果としてプラスチック製の玩具における 5 種類のタッチジェスチャの認識率は 97.8%（SD=2.9%）となった。一方で、携帯情報端末における 6 種類の把持姿勢の認識率は 71.2%（SD=8.2%）にとどまった。

Per-User Classifiers と同様に認識率と学習回数のトレードオフ曲線を求めた（図）。結果として、プラスチック製の玩具は全被験者の 1 ラウンド分のデータを用いた学習において 96.0%（SD=3.8%）と高い認識率を保った。一方で携帯情報端末については 62.1%（SD=8.9%）の認識率にとどまった。また、全被験者の 5 ラウンド分のデータを用いた学習付近で認識率の改善は見られなくなった。

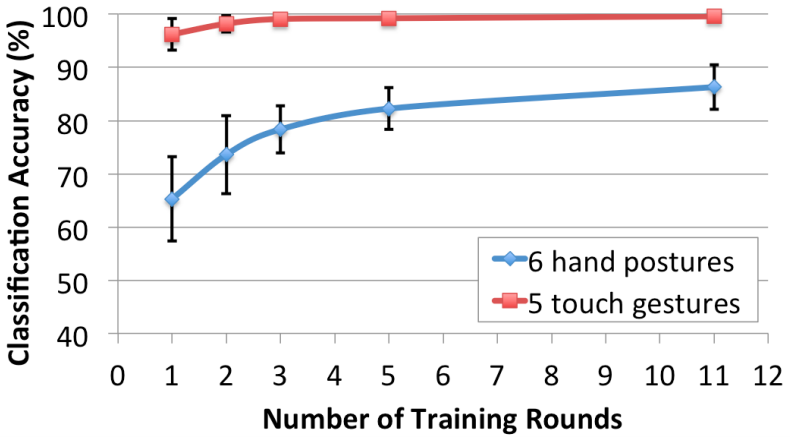

図 7.1: Per-user classifier における認識率。

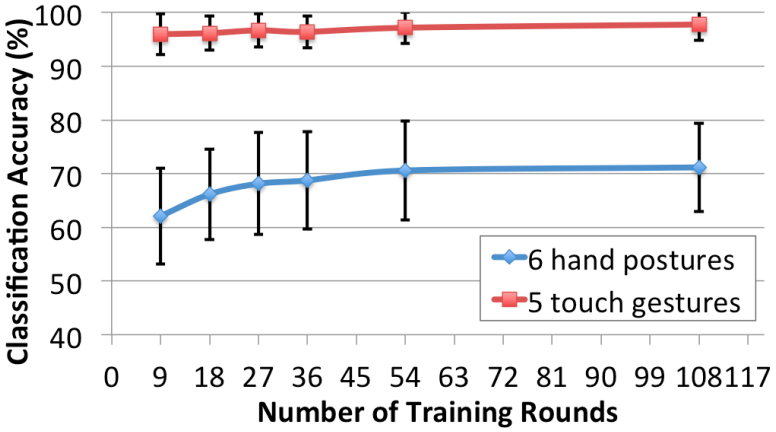

図 7.2: General classifier における認識率。

7.2 連続値入力学習に関する実験

連続値入力型の認識ソフトウェアを用いて学習させた操作に関して、音響特性から操作をどの操作正確に認識できるかを評価する実験を行った。この実験ではいくつかの物体をタッチする際の圧力変化の認識を評価対象とした。

実験に際して、圧力値の真値を用いて評価を行うために圧力センサを用いた実験用システムを構築した。このシステムを用いて構築した認識モデルの決定係数 r2 と二乗平均平方根誤差 RMSE（Root Mean Square Error）を求めることによって、認識モデルの妥当性を求めるものとした。

7.2.1 被験者

被験者は著者 1 名である。

7.2.2 圧力センサを用いた学習システム

圧力センサを用いた実験用システムのハードウェア構成を図に示す。このシステムでは、認識対象となる物体に取り付けるセンサとして、圧電センサの他に圧力センサ（Interlink FSR402）を取り付ける。圧力センサを取り付けた部位をタッチした際の圧力センサの値を教師ラベル、音響特性データを特徴量として SVR による機械学習を行う。圧力センサの値はシリアル通信によって、マイコン（mbed LPC1768）からコンピュータに取り込まれる。取り込む際の値は 100 段階とした。SVR には LIBSVM [CL11] による ε -SVR の実装を、デフォルトパラメータ（gamma=0.0025, C=1, ε =0.1）にて使用した。

図に示すような形状、素材でできた物体に本システムを適用し、次節に示す実験手順に従って実験を行った。

7.2.3 実験手順

まず、認識対象となる物体に対して、圧力センサを取り付ける。このとき、センサの位置は実験中に変更しないものとした。次に実験者が圧力センサの取り付けられた位置を 1 度タッチした状態で、様々な圧力をかけ続ける。この時に 10〜80 の値を示す圧力データにおける音響特性を、各圧力値につき 10 サンプルずつ取得する（各認識対象あたり合計 700 サンプル）。10〜80 という範囲を選んだ理由については、10 未満の値は圧力センサから得られるデータとして不安定であり、80 以上の圧力をかける入力を発生させることが困難であったためである。全てのデータを取得した後、10 分割交差検定にかけることにより、決定係数 r2 と二乗平方根誤差 RMSE を計算した。

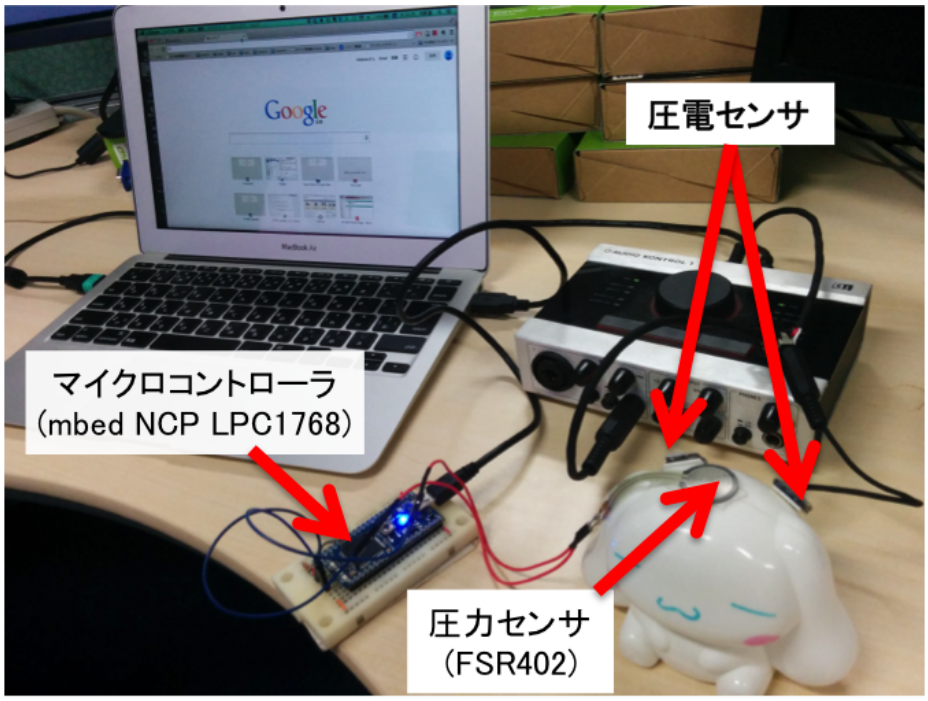

図 7.3: 圧力センサを用いた学習システムの構成。

7.2.4 認識対象

認識対象となる物体として図に示す物体（木の机、プラスチックの玩具、アクリルのオブジェ）を使用した。プラスチックの玩具については頭部分と右耳部分をタッチする場合についてそれぞれ評価を行った。

7.2.5 実験結果

各認識対象における交差検定の結果として、真値に対する推定値の平均をプロットした散布図とその回帰直線を図に示す。結果として、r2 は 0.833〜0.869、RMSE は 7.441〜8.704 の値を示した。認識モデルによる推定値と圧力センサによる真値は強い相関を示しており、誤差も ± 8%程度におさまることが分かった。また、今回は SVR のパラメータとしてデフォルトパラメータを用いているため、パラメータの最適化により、さらなる認識精度の向上も期待される。

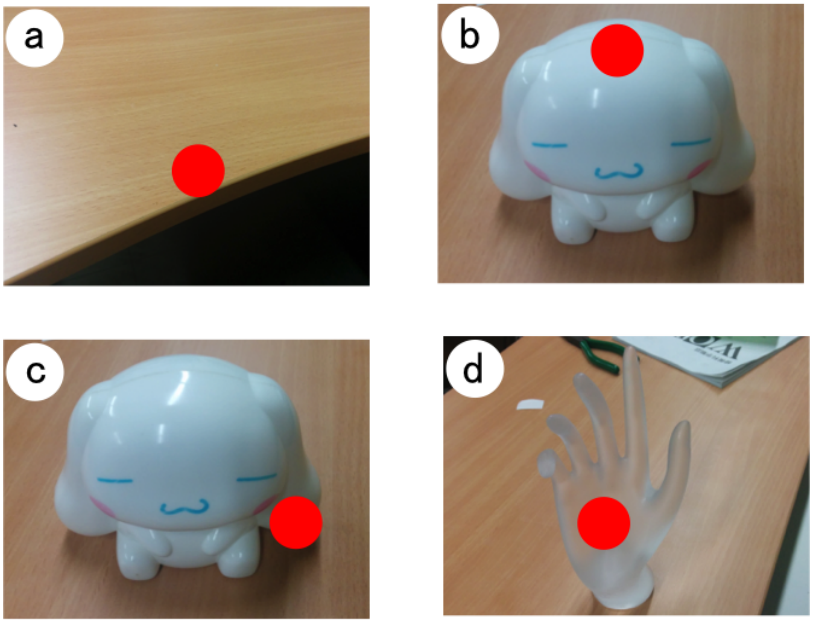

図 7.4: 認識対象：a) 木の机、b) プラスチックの玩具（頭）、c) プラスチックの玩具（右手）、d) アクリルのオブジェ。赤丸は圧力センサの取り付け位置および実験時にタッチした位置を示す。

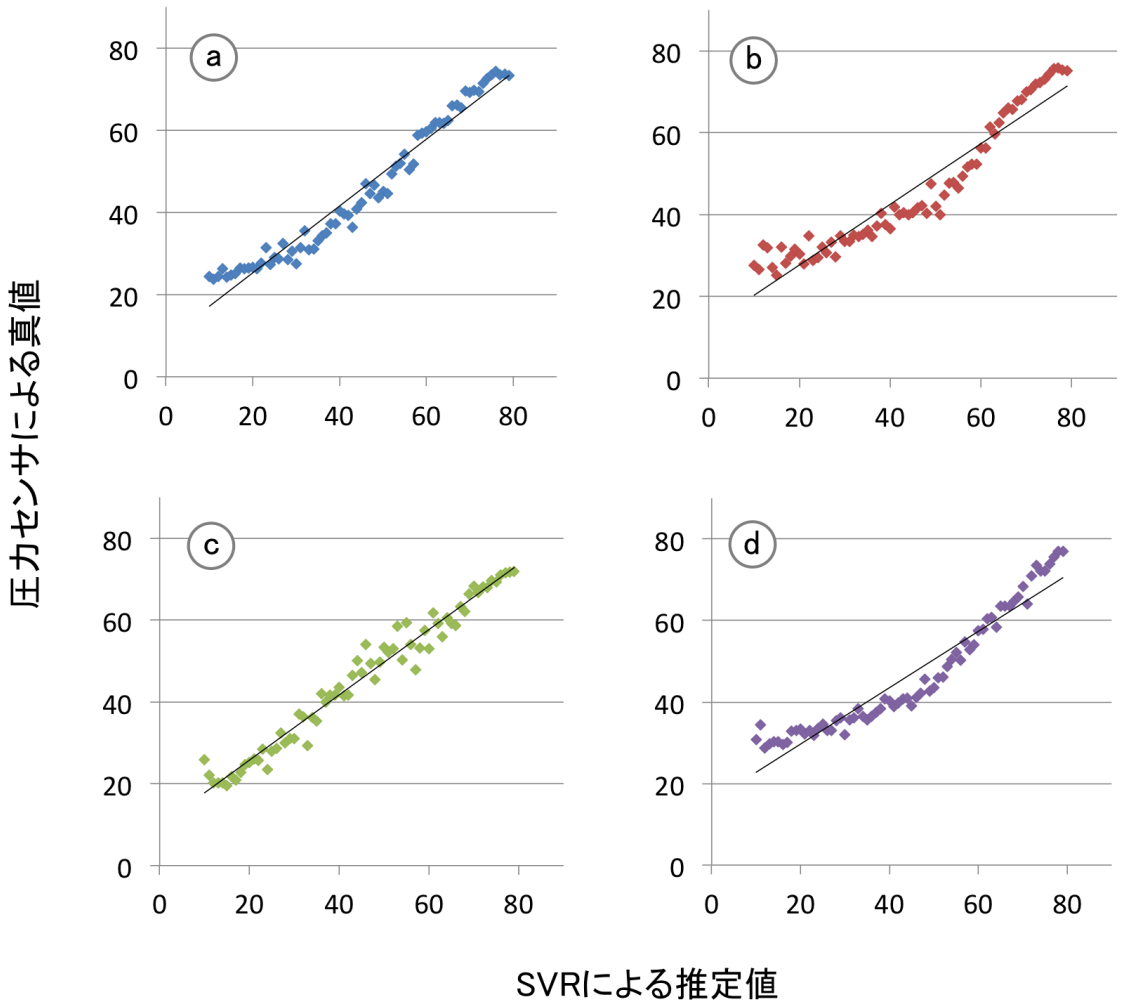

図 7.5: 評価結果：a) 木の机、b) プラスチックの玩具（頭）、c) プラスチックの玩具（右手）、d) アクリルのオブジェ。

第8章　議論

本章では本手法の制約、および今後の展望について述べる。

8.1　制約

これまでの章で示したように、本手法は既存物体に対する様々な操作を認識可能である。しかしながらいくつかの制約を持つ。

8.1.1　センサの取付位置

まず、センサの取付位置によって、物体の操作時に生じる音響特性の変化がよく現れる場合とそうでない場合がある。ただし、現在その判断基準が無いため、取付位置を決める際に経験則が求められる。例えば図 b のペダルの場合、水色の台座ブロックにセンサを取り付けても、認識に効果的な音響特性の変化は生じない。一方、実際に手や足が触れる黄色のブロックにセンサを取り付けた場合、ペダルの踏み具合と連動して音響特性に変化が生じる。これはペダルを押し込むことによって黄色のブロックにひずみが生じることによるものと思われる。今後はそのような事例を調査し、センサの取り付け位置に関する判断基準を明らかにする必要がある。

8.1.2　物体の構造

本手法の適用対象となる既存物体は単一構造物であることが望ましい。例えばスマートフォンは複数のパーツが組み合わさって構成されており、これらのパーツ間の結合部分によって振動の伝達が弱くなるおそれがある。実際にスマートフォンの画面にセンサを取り付けたとしても、背面のタッチを検出することは難しい。このような制約からスマートフォンの背面のタッチ等を検出する場合には背面にセンサを取り付けるかケースの内側にセンサを取り付ける必要がある。

8.1.3　学習の習熟

また、操作の認識精度は学習の良し悪しに依存する。離散値学習の場合、なるべく各操作についてまんべんなく学習させる必要がある。このことについては 7.1 節の実験で示したと

おりである。さらに、連続値学習の場合、認識ソフトウェア上に表示されるアニメーションガイドに従って、正確に一定速度で操作を行う必要がある。これにはユーザーの習熟が求められるため、今後はよりユーザにとって負担が少なくかつ効率的に学習を行える機械学習 UI を探る必要がある。

8.2 今後の展望

8.2.1 センサ取付方法の検討

現在の実装では、センサを物体に取り付ける方法として 2 種類の方法（両面テープ、レゴブロック）を採用している。両面テープを用いることにより、大抵の物体に取り付けることが可能であるが、その度に両面テープを新たにカットすることが煩わしい作業となる。レゴブロックはその点を解消しているが、適用対象がレゴブロックに限定される。この点についてより使いやすい取り付け方法について模索していく必要がある。例えば金属の物体に本手法を適用する場合にはセンサ部分に磁石を取り付ける方法が考えられる。また、両面テープについても、片面が固定強粘着剤、もう片面がウレシート粘着剤を採用しているもの等様々なタイプが存在する。これらの方法についても今後試していく必要がある。

8.2.2 ツールキットとしての使いやすさの評価

本研究では、評価実験として本手法が採用しているセンシング原理とその認識精度に関する実験を行った。一方で、ユーザにとっての本手法の使いやすさを測る評価実験はまだ行われていない。具体的には、ユーザが思い通りのプロトタイプを作成できるのか、どのくらいの速さでプロトタイプを作成できるのか等の指標に基いて評価を行う必要がある。今後はそのような実験を行い、ツールキットとして改善できる点を探し改良を行いたい。

8.2.3 3D プリンタで出力された物体への応用

近年 3D プリンタの普及により、様々な物体を容易に出力することが可能となってきた。3D プリンタを用いて物体を出力する場合、ABS や PLA 等のプラスチック素材が主な素材として使用される。また、3D プリンタによる出力物は単一構造物であり、机のような大きな物体の出力は困難である。これらのことから、3D プリンタによる出力物は本手法の適用対象としての条件を満たすことになる。既存物体を用いてプロトタイピングを行う際には、素材や大きさの制約を満たした条件の物体を探す必要がある一方で、3D プリンタによって出力された物体へ適用する際には、それらの制約について検討する必要が無くなる。今後 3D プリンタがより一般的に普及するのであれば、3D プリンタに本手法を適用することに特化した発展を考えることは有用となる。今後はそれらの方向についても本研究を展開していきたい。

第9章　結論

本研究では、物体の音響特性の性質を利用することにより、既存物体を用いてフィジカルなインタフェースのプロトタイピングを行う手法を提案し、それに基づいたプロトタイピングツールキットの実装を行った。物体の音響特性は、その物体に対して触れる、把持する、圧力を加える、構造を変化させることによって変化する性質がある。本手法では能動的音響計測を用いて既存物体の音響特性取得し、コンピュータ上で機械学習を適用することによってその音響特性から既存物体に対する操作を認識する。これにより、ユーザは自分の使いたい操作を既存物体に対して実演することによって、フィジカルなインタフェースを容易に構築することが可能である。本手法ではプロトタイプに対する離散的な操作および、連続的な操作を認識するために機械学習のアルゴリズムとして SVC および SVR をそれぞれ用いた。

本研究では、本手法によって作成可能なインタフェースのプロトタイプを 4 種類の分類に基いて作成することにより、本手法の適用可能性について述べた。また、評価実験として、本手法を用いて学習させた操作の認識精度に関する実験を行った。離散的な操作の認識に関する実験を行い、5 種類のタッチ操作の認識率が per-user test において 99.6%、general-user test において 97.8%となった。また、携帯情報端末における 6 種類の把持操作の認識率は、それぞれ 86.3%、71.2%となった。連続的な操作の認識に関する実験を行い、タッチの圧力による操作について、圧力センサによる真値と本手法による推定値間のデータを解析したところ、決定係数 r2 は 0.833〜0.869、2 乗平均平方根誤差 RMSE は 7.441〜8.704 となった。これらの結果から、音響特性を用いて既存物体に対する操作を認識することは可能であり、新たなプロトタイピング技術としての応用可能性を示した。

謝辞

本研究を進めるにあたり、志築文太郎先生、田中二郎先生、三末和男先生、高橋伸先生、嵯峨智先生には多くのご指導およびご助言をいただきました。特に志築先生には日々の研究生活の中で、論文の執筆、研究の進め方、物品の購入等多くの事について手厚いサポートをしていただきました。トップカンファレンスのベストペーパーを受賞するという大変貴重な経験ができたのも、志築先生にご指導いただいたおかげだと感じております。また、研究アイディアの着想段階において素早く必要な物品を購入していただけたため、研究を円滑に進めることが出来ました。本当にありがとうございました。

また、本研究の一部は IPA の 2013 年度未踏 IT 人材発掘・育成事業による助成を受けて行われました。IPA、PM の皆様には多くのご指導とご助言をいただきました。特に私のプロジェクトの担当 PM の後藤真孝先生にはプロジェクトの内容だけでなく、今後の人生についてまでも大変熱のこもったご指導をいただきました。同期メンバーのプロジェクトや取り組みの姿勢にも多くの刺激を受けました。また、近藤秀和様にはプロジェクトをビジネス化する上で大変多くのご指導およびご助言をいただきました。萬成亮太君にはプロジェクトの Web サイトを作成していただきました。そのおかげで様々な人から連絡をいただき、多くの貴重な機会が得られました。ありがとうございました。

本研究のハードウェアを開発する上で、大変多くの方にお世話になりました。本大学の庄野和宏先生には、他研究室の所属であるにもかかわらず電気回路の設計について大変多くのご助言を頂きました。無事にハードウェアを完成させることができたのも庄野先生のおかげだと感じております。また、橋場康人君にはハードウェアの筐体を作成するにあたり、多大なるご協力をいただきました。ありがとうございました。

研究室のメンバー、特に WAVE チームのメンバーには、研究についてのアドバイスだけでなく、研究以外の生活の中でも大変お世話になりました。ここまで充実した研究室生活を過ごすことができたのは彼らのおかげだと感じております。ありがとうございました。

最後に金銭面や精神面で支えてくれた家族や友人に深く感謝いたします。

参考文献

[AHT02] Brian Amento, Will Hill, and Loren Terveen. The sound of one hand: A wrist-mounted bio-acoustic fingertip gesture interface. In *CHI '02 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '02, pp. 724–725, New York, NY, USA, 2002. ACM.

[ard] arduino. http://www.arduino.cc/ . (accessed 2014.12.17).

[AZ12] Shiri Azenkot and Shumin Zhai. Touch behavior with different postures on soft smartphone keyboards. In *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services*, pp. 251–260. ACM, 2012.

[Bde09] Ayah Bdeir. Electronics as material: littleBits. In *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction*, pp. 397–400. ACM, 2009.

[BF87] Michael C Brenner and James J Fitzgibbon. Surface acoustic wave touch panel system, February 17 1987. US Patent 4,644,100.

[Bla73] William Carroll Blanchard. Resonant pressure sensor, 1973. US Patent 3,745,384.

[CHLC12] Lung-Pan Cheng, Fang-I Hsiao, Yen-Ting Liu, and Mike Y. Chen. iRotate grasp: Automatic screen rotation based on grasp of mobile devices. In *Adjunct proceedings of the 25th annual ACM symposium on User interface software and technology*, UIST Adjunct Proceedings '12, pp. 15–16, New York, NY, USA, 2012. ACM.

[Chr03] D'Ambrose Christopher. Frequency Range of Human Hearing. In *The Physics Factbook*. 2003.

[CL11] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, Vol. 2, No. 3, pp. 27:1–27:27, 2011.

[Col09] Tim Collins. Active acoustic touch interface. *Electronics Letters*, Vol. 45, No. 20, pp. 1055 –1056, 24 2009.

[Cra03] Lawrence Cralg. Right–Rapid–Rough. *Ask Magazine*, No. 13, pp. 12–15, 2003.

[eve82] Electronic pressure sensitive transducer apparatus, 1982. US Patent 4,314,227.

[FT12] Adam Fourney and Michael Terry. PICL: Portable in-circuit learner. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, UIST '12, pp. 569–578, New York, NY, USA, 2012. ACM.

[GF01] Saul Greenberg and Chester Fitchett. Phidgets: Easy development of physical interfaces through physical widgets. In *Proceedings of the 14th annual ACM symposium on User interface software and technology*, UIST '01, pp. 209–218, New York, NY, USA, 2001. ACM.

[GFCC12] Ankit Gupta, Dieter Fox, Brian Curless, and Michael Cohen. DuploTrack: A real-time system for authoring and guiding duplo block assembly. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, pp. 389–402. ACM, 2012.

[GMPT12] Sidhant Gupta, Daniel Morris, Shwetak Patel, and Desney Tan. SoundWave: Using the Doppler effect to sense gestures. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, CHI '12, pp. 1911–1914, New York, NY, USA, 2012. ACM.

[GWP12] Mayank Goel, Jacob Wobbrock, and Shwetak Patel. GripSense: Using built-in sensors to detect hand posture and pressure on commodity mobile phones. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, UIST '12, pp. 545–554, New York, NY, USA, 2012. ACM.

[HAMK07] Björn Hartmann, Leith Abdulla, Manas Mittal, and Scott R. Klemmer. Authoring sensor-based interactions by demonstration with direct manipulation and pattern recognition. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pp. 145–154, New York, NY, USA, 2007. ACM.

[Han05] Jefferson Y Han. Low-cost multi-touch sensing through frustrated total internal reflection. In *Proceedings of the 18th annual ACM symposium on User interface software and technology*, pp. 115–118. ACM, 2005.

[HBW11] Chris Harrison, Hrvoje Benko, and Andrew D. Wilson. OmniTouch: Wearable multitouch interaction everywhere. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, UIST '11, pp. 441–450, New York, NY, USA, 2011. ACM.

[HBW12] Sungjae Hwang, Andrea Bianchi, and Kwangyun Wohn. MicPen: Pressure-sensitive pen interaction using microphone with standard touchscreen. In *Proceedings of the 2012 ACM annual conference extended abstracts on Human Factors in Computing*

Systems Extended Abstracts, CHI EA '12, pp. 1847–1852, New York, NY, USA, 2012. ACM.

[HH08] Chris Harrison and Scott E. Hudson. Scratch input: Creating large, inexpensive, unpowered and mobile finger input surfaces. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*, UIST '08, pp. 205–208, New York, NY, USA, 2008. ACM.

[HKB+06] Björn Hartmann, Scott R. Klemmer, Michael Bernstein, Leith Abdulla, Brandon Burr, Avi Robinson-Mosher, and Jennifer Gee. Reflective physical prototyping through integrated design, test, and analysis. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*, UIST '06, pp. 299–308, New York, NY, USA, 2006. ACM.

[HL11] Seongkook Heo and Geehyuk Lee. Force Gestures: Augmenting touch screen gestures with normal and tangential forces. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, pp. 621–626. ACM, 2011.

[HM06] Scott E Hudson and Jennifer Mankoff. Rapid construction of functioning physical interfaces from cardboard, thumbtacks, tin foil and masking tape. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*, pp. 289–298. ACM, 2006.

[HSH11] Chris Harrison, Julia Schwarz, and Scott E. Hudson. TapSense: Enhancing finger interaction on touch surfaces. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, UIST '11, pp. 627–636, New York, NY, USA, 2011. ACM.

[HSP12] Chris Harrison, Munehiko Sato, and Ivan Poupyrev. Capacitive fingerprinting: Exploring user differentiation by sensing electrical properties of the human body. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, pp. 537–544. ACM, 2012.

[HTM10] Chris Harrison, Desney Tan, and Dan Morris. Skinput: Appropriating the body as an input surface. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pp. 453–462, New York, NY, USA, 2010. ACM.

[HXH12] Chris Harrison, Robert Xiao, and Scott Hudson. Acoustic barcodes: Passive, durable and inexpensive notched identification tags. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, UIST '12, pp. 563–568, New York, NY, USA, 2012. ACM.

[KCC+06] Kee-Eung Kim, Wook Chang, Sung-Jung Cho, Junghyun Shim, Hyunjeong Lee, Joonah Park, Youngbeom Lee, and Sangryong Kim. Hand grip pattern recognition for mobile user interfaces. In *Proceedings of the 18th conference on Innovative applications of artificial intelligence - Volume 2*, IAAI'06, pp. 1789–1794. AAAI Press, 2006.

[KHC+13] Yoshihiro Kawahara, Steve Hodges, Benjamin S Cook, Cheng Zhang, and Gregory D Abowd. Instant inkjet circuits: Lab-based inkjet printing to support rapid prototyping of ubicomp devices. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pp. 363–372. ACM, 2013.

[KID+14] David Kim, Shahram Izadi, Jakub Dostal, Christoph Rhemann, Cem Keskin, Christopher Zach, Jamie Shotton, Timothy Large, Steven Bathiche, Matthias Nießner, et al. RetroDepth: 3D silhouette sensing for high-precision input on and above physical surfaces. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pp. 1377–1386. ACM, 2014.

[KR01] Alok Kataria and David W Rosen. Building around inserts: Methods for fabricating complex devices in stereolithography. *Rapid Prototyping Journal*, Vol. 7, No. 5, pp. 253–262, 2001.

[LG12] Shenwei Liu and François Guimbretière. FlexAura: A flexible near-surface range sensor. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, UIST '12, pp. 327–330, New York, NY, USA, 2012. ACM.

[LJJ11] Pedro Lopes, Ricardo Jota, and Joaquim A. Jorge. Augmenting touch interaction through acoustic sensing. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS '11, pp. 53–56, New York, NY, USA, 2011. ACM.

[LLL+12] Bhoram Lee, Hyunjeong Lee, Soo-Chul Lim, Hyungkew Lee, Seungju Han, and Joonah Park. Evaluation of human tangential force input performance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 3121–3130. ACM, 2012.

[LSFI13] Suzanne Low, Yuta Sugiura, Kevin Fan, and Masahiko Inami. Cuddly: Enchant your soft objects with a mobile phone. In *Advances in Computer Entertainment*, pp. 138–151. Springer, 2013.

[mbe] mbed. https://mbed.org/ . (accessed 2014.12.17).

[mes] Sony, MESH. http://www.gizmodo.jp/2014/05/diymesh.html. (accessed 2014.12.17).

[Min] LEGO, Mindstorms. http://mindstorms.lego.com/ . (accessed 2014.12.17).

[MMG+14] Stefanie Mueller, Tobias Mohr, Kerstin Guenther, Johannes Frohnhofen, and Patrick Baudisch. faBrickation: Fast 3D printing of functional objects by integrating construction kit building blocks. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pp. 3827–3834. ACM, 2014.

[MSWHQ08] Roderick Murray-Smith, John Williamson, Stephen Hughes, and Torben Quaade. Stane: Synthesized surfaces for tactile input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pp. 1299–1302, New York, NY, USA, 2008. ACM.

[PJY+07] Duc Truong Pham, Ze Ji, Ming Yang, Zuobin Wang, and Mostafa Al-Kutubi. A novel human-computer interface based on passive acoustic localisation. In *Human-Computer Interaction. Interaction Platforms and Techniques*, pp. 901–909. Springer, 2007.

[PLCH02] Joseph A Paradiso, Che King Leo, Nisha Checka, and Kaijen Hsiao. Passive acoustic knock tracking for interactive windows. In *CHI'02 Extended Abstracts on Human Factors in Computing Systems*, pp. 732–733. ACM, 2002.

[PSLS12] Ivan Poupyrev, Philipp Schoessler, Jonas Loh, and Munehiko Sato. Botanicus interacticus: interactive plants technology. In *ACM SIGGRAPH 2012 Emerging Technologies*, p. 4. ACM, 2012.

[RGH+12] Christian Rendl, Patrick Greindl, Michael Haller, Martin Zirkl, Barbara Stadlober, and Paul Hartmann. PyzoFlex: Printed piezoelectric pressure sensing foil. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, pp. 509–518. ACM, 2012.

[RMMH+09] Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, et al. Scratch: Programming for all. *Communications of the ACM*, Vol. 52, No. 11, pp. 60–67, 2009.

[RP09] Ilya Rosenberg and Ken Perlin. The UnMousePad: An interpolating multi-touch force-sensing input pad. In *ACM Transactions on Graphics*, Vol. 28, pp. 65:1–65:9. ACM, 2009.

[RS00] Jun Rekimoto and Eduardo Sciammarella. ToolStone: Effective use of the physical manipulation vocabularies of input devices. In *Proceedings of the 13th annual ACM symposium on User interface software and technology*, UIST '00, pp. 109–117, New York, NY, USA, 2000. ACM.

[SBG+11] Hyunyoung Song, Hrvoje Benko, François Guimbretière, Shahram Izadi, Xiang Cao, and Ken Hinckley. Grips and gestures on a multi-touch pen. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, CHI '11, pp. 1323–1332, New York, NY, USA, 2011. ACM.

[SKW+11] Yuta Sugiura, Gota Kakehi, Anusha Withana, Calista Lee, Daisuke Sakamoto, Maki Sugimoto, Masahiko Inami, and Takeo Igarashi. Detecting shape deformation of soft objects using directional photoreflectivity measurement. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pp. 509–516. ACM, 2011.

[SLO+12] Yuta Sugiura, Calista Lee, Masayasu Ogata, Anusha Withana, Yasutoshi Makino, Daisuke Sakamoto, Masahiko Inami, and Takeo Igarashi. Pinoky: A ring that animates your plush toys. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 725–734. ACM, 2012.

[SPH12] Munehiko Sato, Ivan Poupyrev, and Chris Harrison. Touché: Enhancing touch interaction on humans, screens, liquids, and everyday objects. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, CHI '12, pp. 483–492, New York, NY, USA, 2012. ACM.

[SR99] Brian J Schwarz and Mark H Richardson. Experimental modal analysis. *CSI Reliability week*, 1999.

[SRS12] Jay Silver, Eric Rosenbaum, and David Shaw. Makey Makey: Improvising tangible and nature-based user interfaces. In *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction*, pp. 367–370. ACM, 2012.

[SW59] JL Stewart and E Westerfield. A theory of active sonar detection. *Proceedings of the IRE*, Vol. 47, No. 5, pp. 872–881, 1959.

[SZH12] Valkyrie Savage, Xiaohan Zhang, and Björn Hartmann. Midas: Fabricating custom capacitive touch sensors to prototype interactive objects. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, pp. 579–588. ACM, 2012.

[TB09] Brandon T. Taylor and V. Michael Bove, Jr. Graspables: Grasp-recognition as a user interface. In *Proceedings of the 27th international conference on Human factors in computing systems*, CHI '09, pp. 917–926, New York, NY, USA, 2009. ACM.

[TITO11] Kentaro Takemura, Akihiro Ito, Jun Takamatsu, and Tsukasa Ogasawara. Active bone-conducted sound sensing for wearable interfaces. In *Proceedings of the 24th*

annual ACM symposium adjunct on User interface software and technology, UIST '11 Adjunct, pp. 53–54, New York, NY, USA, 2011. ACM.

[VC11] Daniel Vogel and Géry Casiez. Conté: Multimodal input inspired by an artist's crayon. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pp. 357–366. ACM, 2011.

[VSH+12] Nicolas Villar, James Scott, Steve Hodges, Kerry Hammil, and Colin Miller. . NET Gadgeteer: A platform for custom devices. *Pervasive Computing*, pp. 216–233, 2012.

[WB09] Raphael Wimmer and Sebastian Boring. HandSense: Discriminating different ways of grasping and holding a tangible user interface. In *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction*, TEI '09, pp. 359–362, New York, NY, USA, 2009. ACM.

[WBHP12] Karl Willis, Eric Brockmeyer, Scott Hudson, and Ivan Poupyrev. Printed Optics: 3D printing of embedded optical elements for interactive devices. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, UIST '12, pp. 589–598, New York, NY, USA, 2012. ACM.

[Wim10] Raphael Wimmer. FlyEye: Grasp-sensitive surfaces using optical fiber. In *Proceedings of the fourth international conference on Tangible, embedded, and embodied interaction*, TEI '10, pp. 245–248, New York, NY, USA, 2010. ACM.

[宮田 13] 宮田章裕, 有賀玲子, 宮下広夢, 佐藤隆, 井原雅行, 小林透. 貼りつけるだけで家具をインテリジェント化するデバイス. インタラクション 2013, pp. 796–801. 情報処理学会, 2013.

[椎尾 14] 椎尾一郎, 橋本菜摘. StudI/O: 立体への投影を利用したトイブロック組み立て支援. WISS'14: 第 22 回インタラクティブシステムとソフトウェアに関するワークショップ, pp. 37–42. ソフトウェア科学会, 2014.

付 録A　評価実験に使用した書類

7.1 節における実験の際に使用した実験同意書を付録として以下に示す。

触れ方データ収集のお願い

本調査の目的は、大野の研究を行うにあたり、触れ方認識のためのデータを収集することです。本調査では 2 種類の物体（玩具と携帯端末）に対して以下のタスクを行なっていただきます。

実験タスク

1. 各物体を特定の触れ方（持ち方）で触れた写真が 7 秒間画面に表示されるので、その写真に合わせて物体に触れる、もしくは物体を持ってください。写真の触れ方に厳密に合わせる必要はありませんので、自然に行なってください。
2. 物体に触れたら、次の写真に切り替わるまで手を少し動かしたり、物体を傾けたりしてください。このとき動きにむらがでないように注意してください。
3. 画面に写真が表示されてから 7 秒経過すると異なる写真に切り替わりますので，同様に触れてください。

以降 1〜3 の繰り返しとなります。

本調査において収集する触れ方データは玩具について 5 種類、携帯端末について 6 種類になります。また、同一の触れ方を 12 セット行います。各タスク終了までの所要時間は 7〜9 分となります。

データの取扱いに関して

本調査において得られたデータは、分析、論文、学内外の発表において使用する可能性があります。また実験の様子を撮影することがあります。その際は個人が特定できないように処理します。

実験に関して、上記内容を理解、同意した上で実験に参加して頂ける場合は、下部の署名欄に署名をお願いします。

平成　　年　　月　　日

所属　　署名

説明者 : 所属　システム情報工学研究科　CS 専攻　署名