

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

磁石を利用した携帯端末周辺空間における
3次元ジェスチャーインタフェースの設計と実装

PHAM THANH SON

修士（工学）

（コンピュータサイエンス専攻）

指導教員 高橋伸

2014年3月

概要

携帯端末のコンパクトなサイズを保ちながら、入力領域を拡張するために、携帯端末のセンサ（カメラや、近接センサや、磁気センサなど）を利用し、ユーザの手の動きを認識し、入力を行うという携帯端末周辺空間の入力インタフェースの研究は多く注目されている。本研究は携帯端末周辺空間の入力インタフェースの一つの手法として、磁石を利用したジェスチャーインタフェースを提案する。本研究のシステムでは、ユーザは磁石を指につけ、指の動きでジェスチャーを行い、タッチの入力インタフェースを補足し、素早い操作ができるインタフェースである。また、本研究では磁石の位置と角度を利用して3種類のジェスチャー：相対位置ジェスチャー、繰り返しジェスチャー、図形ジェスチャーを提案した。ジェスチャーの認識の実験を行い、実験の結果に基づいて、それらのジェスチャーを利用する描画アプリケーションも実装した。磁石トラッキングのためには、携帯端末の裏に3次元磁気センサの配列を装着し、線形な手法で磁石の位置と角度を計算する。そのため、携帯端末の処理能力でもリアルタイムの入力ができるようになった。以前の携帯端末ジェスチャー入力インタフェースと比べ、タッチ入力に邪魔せず、学習過程が必要としなく、また多くの種類のジェスチャーに対応できる入力インタフェースである。

目次

第1章 序論	1
1.1 携帯端末周辺空間インタラクション	1
1.2 ジェスチャーのインタラクション	2
1.3 本研究の目的とアプローチ	2
1.4 本論文の構成	2
第2章 関連研究	4
2.1 携帯端末のジェスチャーインタラクションの研究	4
2.2 磁気センシングを利用する研究	5
2.3 研究の位置づけ	5
第3章 磁石の位置と角度の取得アルゴリズム	6
3.1 計算モデル	6
3.2 3次元位置情報の取得	7
3.3 磁石角度を計算するアルゴリズム	9
3.4 計算アルゴリズムの利点と欠点	9
第4章 システム実装	11
4.1 ハードウェアの実装	11
4.1.1 磁気センサ配列	11
4.1.2 携帯端末と通信	12
4.2 ソフトウェアの実装	13
4.2.1 通信モジュール	13
4.2.2 磁石の位置と角度の計算の実装	13
位置の計算	14
角度の計算	14
第5章 認識ジェスチャーの検討	16
5.1 相対位置ジェスチャー	16
5.2 繰り返しジェスチャー	16
5.3 図形ジェスチャー	17
第6章 図形ジェスチャー認識の実験	20

6.1	図形ジェスチャーの認識アルゴリズム	20
6.2	実験の内容	21
6.3	実験の流れ	21
6.4	実験の結果	22
6.5	検討	23
第 7 章	磁石入力ジェスチャーを応用した描画アプリ	25
7.1	アプリの GUI	25
7.2	操作の流れ	25
7.3	操作とジェスチャーの割当	25
7.3.1	図形入力モードの操作	25
7.3.2	編集モードの操作	26
7.3.3	色選択モードの操作	26
7.3.4	undo-redo モードの操作	27
第 8 章	今後の課題と発展	29
	謝辞	30
	参考文献	31
付 録 A	基盤の設計	34

図目次

3.1	磁気センサの座標	7
4.1	システム構成	11
4.2	磁気センサの基盤	12
4.3	携帯の後ろに取り付けた磁石の基盤	13
4.4	データ受信モジュール	14
5.1	携帯端末の裏からスワイプ操作でナビゲーションできる	17
5.2	選択後の移動方向により、オブジェクトをその仮想スペースにコピーする . .	17
5.3	タッチ・ジェスチャー・タッチの入力	18
5.4	タッチ・ジェスチャー・タッチの入力	19
6.1	実験	21
6.2	実験の画面	22
6.3	6 人の被験者の正解率	23
6.4	ジェスチャー別の正解率	23
6.5	被験者が提案した図形	24
7.1	アプリの画面	26
A.1	基盤実装図	34
A.2	センサの回路	35

第1章 序論

1.1 携帯端末周辺空間インタラクション

Hinckley らがはじめて携帯端末に加速度センサ、近接センサなどを追加して、周辺環境やコンテキストのセンシング能力をさらに豊富にするというアイデア [1] を提案してから現在まで、たった 13 年で携帯端末は単なるボタンを押して入力するデバイスから進化して来た。最新の携帯端末はカメラ、加速度センサ、デジタルコンパスを内蔵することは基本とし、さらに圧力センサや、輝度センサや、温度センサなどを内蔵する機種も存在し、大幅なセンシングの能力を持つようになっている。

またコンテキストをセンシングだけではなく、それらのセンサのデータも入力としても利用されるようになってきた。加速度センサ、磁気センサのデータを利用することでユーザは携帯端末を動かす入力にはゲームなどに応用することも多くなった。カメラを利用して画像認識技術でマークや実物のものを認識し、入力に利用することも多く利用されている。この流れで、Butler らはセンサを利用して、携帯周辺空間に行われるユーザの手の動きを検出し、入力に変換する「携帯端末周辺空間インタラクション」(Around Device Interaction) という概念を導入した [2]。彼らは携帯端末での制限される入力領域の問題とタッチ画面のオクルージョンの問題に着目し、それらの問題を解決するために入力領域を携帯端末の 3 次元周辺空間に拡張するというアイデアを挙げた。例として彼らは IR 距離センサを利用して、ユーザの手のひらの 3 方向の動きを認識する SideSight システムを実装した。

その次に様々なセンサを利用して「携帯端末周辺空間インタラクション」を実装する研究は多く出てきた。例として PalmSpace[3]、HoverFlow[4] は同じく IR センサを利用し、Abracadabra[5] は磁気センサ、SoundWave[6] は超音波センサを利用していた。その中で Harrison らの Abracadabra[5] システムは磁気センサを利用し、磁石の位置を検出し携帯周辺空間の精密なインタラクションをでき、様々な可能性、拡張性を持つシステムであった。しかし彼らは磁石の 2 次元位置しか検出し、利用できるインタラクションは束縛された。

本研究も磁石でのインタラクションに着目し、磁石の位置情報を利用して、新しい「携帯端末周辺空間インタラクション」を検討する。理論上には磁気センサのデータから磁石の 3 次元の位置と角度を検出することはできる。磁石の 3 次元の位置と角度を検出できるとすると、磁石を指先につけ、指先の些細な動きでも検出できるようになり、様々なインタラクションが可能になると考えられる。

一方、現在携帯端末でよく利用されるタッチ入力ではユーザが画面に直接入力することができ、直感的に画面上の対象とインタラクションできる。磁石の 3 次元入力はタッチ入力のアフォーダンスを完全に入れ替えることは難しいが、タッチ入力と合わせて新たな一連のイ

インタラクションに拡張することも考えられる。タッチ画面で詳細な入力を行い、磁石の入力でモードの切り替えや、コントロールの操作などを行い、タッチ入力を補足すると携帯端末で素早く入力できることが可能になる。そのため、本研究は磁石の位置と角度のデータを利用して、ジェスチャーの入力として認識することに注目する。

1.2 ジェスチャーのインタラクション

入力が制限される携帯端末では、素早く入力するため、ユーザが決まった動き（ジェスチャー）を行い、携帯端末が認識して入力に変換するのは一つの有力な入力手法である。ユーザが行うジェスチャーを認識するため手法も様々に存在する。タッチ画面を利用してストロークのジェスチャーを入力させるや、加速度センサを利用して携帯端末の変動量を計算し、ジェスチャーを認識するという方法もある。しかしタッチ画面は GUI の操作に利用され、ジェスチャーインタラクションを導入すると操作できる GUI の領域が削減できる。カメラ、加速度センサを利用するジェスチャーのパターンが少なく、またユーザは大きく動かさなければ、システムがその動きを検出することが難しい。

逆に磁石の3次元位置と角度の情報も利用すると豊かなジェスチャー種類を利用できると期待できる。詳細な位置と角度を検出できるのでユーザの些細な動き（例えば指先を振る）を認識することが可能になるとユーザの入力の負担を減らすことができるとも考えられる。

1.3 本研究の目的とアプローチ

本研究は磁気センサを利用して、ユーザの指先につけられる磁石の位置と角度を検出し、指先のジェスチャーを認識し、タッチの入力を補足し携帯端末での素早い操作をできるようにするという目的を目指す。また、今までの携帯端末周辺空間のインタラクションよりも些細な動きのインタラクションを認識できるようにする。磁石の位置情報検出の処理とジェスチャー認識の処理を携帯端末でもリアルタイムに実行できるようにすることも本研究の一つの課題とする。そのため本研究で磁石の位置情報と角度の計算は非線形の連立方程式を反復法で解く方法を避け、磁気センサの配列を利用して非線形の部分をできるだけ削減し、線形の処理で計算を行う。ジェスチャーの認識は隠れマルコフモデルやニューラルネットワークなどの複雑なパターン認識手法ではなく、Wobbrock らの 1\$手法を拡張し、認識を行う。

1.4 本論文の構成

本論文は以下の通りに構成されている。2章では本研究の関連研究として携帯端末のジェスチャーの研究と磁石を利用する研究を紹介する。3章で本研究で利用する位置と角度の取得の流れとそのアルゴリズムの評価について述べ、4章で磁石の検出するシステムのハードウェアとソフトウェアの実装について説明する。5章で磁石を利用する入力のジェスチャーのインタラクションを検討する。6章で図形のジェスチャーの認識を評価し、7章では検討したジェス

チャートインタラクションにより作成した応用アプリケーションについて述べる。最後に 8 章で本研究の結果をまとめ、今後の課題について説明する。

第2章 関連研究

本研究は磁石の位置と角度を計算して、携帯端末のジェスチャーインタラクションに応用するため、携帯端末のジェスチャーインタラクションの研究と磁気センシングを利用する研究を関連研究とする。本章はそれらの関連研究について説明し、本研究の特徴を説明する。

2.1 携帯端末のジェスチャーインタラクションの研究

現在、携帯端末のジェスチャーは2種類：モーションジェスチャーと平面ジェスチャーがある。平面ジェスチャーは2次元タッチ画面の上で行われるジェスチャーで、モーションジェスチャーはカメラ、IR 距離センサまたは加速度センサを利用して3次元の動きのジェスチャーと定義された [7]。

平面ジェスチャーはタッチ画面の普及で、また Android SDK が標準ジェスチャー認識ライブラリを提供することにより [8]、研究でも、アプリマーケットのアプリの中でもよく利用されるようになってきた。例として携帯端末の内部のすべてにジェスチャーで検索する Gesture Search[9]、またはジェスチャーを携帯端末の GUI を操作する Gesture Avatar[10] が挙げられる。しかし、タッチの画面の上にジェスチャーを行う場合、他の GUI のオブジェクトとインタラクションすることができなくなり、タッチ画面の操作性が低下する。本研究の磁石入力では3次元入力により、タッチ画面の入力を補足し、ジェスチャー用の入力レイヤーを追加する。

モーションジェスチャーの例として廣部らは高速カメラを利用して、携帯端末向け空中タイピングインタフェースを提案した [11]。この研究ではカメラでユーザの指の3次元位置情報を検出し、仮想 qwerty キーボードでテキスト入力させるシステムを開発した。HenryssonらはAR技術を応用して、携帯のカメラとマーカーを用いる3Dインタラクションを実現した [12]。加速度センサで携帯端末の文字入力、ジェスチャー入力の研究は多く存在した [13]。しかし、加速度センサを利用する場合では操作中デバイスを動かす必要があるため、画面を注視することが難しい。カメラ、距離センサを利用する場合、センサの前でジェスチャーを行わなければならない。またセンサはデバイスの外側に配置しなければならない、手のオクルージョンの問題を起こる可能性もある。逆に磁気はガラスや、プラスチックなどの障害物を通過できるため、磁石と磁気センサを利用する場合ではセンサは携帯端末の内部に内蔵することもでき、自由に配置することができるし、手のオクルージョンの問題も起こらない。磁気センサと磁石を利用するとき、ユーザは磁石を装着する負担があるが、磁石は電池の必要がなく、小型の携帯ストラップにしたり、スタイラスにつけたりすると簡単に持ち歩くことができる。

2.2 磁気センシングを利用する研究

磁気センシングを利用して3次元入力に応用することは多く、Polhemus¹のような製品化されたものもある。Polhemus で利用するアルゴリズムは磁気的非線形方程式を線形の方程式に変換し、高速な計算ができた。しかし、Polhemus を利用するにはアクティブな磁気の発信源が必要とするため、携帯性が低い。

Daniel らはリング形の永久磁石を利用してNenyaという入力デバイスを開発した[14]。ユーザが自分の指につけた磁石のリングを回してナビゲーション入力を行う手法である。リングの回転角度はユーザの手首に巻かれる磁気センサで認識を行った。Han らは磁石を指につけて、指の文字入力手法を提案した[15]。

磁気センサを利用した携帯端末の入力インタフェースの研究としてHarrison らの研究[5]またはRong-Hao らの研究[16]が挙げられる。Harrison らは磁気センサを利用した2次元の入力インタフェースのプロトタイプAbracadabraを提案した。2次元位置情報を応用し、彼らは小型のデバイス用の簡単なインタフェースについて検討した。Rong-Hao らは高密度の1次元の磁気センサの配列を利用してスタイラスにつけた磁石の2次元位置と傾きを検出するGaussenseを提案した。また彼らも磁気センサの配列を利用して磁石で画面周辺のインタラクションを提案した[17]。本研究は3次元の位置情報を利用する入力インタフェースを開発し、さらに携帯端末に応用することを検討する。Chen らは磁石の3次元位置を利用して携帯端末の操作に応用すると目的として、uTrackのシステムを開発したが、彼らのアルゴリズムは処理能力の高いパソコン端末でぎりぎりリアルタイムの計算ができるが、現在の携帯端末の処理能力で利用することが難しい[18]。本研究では線形的な計算方法で磁石の位置と角度を検出して、携帯端末でもリアルタイムの計算ができるように検討と実装を行った。

一方、異なるアプローチでHamed らはiPhoneに搭載されている方向検出用の磁気センサを使って磁気変化を検出し、パターン認識で入力を区別する手法を提案した[19]。この手法で数字の入力や、仮想楽器の操作[20]などに応用できることを彼らは示した。本研究は複数の磁気センサを必要とするが、3次元位と角度を用いることにより、学習の過程が必要のなく、自由なジェスチャーが可能である。

2.3 研究の位置づけ

本研究は2.2で紹介した研究の磁気センシングを利用した手法であり、さらに携帯端末に適応できるように高速な計算手法をカスタマイズした。磁気センシングの特徴を利用して、今までのジェスチャーインタラクションと違う操作ができ、携帯端末の新しい周辺空間インタラクションを調査する。

¹<http://www.polhemus.com/>

第3章 磁石の位置と角度の取得アルゴリズム

本章は本研究で磁石の位置と角度の取得アルゴリズムについて説明する。

3.1 計算モデル

本研究で利用した3次元位置情報検出アルゴリズムはHuら[21]の磁石のトラッキング手法のアイデアに基づいて提案した。磁石の位置をトラッキングするときによく利用する方法は反復法で非線形の連立方程式を解く手法である[22]。Huらのアプローチはこれまでの計算手法と違い、複数の磁気センサのデータから、非線形の要素を省き、線形の連立方程式を立て位置情報と角度の解を求める。彼らの計算手法では5個のセンサのデータを利用して、非線形のアプローチと比べ10倍の速度で位置と角度を計算することができる。彼らの計算モデルは以下に述べる。

磁石の位置は $\mathbf{R}(x_R, y_R, z_R)$ とし、角度 $\mathbf{H}(m, n, p)$ とする。ある磁気センサは $\mathbf{S}$ の位置には位置される場合、磁石からのその磁気センサが感知する磁場強度 $\mathbf{B}_l(B_{lx}, B_{ly}, B_{lz})$ は式(3.1)で表す。

$$\mathbf{B}_l = B_T \left(\frac{3(\mathbf{H} \cdot \mathbf{X})\mathbf{X}}{R_l^5} - \frac{\mathbf{H}}{R_l^3} \right) \quad (3.1)$$

$\mathbf{X} = (\mathbf{S} - \mathbf{R})$ 、 $R_l = \sqrt{x_R^2 + y_R^2 + z_R^2}$ とする。両辺に $(\mathbf{S} - \mathbf{R})$ と外積をかけて、 $\mathbf{X} \times \mathbf{X} = 0$ のため式(3.2)が求められる。

(3.2)

さらにHuらが気づいたのは $\mathbf{H} \times \mathbf{X}$ は $\mathbf{H}$ と直交するため、式(3.2)の両辺に $\mathbf{H}$ と内積をかけたあと、次のシンプルな式が求められる。

$$(\mathbf{B}_l \times \mathbf{X}) \cdot \mathbf{H} = 0 \quad (3.3)$$

すなわち

$$\begin{pmatrix} \mathbf{B}_l x \\ \mathbf{B}_l y \\ \mathbf{B}_l z \end{pmatrix} \times \begin{pmatrix} x_S - x_R \\ y_S - y_R \\ z_S - z_R \end{pmatrix} \cdot \begin{pmatrix} m \\ n \\ p \end{pmatrix} = 0 \quad (3.4)$$

また、彼らは $\mathbf{H}(m, n, p)$ を単位ベクトルと仮定し、 $m' = m/p$ 、 $n' = n/p$ と置いて、 p を式(3.4)から省き、式(3.5)に簡単化できる。

$$\mathbf{F}_l \mathbf{R} = \mathbf{u} \quad (3.5)$$

その中で

$$\mathbf{F}_l = [B_{lx}, B_{ly}, B_{lz}, m', n']$$

$$\mathbf{T} = [y_R - z_R n', z_R m' - x_R, x_R n' - y_R m', m', n'] = [t_1, t_2, t_3, t_4, t_5]$$

$$u = B_{lx} y_S - B_{ly} x_S$$

5つのセンサからのデータを求めることができれば、 $\mathbf{T} = [t_1, t_2, t_3, t_4, t_5]$ の連立方程式を立てることができる。その連立方程式求めた解から元の方程式に代入し、磁石の位置と角度が求められると Hu らが主張した。

しかし、Hu らの計算をそのまま携帯端末のデバイスでのトラッキングに応用するには難しい。彼らのシミュレーションではすべての磁気センサを固定して地磁気を無視して計算を行った。しかし、携帯端末に利用するとき、信号のノイズ、地磁気の影響で実際に計算した位置の正確さが確保することが難しい。我々は彼らのアイデアを元にし、携帯端末用の磁石の位置と角度トラッキング手法にカスタマイズした。

3.2 3次元位置情報の取得

磁石を指につけて携帯端末のタッチ画面を操作する場面を考える。その場合、磁石と携帯がほとんど平行し、相対角度があまり変わらない。角度を固定すると近似の位置を求めることが考えられる。変数を減らすことで最後の計算結果にノイズなどの影響が削減されるとも考えられる。そのため、位置を計算するとき、我々は Hu らの計算モデルの中の角度 $\mathbf{H}$ を (0,0,1) に固定し、計算の流れを変えた (図 3.1)。

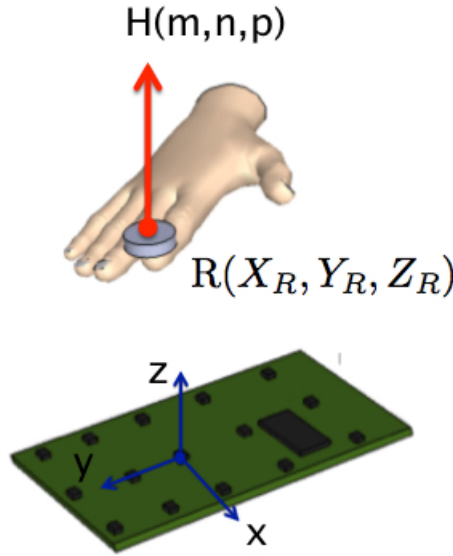


図 3.1: 磁気センサの座標

最初の式 (3.1) を展開すると次の式 (3.6) が求められる。

$$\begin{cases} B_{lx} = B_T \left(\frac{3[m(x_S - x_R) + n(y_S - y_R) + p(z_S - z_R)](x_S - x_R)}{R_l^5} - \frac{m}{R_l^3} \right) \\ B_{ly} = B_T \left(\frac{3[m(x_S - x_R) + n(y_S - y_R) + p(z_S - z_R)](y_S - y_R)}{R_l^5} - \frac{n}{R_l^3} \right) \\ B_{lz} = B_T \left(\frac{3[m(x_S - x_R) + n(y_S - y_R) + p(z_S - z_R)](z_S - z_R)}{R_l^5} - \frac{p}{R_l^3} \right) \end{cases} \quad (3.6)$$

方向ベクトルは $H(m,n,p)=(0,0,1)$ と仮定して、簡略化すると式 (3.7) になる。

$$\begin{cases} B_{lx} = B_T \left(\frac{3(z_S - z_R)(x_S - x_R)}{R_l^5} \right) \quad (1) \\ B_{ly} = B_T \left(\frac{3(z_S - z_R)(y_S - y_R)}{R_l^5} \right) \quad (2) \\ B_{lz} = B_T \left(\frac{3(z_S - z_R)^2}{R_l^5} - \frac{1}{R_l^3} \right) \quad (3) \end{cases} \quad (3.7)$$

(B_l (B_{lx}, B_{ly}, B_{lz}):磁気センサが感知した磁場強度、 S (x_S, y_S, z_S):磁気センサの位置、 B_T :磁石の磁力、)

複雑な B_l と R_l の部分を除外するため、式 (3.7) の (1) を (2) で割り、式 (3.8) が求められる。

$$\frac{B_{lx}}{B_{ly}} = \frac{x_S - x_R}{y_S - y_R} \quad (3.8)$$

同じように、各センサから (3.8) の式を求め、線形の連立方程式をたてることができる。最小二乗法で連立方程式を解くことで、 x_R と y_R は求められる。変数が2つだけなので、厳密的に2つの方程式 (つまり2つのセンサ) があれば、その連立方程式を求めることができる。

高さ z_R は x_R と y_R を元の式に代入し、Newton-Raphson の反復法¹で求められる。例えば式 3.6 の (2) を利用する場合、 x_R と y_R をに代入することで、式 3.9 が求められる。

$$\frac{B_{ly}}{3B_T(y_S - y_R)} \left(\sqrt{(x_S - x_R)^2 + (y_S - y_R)^2 + (z_S - z_R)^2} \right)^5 - (z_S - z_R) = 0 \quad (3.9)$$

式を簡単にするため、

$$M = (x_S - x_R)^2 + (y_S - y_R)^2; N = \frac{B_{ly}}{3B_T(y_S - y_R)} \quad (3.10)$$

とおく。これにより次の z_R の方程式が求められる。

$$M(\sqrt{N + (z_S - z_R)^2})^5 - (z_S - z_R) = 0 \quad (3.11)$$

$$f(z_R) = M(N + (z_S - z_R)^2)^5 - (z_S - z_R) \quad (3.12)$$

とおき、 $f(z_R)$ の微分は

$$f'(z_R) = -5 * M(N + (z_S - z_R)^2)^3(z_S - z_R) + 1 \quad (3.13)$$

¹Endre Suli and David Mayers, An Introduction to Numerical Analysis, Cambridge University Press, 2003.

初期値 $z_{R(0)}$ を決め、

$$z_{R(n+1)} = f(z_{R(n)})/f'(z_{R(n)}) \quad (3.14)$$

を繰り返して計算すると、 z_R の近似解が求められる。Newton-Raphson 法では必ず収束すると言えないが、携帯周辺の解であれば、範囲が狭く、初期値を工夫して設定すれば、発散する場合を少なくすることができる。また、今回の実装では処理を早くするため、計算の流れではまだ地磁気の影響はまだ考慮していない。

3.3 磁石角度を計算するアルゴリズム

以前、センサのノイズ、地磁気の影響により Hu らのアルゴリズムで計算した最後の結果が不正確で携帯端末で利用することが難しいと結論を出したが、途中の結果 $\mathbf{T}(t_1, t_2, t_3, t_4, t_5)$ は多くのセンサ（5 以上）からのデータがある場合、最小二乗法で連立方程式を解くとある程度の正確さがあると実際の実装では分かった。 $t_4 = m'$ また $t_5 = n'$ のため $\mathbf{T}$ の解からそのまま角度を計算することができる。

また、Hu らのモデルで $m'=m/p$ 、 $n'=n/p$ で計算すると、 p が 0 に近づくときノイズの影響が強くなり、 $\mathbf{T}$ の解が不安定である。また m', n' から m, n, p の絶対値は求めることができるが、正負を決めるためまた元の式に代入して計算することが必要とする。そのため、本研究は p の要素を省かなく、式 (3.4) からそのまま $\mathbf{T}'(t'_1, t'_2, t'_3, t'_4, t'_5, t'_6)$ の 6 変数の方程式に変換する。その中で

$$\begin{aligned} t'_1 &= y_{RP} - z_{Rn} \\ t'_2 &= z_{Rm} - x_{RP} \\ t'_3 &= x_{Rn} - y_{Rm} \\ t'_4 &= m \\ t'_5 &= n \\ t'_6 &= p \end{aligned}$$

とする。

$\mathbf{T}'$ をまとめるためせめて 6 個のセンサからのデータが必要とし、Hu らのアルゴリズムより 1 個多くなるが、その分に計算の結果が安定になる。これで、前の位置情報の計算と別に磁石の角度を計算することができる。

3.4 計算アルゴリズムの利点と欠点

本研究の計算アルゴリズムの利点は磁気センサの配列を利用することにより、磁石の位置情報の非線形方程式を線形方程式に変換し、計算を高速化することができる。そのため、初めて携帯端末の処理能力でもその計算をリアルタイムにできるようになった。しかし、この計算アルゴリズムにはいくつかの欠点がある。

一つ目はこの計算モデルにはまだ地磁気を考慮していないため、ユーザは使用中にある程度、携帯端末の方向を固定しなければならない。入力中に携帯端末を方向を変えると、地磁

気の影響が変わり、計算結果がだんだん不正確になる。この問題には、加速度センサなどで携帯端末の移動量を計算し、計算結果を修正するという解決方法が挙げられる。しかし、今度の実装は使用中に、ユーザはあまり携帯端末を動かさないと仮定して実装した。

二つ目は磁石の位置を計算するために、磁石の方向を固定して計算を行ったので、ユーザが磁石を回すときに磁石の位置の計算が不正確になる。そのため、角度の計算結果を参考し、仮定した角度に近いときに限り、位置の計算を利用するというチェックが必要とする。また、計算した角度から、計算した位置を修正することも考えられるが、非線形の計算が必要で、処理時間が上がる可能性が高いため、また計算した角度の結果にエラーが含まれるため利用することが難しく、今度の実装ではそのままの位置の計算結果を利用した。

第4章 システム実装

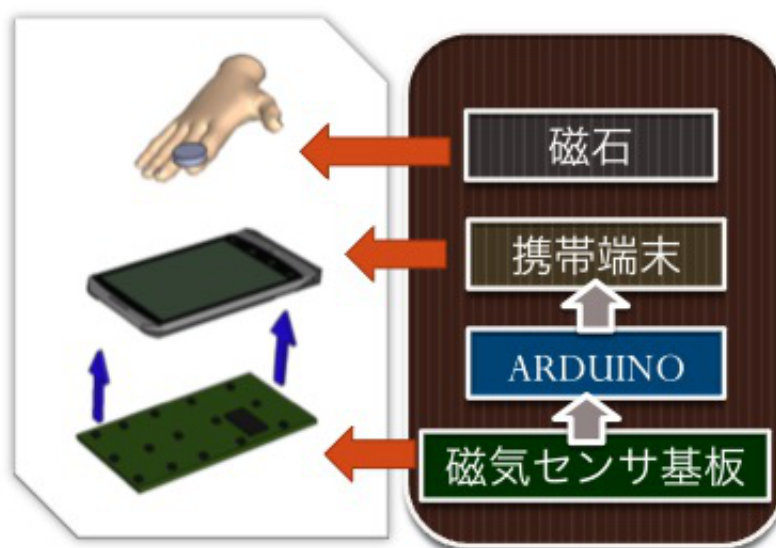


図 4.1: システム構成

4.1 ハードウェアの実装

4.1.1 磁気センサ配列

本研究のアルゴリズムを利用する場合、3次元位置情報を求めるためにせめて2個、角度を検出するためにせめて6個の磁気センサからのデータを取得する必要もある。すなわち、磁石の近くにせめて6つの磁気センサがないと計算することができない。磁気センサを安定化する、また、センサ密度を高くするため磁気センサの基盤を作成した。3 x 5のMAG3110の3軸磁気センサ¹と74HC4067マルチプレクサ²を80mm × 100mmの基盤の上に設置した。MAG3110センサは1000 μ Tまで検知できる3軸磁気センサ。各軸は-1000 μ T ~ 1000 μ Tまで-30000 ~ 30000の値で表現し、80Hzの周波数で提供できる。MAG3110はI²Cイン

¹http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=MAG3110

²<http://www.ti.com/lit/ds/symlink/cd74hc4067.pdf>

タフェースで Arduino³と通信を行う⁴。I²C インタフェースの通信ではデータライン（データ通信用ライン）とクロックライン（同期用ライン）を接続する必要がある。今度の実装では 15 個の磁気センサは同じクロックラインを共有し、マルチプレクサでデータラインを分割すると設計した [付録 A]。

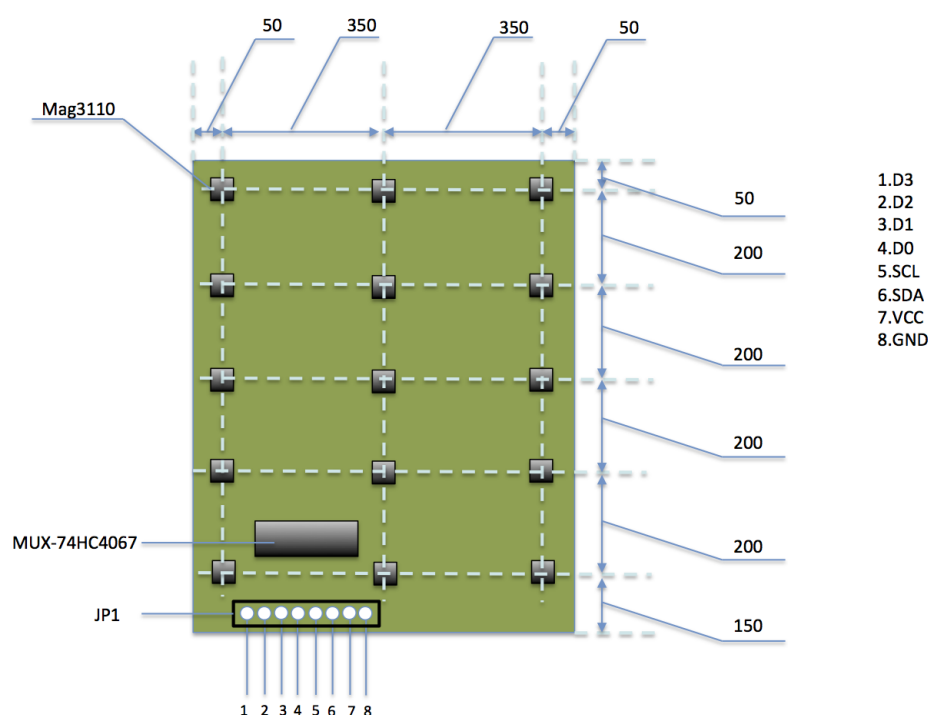


図 4.2: 磁気センサの基盤

4.1.2 携帯端末と通信

磁気センサの基盤は携帯端末の裏面に取り付ける。携帯端末と通信を行うため Bluetooth Mate Gold⁵ の Bluetooth シェルで携帯端末に磁気センサのデータを送信する。送信速度は 115200Hz と設定し、磁気センサを 0 から 14 まで番号をつけて、その番号と各軸のデータをペアでデータを送る。

³<http://arduino.cc/en/Main/arduinoBoardDuemilanove>

⁴<http://arduino.cc/en/reference/wire>

⁵<https://www.sparkfun.com/short/9358>



図 4.3: 携帯の後ろに取り付けた磁石の基盤

4.2 ソフトウェアの実装

4.2.1 通信モジュール

携帯端末は Android 端末 Galaxy Note 3(Android 4.2) を利用した。携帯端末と Arduino の Bluetooth 通信を行うため Bonifar らの Amarino ライブラリ [23] を利用した。Bluetooth の通信を確立は時間がかかる、また不安定であるため、Android 端末では受信データのモジュール (DataReceiver) を独立にし、受信したデータを Intent⁶に添付してブロードキャストする (図 A.1)。

4.2.2 磁石の位置と角度の計算の実装

地磁気、周りの環境の磁気の影響を省くために、データを受信できてから、各々の磁気センサの最初の 10 個のデータの平均を計算して、計算に利用するデータはその平均値からのオフセット値 (平均値を差し引いた値) $\mathbf{B}'_l = (B'_{lx}, B'_{ly}, B'_{lz})$ である。ノイズの影響もあるため、利用するセンサはオフセット値の大きさ $|\mathbf{B}'_l| = \sqrt{(B'_{lx})^2 + (B'_{ly})^2 + (B'_{lz})^2}$ が閾値 E_B より大きいという条件がある (Mag3110 のノイズを参考した上、今度の実装で $E_B = 50$ と設定した)。センサの位置の cm の単位で設定し、真ん中のセンサの位置を原点 (0,0,0) と置

⁶<http://developer.android.com/reference/android/content/Intent.html>

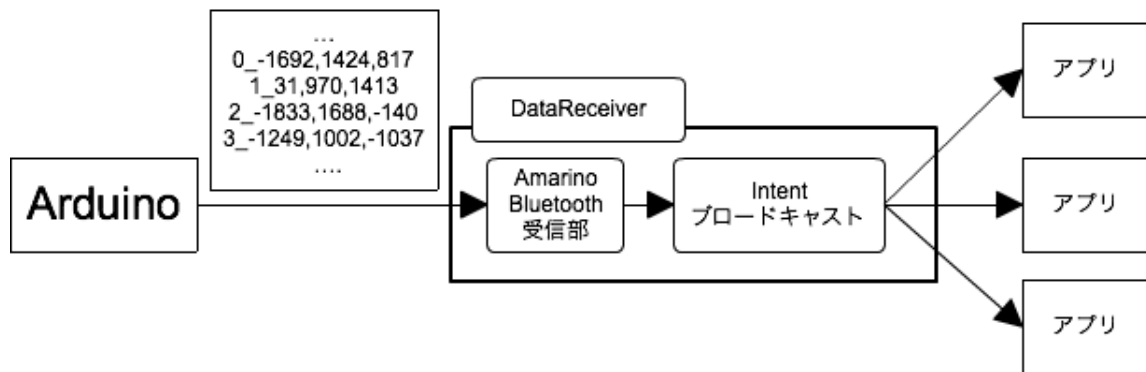


図 4.4: データ受信モジュール

く。また、位置の計算と角度の計算はお互いに関係がないため、速度を上げるために別々のスレッドで計算を行う。

位置の計算

15個のセンサからのデータを利用できる条件をクリアしたセンサが2個以上であれば、位置の計算を行う。各のセンサから式(3.8)を計算し、 x_R と y_R の連立方程式を立てる。連立方程式を最小二乗法で解くために、行列計算のJavaライブラリJAMA⁷を利用した。

z_R を求めるためには式(3.14)の計算を繰り返して、 $z_{R(n+1)} - z_{R(n)}$ の差の絶対値が $\epsilon = 0.001$ より小さくなる場合、計算のループを停止する。計算が収束しない時もあるため、ループが100回以上になれば停止するように設定した。実際の計算で初期値が小さいと設定すると、 $z_{R(n)}$ 値が0に近い値に収束してしまう場合が多いので、初期値は求めたい解より大きくなるように設定した。

磁石は半径1cmのディスクのネオジウム磁石を利用した場合、センサ配列の左右上下の辺から5cm以内、上下15cm以内の空間で、誤差3mm以内で磁石の位置を検出することができる。 $z_R(0) = 20$ と初期化した場合、 z_R を求めるループは20回以下で収束する。平均計算時間は表でまとめたように1ms以下で、一番遅い場合でも20ms以内に終了する。

角度の計算

15個のセンサからのデータを利用できる条件をクリアしたセンサが6個以上であれば、磁石の角度の計算を行う。 $\mathbf{T}'$ の行列を計算できれば、角度 $(m,n,p) = (t'_1, t'_2, t'_3)$ はすぐ求めることができる。

角度の検出は6個のセンサのデータが必要があるため、磁石は半径1cmのディスクのネオジウム磁石を利用した場合、センサ配列の上(または下)の10cmまでしか検出できない。

⁷<http://math.nist.gov/javanumerics/jama/>

利用センサ数	平均計算時間 (μ s)
2	406.012
3	482.089
4	426.433
5	462.120
6	608.773
7	945.124
8	738.418
9	650.285
10	504.163
11	592.515
12	555.073

表 4.1: 7 0 0 データのセットの位置計算の平均時間

この空間内でおおよそ 4 度の誤差までの角度を求めることができる。処理時間は表 4.2.2 のように 1 ms 以下で、特別の場合、処理が長くなっても 3 0 ms 以下で計算が終了する。

利用センサ数	平均計算時間 (μ s)
6	527.8000476
7	391.678
8	654.788375
9	584.3508462
10	436.6262593
11	627.5428235
12	472.6721569

表 4.2: 3 0 0 データのセットの角度計算の平均計算時間

以上の計算はすべて Galaxy Note 3 の端末で行った。これで、携帯端末でも磁石の位置と角度の計算の時間が 1ms 程度で、安易にリアルタイムに携帯端末のインタラクションに応用することができる。今度の実装では Bluetooth 通信を利用するため、無線の通信速度の制限により、1 秒 15 回の計算しかできていない。しかし、磁気センサを携帯端末の内部に内蔵する場合、通信速度が一層高くなり、タッチ入力と同じな高解像度の入力ができる。

第5章 認識ジェスチャーの検討

携帯端末で利用するジェスチャー認識は携帯端末の処理能力の制限があるため、できるだけシンプルで、大きなメモリを必要としなく、処理が早いという要求がある。例えば\$1 手法 [24]、Protractor 手法 [25]、\$3 手法 [26] は携帯端末に適応したジェスチャー認識手法で、処理の早いと認識効率をバランスを取りながら実装したものである。本研究でも同じく磁石の位置と角度を利用して、シンプルで効率よくジェスチャーを認識することを目的とする。

一方、磁石の入力はタッチ画面の入力と違い、画面に触れなくても入力できる。磁石が携帯に近づくだけで、入力モードにも入ってしまい、ユーザが不意にジェスチャーを入力する可能性が高くなる。ジェスチャーの誤認識をさけるために普段の動きに異なるように設計することが必要とする。

本章はそれらの観点から本研究が注目する3つのジェスチャーの種類について議論する。

5.1 相対位置ジェスチャー

磁石と携帯端末との相対位置をそのまま利用して、ユーザの指は運動方向をジェスチャーとして認識する。この種類のジェスチャーは簡単なのでよく研究で利用され、代表として Sven ら [4] は IR センサを利用して携帯端末の周辺で右から左や左から右などの手の動きを認識する HoverFlow システムである。タッチ画面ではスワイプのジェスチャーまたは Flick の操作と同じものである。しかし、タッチ入力と違い、磁石入力のインタフェースは画面外の指の位置を検知することができる。そのため、スワイプの動作は携帯端末の裏で行うことができ、画面をブロックしなくナビゲーション操作などができる (図 5.1)。

しかし、この種類のジェスチャーは単純なので、誤認識する可能性が高い。対策方法の一つとしてジェスチャーの入力スペース、運動速度を制限することが考えられる。たとえば、裏面からナビゲーションの例では、携帯の裏からの2センチまで離れるスペースを入力領域と定義して、その入力の領域内で移動速度 5cm/s 以上の動きをジェスチャーとして認識する。

もう一つの対策方法はタッチ操作と連携し、タッチを認識のトリガーとして扱う。このタッチはさらにオブジェクトの選択操作とすると、「選択+コマンド実行」の一連が自然に行える。

5.2 繰り返しジェスチャー

ユーザが同じ動きを繰り返して行うジェスチャーである。例えば、指を縦軸か横軸に早く振るや、指先を振る (図 5.3) や、指を回すなどのジェスチャーが考えられる。これらのジェス

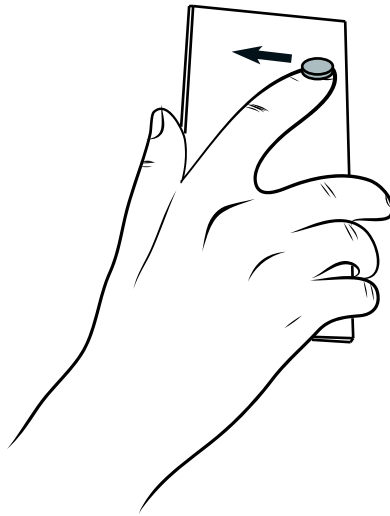


図 5.1: 携帯端末の裏からスワイプ操作でナビゲーションできる

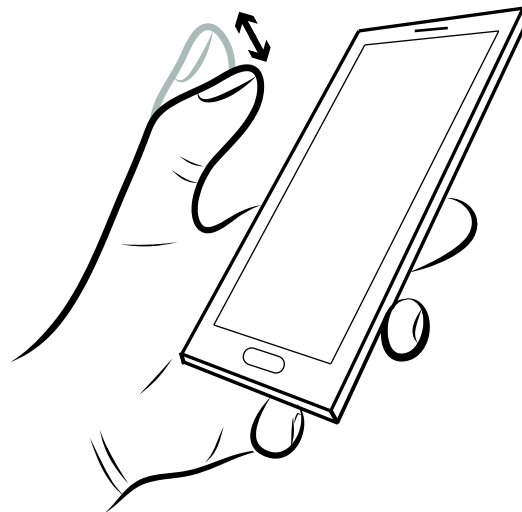


図 5.2: 選択後の移動方向により、オブジェクトをその仮想スペースにコピーする

チャーは位置または角度の変動を観察することで認識することができる。変動が大きくまたある中心の値の回りに分布することを検出することができる。

5.3 図形ジェスチャー

前の2種類のジェスチャーは認識処理は簡単だがパターンが少ない。入力パターンを多くし、簡単に操作に連携できるように本研究は空中の図形ジェスチャーを提案する。人間は3次元でパスを想像するまた記憶するときにはいつも地面平面に参考する。[27] 携帯端末の上に

実装するとき、携帯スクリーンの平面が参考平面とすると、図形のパスがその平面と平行する、また直交すると、ユーザにとって想像しやすくなる。そのため、本研究は2次元の図形に注目し、縦図形と横図形の2種類として認識する。横図形は画面とほぼ平行する図形とし、縦図形は画面とほぼ直交する図形とする。横図形は z -平面の射影の面積が y -平面と x -平面より大きい時に判断され、逆に y,z 平面への射影の面積が大きいときに縦図形と判断される。

しかし、図形ジェスチャーを認識するために、ジェスチャーの始まりとジェスチャーの終わりを区別する必要がある。連続的に位置情報のデータを保存し、ジェスチャーの認識を行う処理は携帯端末に負担をかけて現実的に実現することが難しい。そのため、本研究でタッチ操作と連携し、図形ジェスチャー入力する「タッチ・ジェスチャー・タッチ」の一連操作を利用した。図形ジェスチャーを始める時はユーザ画面にタッチし、画面から離れ、ジェスチャーを空中で書いて、また画面にタッチする 5.4。

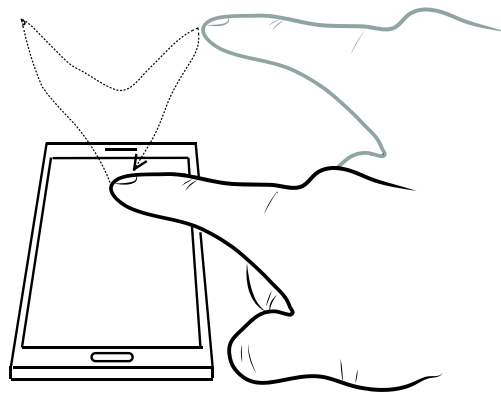


図 5.3: タッチ・ジェスチャー・タッチの入力

誤認識を減らすため、ユーザがジェスチャーを入力していることを意識するために、最初にタッチした点と最後のタッチした点は同じ点とする。それにより、利用するジェスチャーの図形の開始点と終了点と同じという図形に限定する。磁石の入力の精度を考えた上で以下の10種類の図形ジェスチャーを検討した。

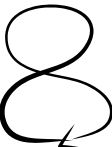
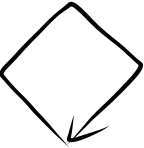
				
Circle	Tomato	Infinite	Light	Moon
				
Triangle	Scirssor	Crown	B	Square

図 5.4: タッチ・ジェスチャー・タッチの入力

第6章 図形ジェスチャー認識の実験

5章で考察した3種類のジェスチャーの中で、相対位置ジェスチャーと繰り返しのジェスチャーは認識が簡単であるが、図形のジェスチャーは複雑で、システム側からは認識が難しく、ユーザの側からは素早く行えるかどうかを検証する必要がある。そのために、図形ジェスチャー認識の実験を行った。

6.1 図形ジェスチャーの認識アルゴリズム

前章で説明したように、3次元の図形を x, y, z の平面の射影の面積で、横図形か縦図形か判断することができる。しかし、面積の計算が複雑で時間がかかるため、各平面への射影の外接長方形の面積で判断を行った。3次元図形のすべての位置の座標 x, y, z の最高値と最低値を $x_{max}, y_{max}, z_{max}, x_{min}, y_{min}, z_{min}$ とすると、各平面の外接長方形は次のようにまとめる。

$$\text{x-平面面積} = |y_{max} - y_{min}| \times |z_{max} - z_{min}|$$

$$\text{y-平面面積} = |z_{max} - z_{min}| \times |x_{max} - x_{min}|$$

$$\text{z-平面面積} = |x_{max} - x_{min}| \times |y_{max} - y_{min}|$$

x-平面面積が一番大きい場合、横図形だと判断され、逆に縦図形と判断される。その後、面積が一番大きい平面への射影をまとめ、2次元の図形として認識する。

2次元の図形を認識するアルゴリズムは Wobbrock らのアルゴリズム [24] を元にして実装した。Wobbrock らのアルゴリズムは次の4つのステップで図形の認識を行う。

- ・ステップ1ーサンプリング：図形の点の数を減らし一定にする ($N=32, 64, \dots$)。
- ・ステップ2ー回転：最初の点と中心点を0度の位置になるように回転する。
- ・ステップ3ースケールとトランスレート：図形を標準正方形に入るように変換する。
- ・ステップ4ーテンプレートのライブラリとのパス距離を計算し、スコアをつける。

本研究のアルゴリズムも同じ流れで行うが、今度のジェスチャーの特徴に対して以下の編集を加えた。

- ・図形の点が少ないためステップ1では、必要の場合、図形の点を減らすではなく、点を増やす。
- ・ジェスチャーを実行する方向はほとんど同じなので回転の必要がなく、ステップ2と4の回転の部分を省く。
- ・ユーザが左右からでも入力できるように対称したものと一緒に検証する。

6.2 実験の内容

図 5.4 からの 10 個の図形ジェスチャーをランダムに表示し、被験者がそのジェスチャーを入力する。各ジェスチャーは 10 回に出現する。今度の実験は 6 人（そのうち 3 人が女、3 人が男、年齢 21～29）の被験者が参加し、各被験者は 10 種類×10 回=100 回入力をしてもらい、合計 600 回のジェスチャーのデータをまとめた。また実験後、被験者にアンケートに記入してもらった。アンケートの内容は以下の自由回答の 4 質問である。

- ・思った通りに図形の入力はできたか？
- ・どの図形の入力が簡単だったか？
- ・どの図形の入力が難しかったか？
- ・自分が 3 次元の図形ジェスチャーを利用するとすれば、利用したい図形の一つを自由に書いてください。

6.3 実験の流れ

今度の実験は携帯端末を机の上において実験を行った。ユーザは磁石を両面テープで人差し指につけ、ジェスチャーを入力する (図 6.1)。実験の画面は図 (6.2) と同じように画面の上部にジェスチャーの種類と名前が表示される。被験者は真ん中の赤い領域から指を置いてジェスチャーの入力を行う。入力した後、入力した図形が表示され、また認識結果も画面の下部に表示され、被験者に確認してもらう。2 秒後に画面は自動的に次の問題に移す。実験の前に被験者に 10 分ほどジェスチャーの入力を練習してもらった。

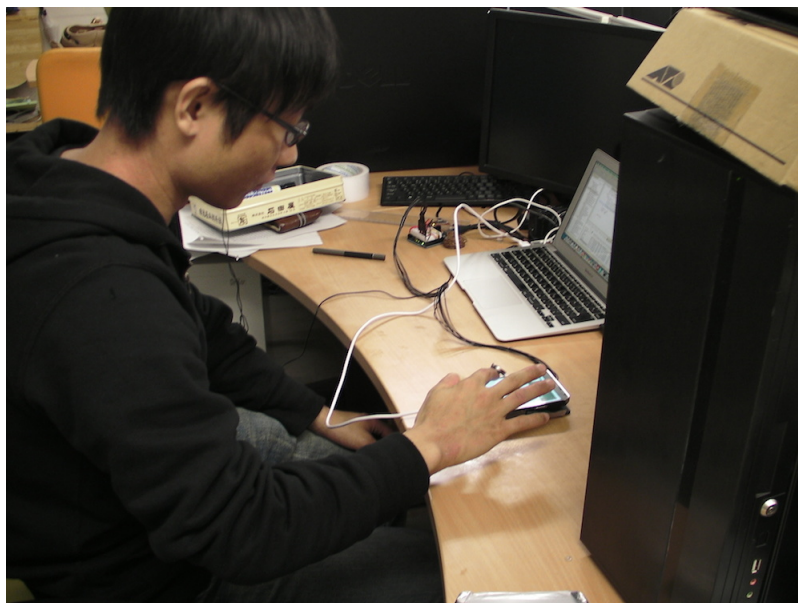


図 6.1: 実験

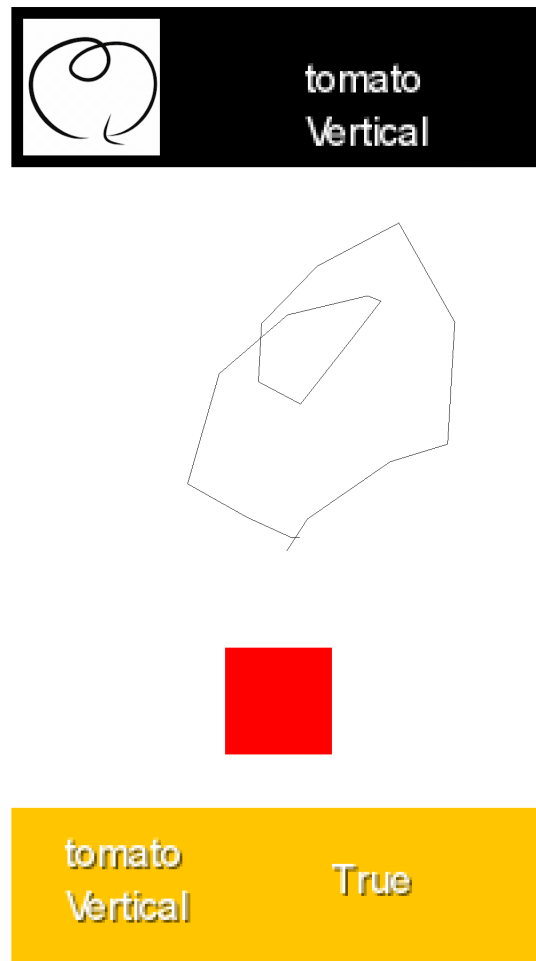


図 6.2: 実験の画面

6.4 実験の結果

被験者はほとんど思った通りにジェスチャーの入力はできたとアンケートに答えていたが、6人の被験者の中で、正解率が多い場合もあるし、入力に慣れず、正解率が50パーセント以下の被験者もいた6.3。被験者の意見により、丸み (infinite, moon) が簡単が角の入力 (crown, scissor) をうまく書けないと、逆に、滑らかな丸みをよく書けないという失敗の2パターンがあった。

図 6.4 が各ジェスチャーの正解率を表示する。アンケートにも記入してもらったように、被験者は circle、triangle、square の入力が難しく、正解率が低かった。特に circle のジェスチャーはよく square と誤認識された。一方、被験者もアンケートに記入したように、複雑に見えるが慣れるときには light, crown のジェスチャーがやりやすい。

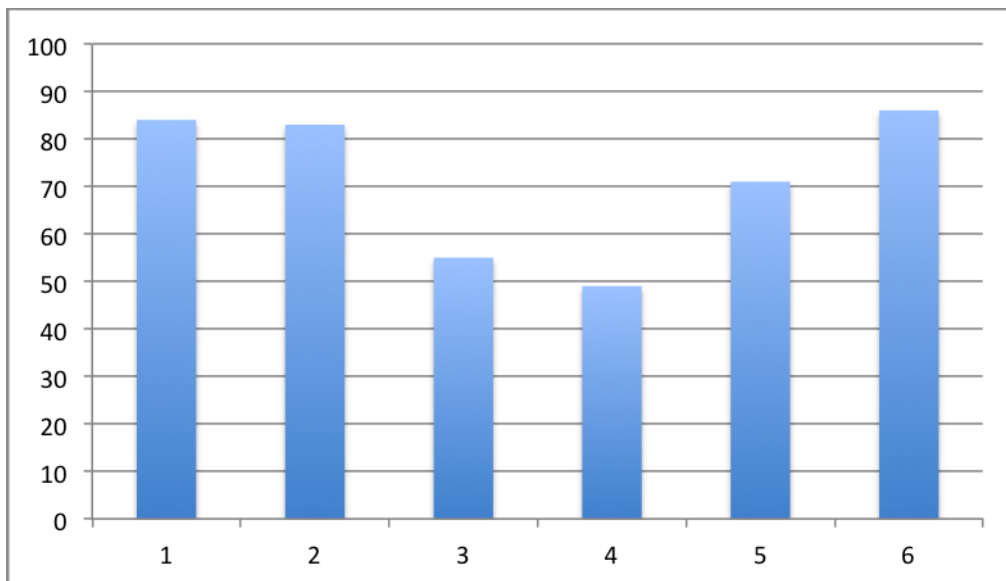


図 6.3: 6 人の被験者の正解率

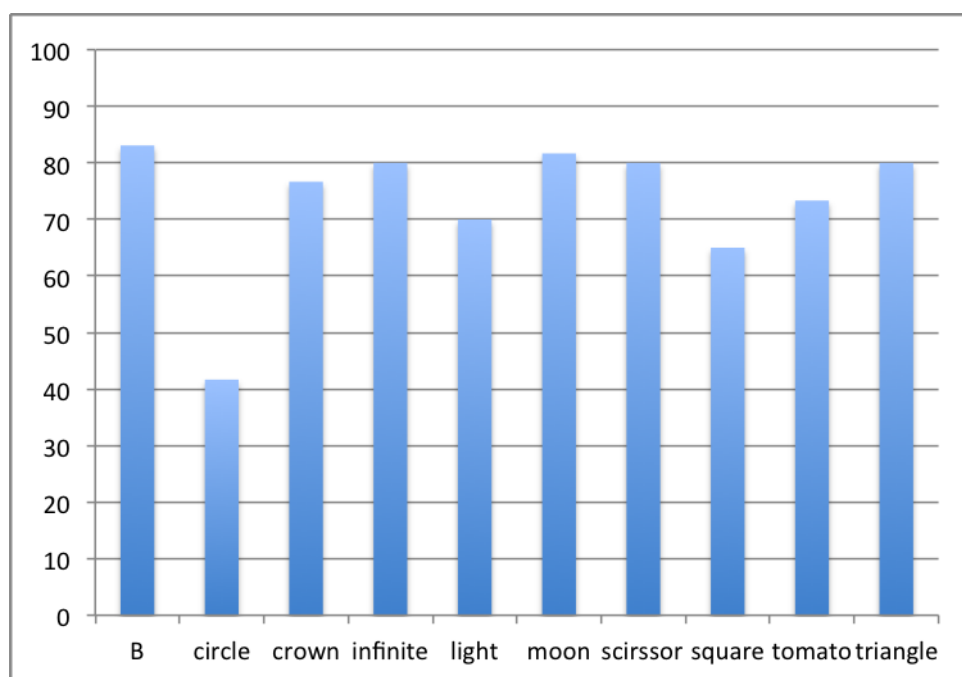


図 6.4: ジェスチャー別の正解率

6.5 検討

実験の結果から図形の角と丸みを入力することが難しいことが分かった。すなわち、角が丸みに、または丸みが角に入力してしまう場合が多いため、間違えても別の図形に認識しな

いように、図形の設計が必要とする。また、尖った角の入力の認識は安易であったので今後の図形の設計に応用すると考えられる。

図形の辞書を拡張をするため、被験者にもジェスチャーの図形を提案してもらった6.5。スターの図形でお気に入りの登録するや、電源のような図形かバツの図形でアプリケーションを閉じるや、楽音の図形で音楽のを再生、停止するなど様々な意見をもらった。

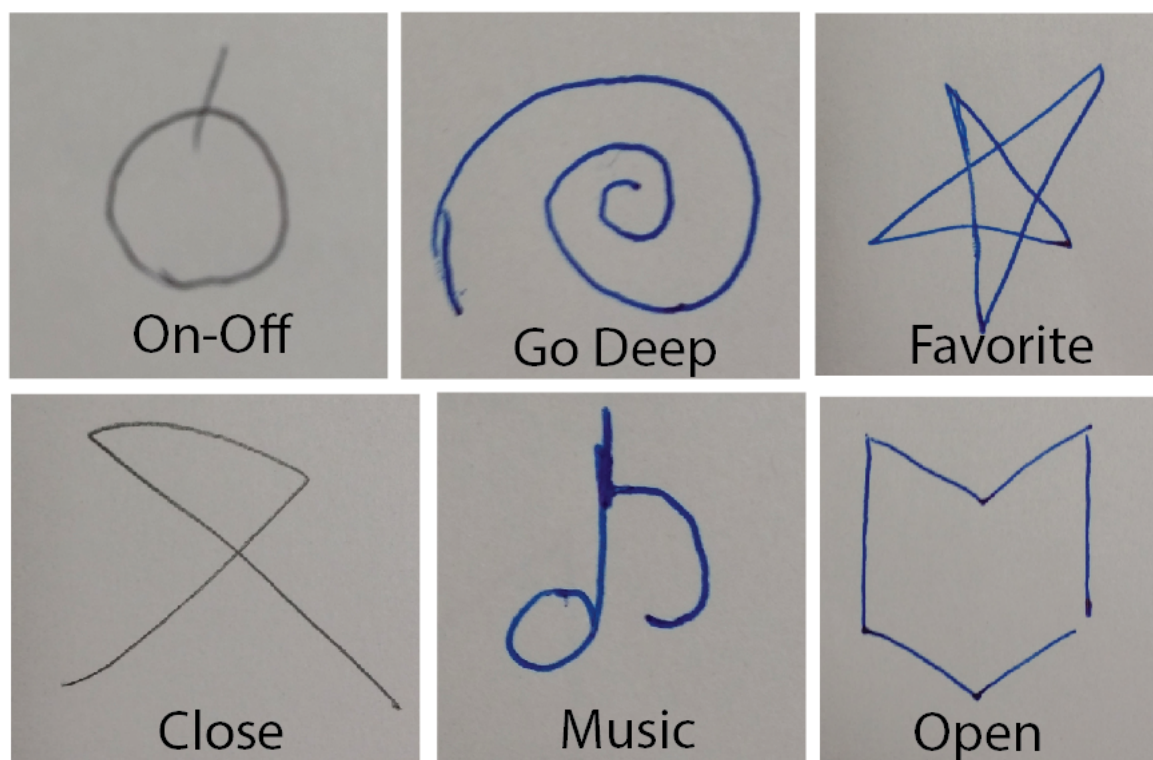


図 6.5: 被験者が提案した図形

また、2次元の図形だけではなく、3次元の図形も面白く、パスワードの入力とかという意見もあった。今度の実装はまだ3次元の認識を実現することが難しいが、今後の課題の一つとして考えられる。

第7章 磁石入力ジェスチャーを応用した描画アプリ

実験の結果を踏まして、提案したジェスチャーの応用の例として、ジェスチャーで操作モードを切り替えできる描画のアプリを実装した。ユーザは図形ジェスチャーで書きたい図形を選択し、また相対位置ジェスチャー、繰り返しジェスチャーで色選択、undo、redo の操作を行える。入力画面の上にタッチ・ドラッグなどで操作する。

7.1 アプリのGUI

；アプリのコントロール操作はすべてジェスチャーで行うため、メニュー、ツールバーは表示なく、画面の上部に現在のモードと入力のタイプしか表示しない(図 7.1)。右下の画面にジェスチャーの参照するためのボタンが用意される。

7.2 操作の流れ

描画アプリケーションで5つのモードがある：

- ・図形入力モード：デフォルトのモードであり、ユーザは図形ジェスチャーで入力したい図形ジェスチャーを選択し、画面にタッチ・ドラッグ操作で図形を描画する。
- ・編集モード：ユーザはタッチして図形を削除する、またはコピー・ペストする。
- ・色選択モード：パレットが表示され、ユーザに色を選択してもらう。
- ・undo-redo モード：空中で指を回しすの繰り返しジェスチャーで undo、redo を行う。

デフォルトでは入力モードで、他のモードに切り替えるためのジェスチャーは以下に提示する。

7.3 操作とジェスチャーの割当

7.3.1 図形入力モードの操作

図形入力モードはデフォルトのモードでアプリに入る時にこのモードに入る。このモードで利用するジェスチャーは表 7.3.2 で説明する。

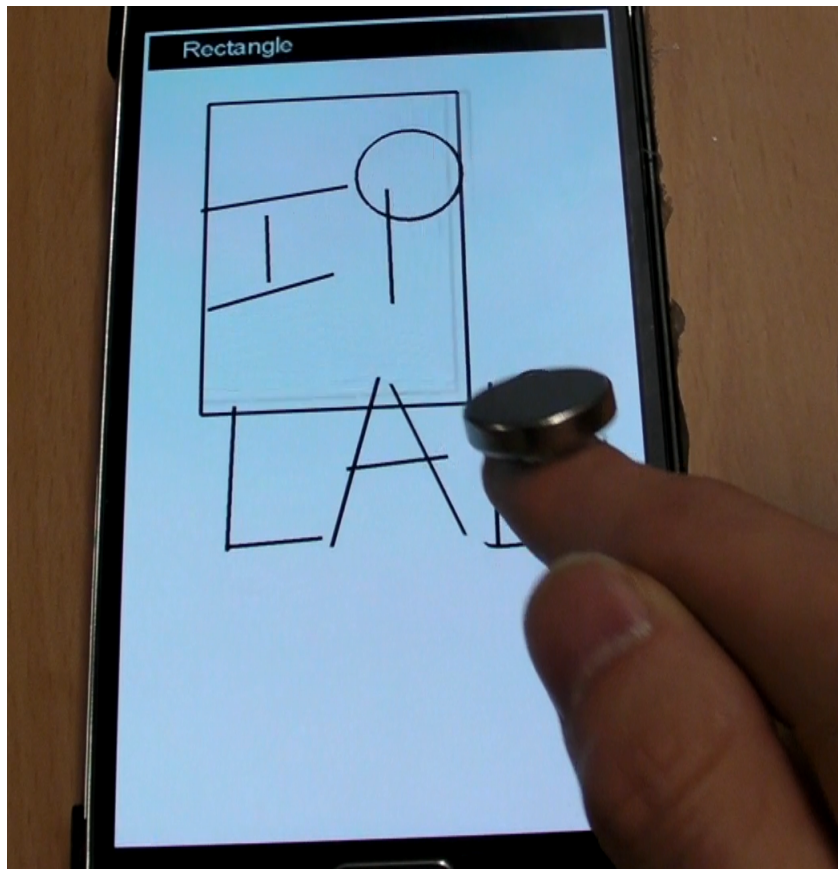


図 7.1: アプリの画面

7.3.2 編集モードの操作

編集モードに切り替えるためにユーザは図形にダブルタッチし、その図形を選択された状態にする。編集モードで図形の内部にタッチし、ドラッグすると図形を移動することができる。外部にタッチし、ドラッグすると図形を回転することができる。編集モードから入力モードに戻るには同じのジェスチャーをもう一回行う。

7.3.3 色選択モードの操作

指先を振ることで色のパレットが表示されて、ユーザに色を選択させる。また同じジェスチャーを実行するとパレット選択を終わらせ、図形入力モードに戻ることができる。

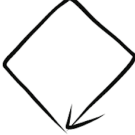
名前	ジェスチャー図形	ジェスチャーの名前	操作説明
直線		triangle	直線の入力に切り替える。直線を入力するとき、最初のタッチの点が始まりの点として、その後ドラッグして直線の終了の点を決める。
長方形		square	長方形の入力に切り替える。長方形を入力するとき、最初のタッチの点と最後に離れた点が長方形の頂点になる。
楕円		tomato	楕円の入力に切り替える。楕円を入力するとき、最初のタッチの点と最後に離れた点が楕円の外接の長方形を設定する
フリー		B	フリースタイル、タッチ・ドラッグして自由な形を書く入力タイプ

表 7.1: 入力モードのジェスチャー

7.3.4 undo-redo モードの操作

指の横振りの繰り返しジェスチャーで undo モードに入り、ジェスチャーを続けるとさらに前に戻ることができる。同じく指の縦振りの繰り返しジェスチャーで redo モードに入り、ジェスチャーを続けるとさらに先の操作に進めることができる。

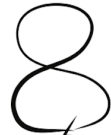
名前	ジェスチャー図形	ジェスチャーの名前	操作説明
カット		scissor	ジェスチャーを実行するとき、バッファーに保存し、図形を削除する
コピー		moon	ジェスチャーを実行するとき、バッファーに保存する
ペスト		B	ジェスチャーを実行するとき、最初にタッチした点にバッファーから図形を貼り、貼った図形は選択された状態になる
削除		infinite	ジェスチャーを実行するとき、図形が削除される

表 7.2: 編集モードのジェスチャー

第8章 今後の課題と発展

本研究は磁石の3次元位置と角度を利用して、携帯端末周辺空間のジェスチャーの入力でタッチ入力を補足し、素早い操作ができるインタフェースを提案した。携帯端末に3次元磁気センサの配列を装着し、線形手法で磁石の位置と角度の計算することができ、携帯端末の処理能力でもリアルタイムの入力ができるようになった。現段階の実装では角度と位置のジェスチャー、または空中の2次元図形ジェスチャーができるようになったが、まだ課題が残り、また発展ができるところが多いと考えられる。実装の課題として、位置情報と角度の計算の結果の修正とキャリブレーションの効率的なアルゴリズムが必要とする。また、地磁気の対策もできたら、磁石の入力インタフェースすぐ現在の携帯端末に応用することが考えられる。入力のインタフェースをさらに入力図形の種類を増やし、さらに3次元の図形ジェスチャーを導入し、様々なタスクに対応できることも考えられる。

謝辞

本論文を執筆するに当たり、指導教員である高橋伸先生をはじめ、田中二郎先生、三末和男先生、志築文太郎先生には、丁寧なご指導と適切な助言を多数いただきました。深く御礼申し上げます。また、IPLAB(田中研究室)の皆様にもゼミなどを通じて大変貴重なご意見をいただきました。特にユビキタスチームの皆様にはグループでのミーティングだけでなく、日常的に多くのご意見やご指摘を頂きました。ありがとうございました。

最後に、私を支えてくれた家族、友人たちに、心より感謝いたします。

参考文献

- [1] Ken Hinckley, Jeff Pierce, Mike Sinclair, and Eric Horvitz. Sensing techniques for mobile interaction. In *Proceedings of the 13th annual ACM symposium on User interface software and technology - UIST '00*, pp. 91–100, New York, USA, November 2000. ACM Press.
- [2] Alex Butler, Shahram Izadi, and Steve Hodges. SideSight: multi-touch interaction around small devices. *Proceedings of the 21st annual ACM . . .*, pp. 3–6, 2008.
- [3] Sven Kratz, Michael Rohs, Dennis Guse, Jörg Müller, Gilles Bailly, and Michael Nischt. PalmSpace. In *Proceedings of the International Working Conference on Advanced Visual Interfaces - AVI '12*, p. 181, New York, New York, USA, 2012. ACM Press.
- [4] Sven Kratz and Michael Rohs. Hoverflow. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services - MobileHCI '09*, p. 1, New York, New York, USA, September 2009. ACM Press.
- [5] Chris Harrison and Scott E. Hudson. Abracadabra. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology - UIST '09*, p. 121, New York, New York, USA, October 2009. ACM Press.
- [6] 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*, p. 1911, New York, New York, USA, May 2012. ACM Press.
- [7] Jaime Ruiz, Yang Li, and Edward Lank. User-defined motion gestures for mobile interaction. In *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11*, p. 197, New York, New York, USA, May 2011. ACM Press.
- [8] Yang Li. Beyond Pinch and Flick: Enriching Mobile Gesture Interaction. *Computer*, Vol. 42, No. 12, pp. 87–89, 2009.
- [9] Yang Li. Gesture Search: A Tool for Fast Mobile Data Access. In *Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology, UIST '10*, pp. 87–96, New York, NY, USA, 2010. ACM.

- [10] Hao Lü and Yang Li. Gesture Avatar: A Technique for Operating Mobile User Interfaces Using Gestures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pp. 207–216, New York, NY, USA, 2011. ACM.
- [11] Yuki Hirobe, Kazunori Funahashi, Takehiro Niikura, Yoshihiro Watanabe, Takashi Komuro, and Masatoshi Ishikawa. In-air Typing Interface for Mobile Devices using a Single Camera. *ITE Technical Report*, Vol. 33, No. 23, pp. 65–68, June 2009.
- [12] Anders Henrysson, Joe Marshall, and Mark Billinghurst. Experiments in 3D interaction for mobile phone AR. In *Proceedings of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia - GRAPHITE '07*, p. 187, New York, New York, USA, December 2007. ACM Press.
- [13] Hai-Ning Liang, Cary Williams, Myron Semegen, Wolfgang Stuerzlinger, and Pourang Irani. User-defined surface+motion gestures for 3d manipulation of objects at a distance through a mobile device. *Proceedings of the 10th asia pacific conference on Computer human interaction - APCHI '12*, No. c, p. 299, 2012.
- [14] Daniel Ashbrook, Patrick Baudisch, and Sean White. Nanya: Subtle and Eyes-free Mobile Input with a Magnetically-tracked Finger Ring. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pp. 2043–2046, New York, NY, USA, 2011. ACM.
- [15] Xinying Han, Hiroaki Seki, and Hikizu Masatoshi. Wearable handwriting input device using magnetic field. In *SICE, 2007 Annual Conference*, pp. 365–368, 2007.
- [16] Rong-Hao Liang, Kai-Yin Cheng, Chao-Huai Su, Chien-Ting Weng, Bing-Yu Chen, and De-Nian Yang. GaussSense: Attachable Stylus Sensing Using Magnetic Sensor Grid Rong-Hao. In *Proceedings of the 25th annual ACM symposium on User interface software and technology - UIST '12*, p. 319, New York, New York, USA, October 2012. ACM Press.
- [17] Rong-Hao Liang, Kai-Yin Cheng, Liwei Chan, Chuan-Xhyuan Peng, Mike Y. Chen, Rung-Huei Liang, De-Nian Yang, and Bing-Yu Chen. GaussBits: Magnetic Tangible Bits for Portable and Occlusion-Free Near-Surface Interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13*, p. 1391, New York, New York, USA, April 2013. ACM Press.
- [18] Ke-Yu Chen, Kent Lyons, Sean White, and Shwetak Patel. uTrack: 3D input using two magnetic sensors. *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, pp. 237–244, October 2013.

- [19] Hamed Ketabdar, Mehran Roshandel, and Kamer Ali Yüksel. Towards using embedded magnetic field sensor for around mobile device 3D interaction. In *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services - MobileHCI '10*, p. 153, New York, New York, USA, September 2010. ACM Press.
- [20] Hamed Ketabdar, Hengwei Chang, Peyman Moghadam, Mehran Roshandel, and Babak Naderi. MagiGuitar. In *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services companion - MobileHCI '12*, p. 181, New York, New York, USA, September 2012. ACM Press.
- [21] Chao Hu, M.Q.-H. Meng, and M Mandal. A Linear Algorithm for Tracing Magnet Position and Orientation by Using Three-Axis Magnetic Sensors. *Magnetics, IEEE Transactions on*, Vol. 43, No. 12, pp. 4096–4101, 2007.
- [22] P. Kucera V. Schlageter, P.-A. Besse, R.S. Popovic. Tracking system with five degrees of freedom using a 2D-array of Hall sensors and a permanent magnet. *Sensors and Actuators A: Physical*, Vol. 92, No. 1, pp. 37–42, 2001.
- [23] Bonifaz Kaufmann and Leah Buechley. Amarino. In *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services - MobileHCI '10*, p. 291, New York, New York, USA, September 2010. ACM Press.
- [24] Jacob O. Wobbrock, Andrew D. Wilson, and Yang Li. Gestures without libraries, toolkits or training: a \$1 recognizer for user interface prototypes. In *Proceedings of the 20th annual ACM symposium on User interface software and technology - UIST '07*, p. 159, New York, New York, USA, October 2007. ACM Press.
- [25] Yang Li. Protractor. In *Proceedings of the 28th international conference on Human factors in computing systems - CHI '10*, p. 2169, New York, USA, April 2010. ACM Press.
- [26] Sven Kratz and Michael Rohs. The \$3 recognizer. In *Proceedings of the 15th international conference on Intelligent user interfaces - IUI '10*, p. 419, New York, New York, USA, February 2010. ACM Press.
- [27] Manuel Vidal and Alain Berthoz. Navigating in a Virtual 3D Maze: Body and Gravity, Two Possible Reference Frames for Perceiving and Memorizing. *Spatial Cognition & Computation*, Vol. 5, No. 2-3, pp. 139–161, September 2005.

付 録 A 基盤の設計

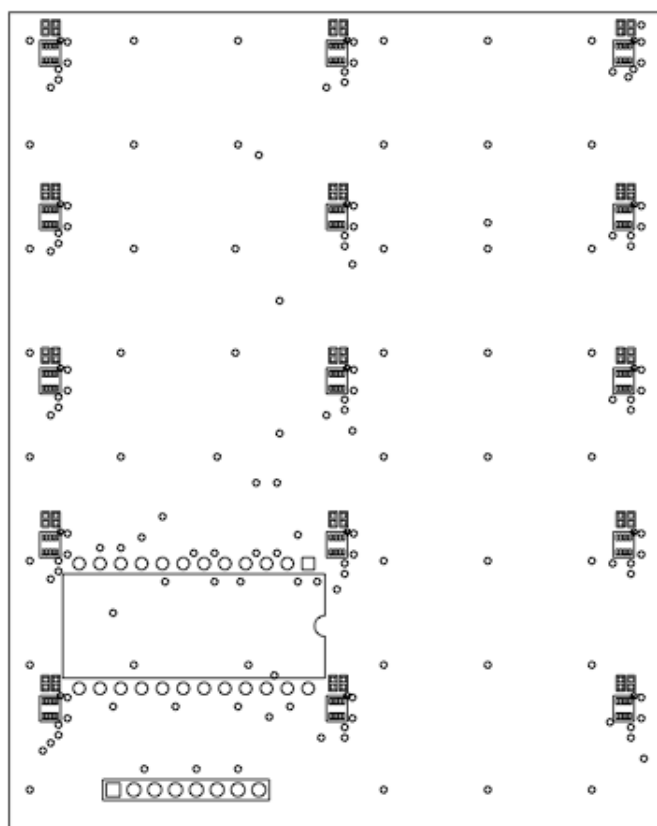


図 A.1: 基盤実装図

