

筑波大学 情報学群 情報メディア創成学類

卒業研究論文

3Dモデリングのための輪型入力装置

大江 龍人

指導教員 志築文太郎 三末和男 田中二郎

2011年2月

概要

3D モデリングにはスカルプティングと呼ばれる変形操作がある。スカルプティングは粘土をこねる様な操作という特徴を持つ。本論文は3D モデリングの中でスカルプティングに焦点を当て、これに関する問題の解決や要求の満足を図る。そのために、本論文は輪型入力装置である「わくこん」とわくこんを用いた3D モデラ的设计と実装を行った。開発したわくこんは、距離センサ、赤外線カメラ、加速度センサを備える。これらにより、手の姿勢とジェスチャの認識、3次元方向のポインティング、わくこんの回転操作が可能になる。手の姿勢に関する評価実験と本システムを用いた3D モデルの自由制作を通して、3次元方向の移動が可能であることや、手を用いた掴む操作が可能であることを示した。その一方で、手の挿入位置に関する問題や3D モデリングのユーザビリティに関する問題を明らかにし、それらの問題についての議論を行った。

目次

第 1 章	はじめに	1
1.1	背景	1
1.2	3D モデリングの問題	1
1.3	3D モデリングに対する要求	2
1.4	本研究の目的	3
1.5	本論文の構成	3
第 2 章	関連研究	4
2.1	手の姿勢やジェスチャを認識する入力装置	4
2.2	輪型入力装置	5
第 3 章	システム設計	6
3.1	システムの設計方針	6
3.2	入力装置の設計	6
3.2.1	入力装置の形状設計	7
3.2.2	輪型入力装置の設計	8
3.3	輪型入力装置を用いた 3D モデラの設計	8
3.3.1	3D モデラの要件	8
3.3.2	インタラクション手法の設計	9
第 4 章	輪型入力装置の試作	12
4.1	プロトタイプ 1 の試作	12
4.1.1	駆動回路	12
4.1.2	ハードウェア実装	12
4.1.3	プロトタイプ 1 のセンシング特性	14
4.2	プロトタイプ 2 の試作	16
4.2.1	駆動回路	16
4.2.2	ハードウェア実装	16
4.2.3	ソフトウェア実装	19
4.2.4	プロトタイプ 2 のセンシング特性	19
4.3	回転検出部とポインティング位置検出部の実装	22
第 5 章	輪型入力装置を用いた 3D モデラの実装	23
5.1	認識部の実装	23

5.1.1	姿勢認識部の実装	23
	SVM 学習	23
	SVM 認識	25
5.1.2	ジェスチャ認識部の実装	25
5.2	3D モデラの実装	26
5.2.1	3D モデラの概観	26
5.2.2	各操作の実装	27
第 6 章	手の姿勢に関する評価実験	30
6.1	実験方法	31
6.2	実験結果	31
6.2.1	視覚フィードバックが無い場合	31
6.2.2	視覚フィードバックが有る場合	32
6.3	考察	32
第 7 章	本システムを用いた 3D モデルの自由制作	35
7.1	実験方法	35
7.2	実験結果	36
7.3	考察	36
第 8 章	議論	39
8.1	手の挿入位置に関する議論	39
8.1.1	姿勢認識部の改善による解決	40
8.1.2	ハードウェア部分の改善による解決	40
8.1.3	姿勢のフィードバックによる解決	41
8.2	3D モデリングのユーザビリティに関する議論	41
第 9 章	結論	42
	参考文献	43

図目次

1.1	スカルプティングの操作例	1
3.1	想定される入力装置の形状	7
3.2	設計する輪型入力装置	8
3.3	目標とする 3D モデル	8
3.4	視点の回転操作	10
3.5	原型となる球の生成操作	10
3.6	移動操作	10
3.7	回転操作	10
3.8	拡大・縮小操作	11
3.9	掴む操作	11
3.10	スカルプティング領域の拡大・縮小	11
4.1	プロトタイプ 1 の回路図	13
4.2	プロトタイプ 1 の概観	13
4.3	プロトタイプ 1 のセンシング値	15
4.4	プロトタイプ 2 で用いる距離センサ	16
4.5	RPR-220 のセンシング特性	16
4.6	プロトタイプ 2 の回路図	17
4.7	紙粘土を用いて整形した輪	18
4.8	輪に埋め込まれた RPR-220	18
4.9	プロトタイプ 2 の配線後の状態	18
4.10	プロトタイプ 2 の配線を拡大した様子	18
4.11	ポリプロピレンのカップにより覆われた様子	19
4.12	ウレタンにより覆われた様子	19
4.13	プロトタイプ 2 のセンシング値を取得する処理	20
4.14	プロトタイプ 2 のセンシング値	21
4.15	プロトタイプ 2 に取り付けられた赤外線カメラ	22
4.16	プロトタイプ 2 が検出する回転方向	22
5.1	SVM 学習の大まかな処理	24
5.2	3D モデルの概観	26
5.3	実際にわくこんを用いて 3D モデリングを行う様子	26

5.4	画面外にポインティングする場合の表示	27
5.5	画面外にポインティングしている様子	27
5.6	領域の決定画面	28
5.7	引っ張る操作を行う画面	28
6.1	被験者に示す視覚フィードバック	30
6.2	視覚フィードバックが無い場合に被験者が示した手を開いた姿勢	31
6.3	視覚フィードバックが無い場合の被験者 B, E の手を開いた姿勢	34
6.4	被験者 C のポインティング姿勢	34
7.1	手描きした制作目標	35
7.2	実際に作成した 3D モデル	36
7.3	視点の回転操作後の画面	37
7.4	足を変形させた際のカーソルの大きさ	38
8.1	手を開いた姿勢の挿入位置によるセンシング値の違い	39

第1章 はじめに

1.1 背景

3D モデリングではポリゴンの頂点や制御点を操作し 3D モデルを作成することが多い。これらの操作には押し出し、回転、平行移動等の操作がある。頂点や制御点の操作を用いて点列を任意の方向に平行移動させ、3D モデルを作成する。

3D モデリングにはスカルプティングと呼ばれる変形操作がある。スカルプティングは頂点や制御点を移動させる操作であり、点列の移動は平行移動よりも複雑に行われる。スカルプティングには 3D オブジェクトを引く、押す、掴む等の操作がある。これらの操作例を図 1.1 に示す。

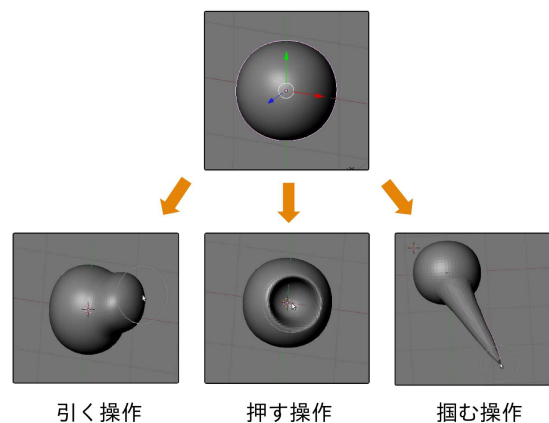


図 1.1: スカルプティングの操作例

1.2 3D モデリングの問題

従来 3D モデリングには入力装置としてマウスやスタイラスが用いられてきた。これらの入力装置を用いた 3D モデリングには以下に挙げる問題がある。

モード切り替え問題 モードの切り替えが頻繁に起こる問題を「モード切り替え問題」と定義する。モード切り替え問題は 3D モデリングの操作の種類が多いことから生じる。3D モデリングの操作には 3D オブジェクトの移動、回転、縮小等の操作、制御点の移動やスカルプティング等の変形操作、環境光の設定、テクスチャの設定など多くの操作がある。また、これらの操作の他にもアニメーションやゲームを作成するための操作を備える 3D モデラも存在する。以上の様に 3D モデ

リングでは操作の種類が多い。そして、操作を切り替えるためにユーザは GUI のメニューからモードを切り替える必要がある。モードの切り替えは頻繁に行われ、効率的に 3D モデルを作成したい場合は障害となり得る。

3 次元操作問題 3 次元方向の操作が難しい問題を「3 次元操作問題」と定義する。3 次元操作問題は 3D モデリングに用いられるマウスやスタイラスが 2 自由度であることから生じる。自由度は以下の様に定義される [BJKP05]。

本体の動作における独立した次元の数。

入力装置における自由度はその入力装置の実世界での動き方を表している。つまり、2 自由度の入力装置は x と y の 2 次元平面上の移動が主な動作となる。この様な入力装置は一度の操作に 2 次元方向の操作を行う事はできる。しかしながら、一度に 3 次元方向の操作を行うことはできない。例えば、スカルプティングの掴む操作は画面の上下左右に掴むことはできるが、奥行き方向に掴むことはできない。

1.3 3D モデリングに対する要求

3D モデリングには 2 つの要求があると考えられる。1 つ目の要求は 3D モデリング、特にスカルプティングの要求である。以下に 2 つの要求を示す。

手を使う要求 手全体を用いてスカルプティングを行う要求を「手を使う要求」と定義する。スカルプティングには粘土をこねる様な操作という特徴があり、手を使う要求はこの特徴から生じる。

スカルプティングの押す操作や掴む操作は、実際に手を用いて粘土にさわる様な操作感をユーザに与える。この理由には、スカルプティング操作が点列の平行移動よりも実世界の変形操作に近く、ユーザが変形操作を理解し易いということが挙げられる。

マウスやスタイラスを用いる場合でも、手を用いて粘土にさわる様な操作感が得られるかも知れない。しかしながら、この操作は手全体を用いる操作ではない。実際に手全体を用いて掴む操作を行うことにより「手を用いて粘土にさわる様な」スカルプティングの特徴を活かすことができる。

大画面に対する要求 大画面環境において 3D モデリングを行う要求を「大画面に対する要求」と定義する。3D モデリングはデスクトップ環境にて行われることが多い。これに対し、大画面環境においても 3D モデリングを行いたいという要求が存在する。

大画面環境で行う 3D モデリングの例に自動車の設計が挙げられる。自動車の設計にはテープドロ잉という作業がある。テープドロ잉とは自動車の大まかな曲線をテープを用いて描く作業である。この曲線は実際の自動車の大きさに描かれる。そのため、3D モデリングを通してテープドロ잉を行う場合その環境は大画面環境となる。

大画面環境において 3D モデリングを行う場合、デスクがない環境が想定される。この様な環境ではマウスやスタイラスの置く場所は存在しない。それゆえ、マウスやスタイラス以外の入力装置を使用する必要がある。

1.4 本研究の目的

本研究は上述した問題を解決し要求を満たすことで、手全体を用いて 3D モデリングを行えるようにすることを目的とする。そのためのアプローチとして、輪型入力装置とその入力装置を用いた 3D モデラを設計し、実装する。

1.5 本論文の構成

第 1 章以降の本論文の構成は以下の通りである。第 2 章で関連研究を紹介する。第 3 章でシステムの設計について述べ、第 4 章と第 5 章でその実装について述べる。第 6 章で手の姿勢に関する実験について述べ輪型入力装置の特徴を明らかにし、第 7 章でユーザビリティに関する問題や要求を明らかにする。第 8 章で実験を通して得られた発見を基に議論を行い、最後に第 9 章で本研究の結論について述べる。

第2章 関連研究

3D モデリングを行うために様々な入力装置が研究されてきた。本章ではこれらの研究について述べる。

また、輪型入力装置を用いた研究も成されてきたのでこれについても述べる。

2.1 手の姿勢やジェスチャを認識する入力装置

3次元操作問題やモード切り替え問題の解決を目的とし、さらに、手を使う要求を満たす研究が成されてきた。これらの研究は、手の姿勢とジェスチャを認識する入力装置とその3Dモデリング手法に関するものである。

グローブ型入力装置

Sato ら [SSO04] は両手に CyberGlove¹を装着し、3次元サーフェスを変形する手法を提案した。CyberGloveは有線接続の入力装置であるが、輪型入力装置は無線接続である。Nishino ら [NUK98] は両手に CyberGlove を装着し、ろくろを用いて粘土を変形させる様に3Dモデリングを行う手法を提案した。Nishino らの研究では間接的な手のジェスチャを用いるが、本研究は掴む操作等の直接的な手のジェスチャを用いる。Schkolne ら [SPS01] は Surface Drawing と名付けられた3Dモデリング手法を提案した。Surface Drawing では手のひらの動きを用いて、3次元サーフェスを作成する。本研究は球のプリミティブを変形させることにより3Dモデリングを行う。

生のセンシング値を活かした入力装置

Lee ら [LHS06] は静電容量センサを備えた正12面体の据え置き型入力装置を開発し、手をかざすことによる3Dモデリング手法を提案した。Smith ら [STP08] は162個の圧力センサを備えた球状の把持型入力装置を開発し、手をセンサに押し込むことによる3Dモデリング手法を提案した。Murakami ら [MN94] はポリウレタンを用いた弾性のある把持型入力装置を開発し、入力装置を押す、曲げる、ひねることによる3Dモデリング手法を提案した。以上の3つの研究は生のセンシング値を活かし3Dモデリングを行う。輪型入力装置は生のセンシング値を活かすのみではなく、機械学習を通して得られる手の姿勢を用いて3Dモデリングを行う。

ビジョンを用いた入力装置

Gross ら [GK01] はテンプレートマッチングを用いて手の姿勢を認識し、その姿勢を用いて3Dモデリングを行う手法を提案した。Gross らの研究は使用環境として黒い手袋の装着と白い背景を必要とする。本研究は使用環境にこのような制限は無い。横川ら [横川 07] はステレオカメラを用

¹<http://www.cyberglovesystems.com/>

いてユーザの手振り動作から成る 3 次元点群を取得し，その点群を用いて 3D モデルを作成する手法を提案した．ステレオカメラを用いた手法では両手の 3 次元位置を観測可能である．しかしながら，配置によってはオクルージョンが発生し手を観測できない欠点がある．本研究は入力装置自体が手の情報を取得可能なセンサを備えるので，このようなオクルージョンの問題が無い．

2.2 輪型入力装置

輪型入力装置を用いた研究がある．Nagashima[Nag10] は音楽を演奏する目的で，2 つの輪により構成される Peller-Min を開発した．Peller-Min の輪にはそれぞれ 8 個の距離センサが備え付けられ，ユーザはテルミンを演奏する様に Peller-Min を演奏する．Peller-Min に対し，本研究は 3D モデリングのための入力装置である．

第3章 システム設計

本章では第1章に挙げた要求と問題を受け，システムの設計方針を定義する．その設計方針を基に，輪型入力装置と輪型入力装置を用いた3Dモデラの設計を行う．

3.1 システムの設計方針

第1章に挙げた3Dモデリングに対する要求と問題を受け，システムの設計方針を以下の様に定める．

手を使う要求 手全体を用いて操作を行うために，手の姿勢とジェスチャを認識する

大画面に対する要求 ユーザが様々な位置と姿勢を取っても3Dモデリングを行うことができる

モード切り替え問題 手を用いて全ての操作を行うことでモード切り替えを不要にする

3次元操作問題 3次元の移動操作と回転操作を容易にする

この設計方針の基，輪型入力装置と3Dモデラの設計を行う．

3.2 入力装置の設計

システムの設計方針を受けると入力装置の条件は以下のものとなる．

条件1 手の姿勢とジェスチャを認識する

条件2 計算機に無線接続される

条件3 照明などの外部環境の影響を受けない

条件4 3次元の移動を検出する

条件5 3次元の回転を検出する

条件1は「手を使う要求」と「モード切り替え問題」，条件2と条件3は「大画面に対する要求」，条件4と条件5は「3次元操作問題」にそれぞれ対応する．

3.2.1 入力装置の形状設計

満たすべき条件から入力装置の形状を決定する。まず、条件 1 を満たす入力装置について考える。手の姿勢とジェスチャを認識するために、以下の 3 つに示す方式のいずれかを採用する。

- 入力装置自体に備わるセンサを用いる方式
- ビジョンを用いる方式
- データグローブを用いる方式

ビジョンを用いる方式は一般的なカメラを用いる場合、照明環境の影響を受けやすく条件 3 に適さない。また、データグローブを用いる方式は計算機に有線接続される場合が多く条件 2 に適さない。したがって、入力装置自体に備わるセンサを用いる方式を採用する。

入力装置には把持して用いられる入力装置と把持して用いられない入力装置の 2 種類に分類できる。前者の入力装置は後者と比較して、入力装置自体の姿勢を活かす事ができ条件 5 に有効である。したがって、把持して用いられる入力装置を採用する。

入力装置の形状に関する以上の議論から、入力装置の条件として以下の 2 つが加わった。

条件 6 入力装置自体にセンサが備わる

条件 7 把持して用いられる

これらの条件を満たす様な入力装置の形状は図 3.1 に示す形状になると想定される。

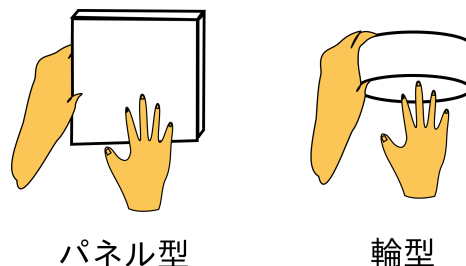


図 3.1: 想定される入力装置の形状

今回は輪型の入力装置を採用する。なぜなら、輪型の入力装置が手を使う要求を満たす入力装置であると考えたからである。パネル型の場合、手全体では無く指先を用いたタッチ操作が行われる。これに対し、輪型の場合は輪の内側で手全体を用いた操作が行われる。

スカルプティングの掴む操作を考えてみる。実世界の掴む操作と同様の操作を行う場合、パネル型では全ての指先を内側へ移動する操作になると考えられる。これに対し輪型では輪の内側で、開いた手を閉じる操作になると考えられる。輪型での操作は 3 次元空間を掴む操作であり、パネル型と比べ実際の掴む操作に近い。

3.2.2 輪型入力装置の設計

設計する輪型入力装置を図 3.2 に示す。この輪型入力装置を「わくこん」と名付ける。わくこんの形状は直径 20cm の輪である。直径 20cm という大きさを採用した理由は片手で把持しやすく、手全体が輪の内側に入る大きさとして適切であると考えたからである。

わくこんは条件 1 を満たすために距離センサを備える。輪の内側に向けて距離センサを等間隔に配置することで、手の姿勢とジェスチャを認識する。また、条件 4 を満たすために赤外線カメラを備える。さらに、条件 5 を満たすために加速度センサを備える。

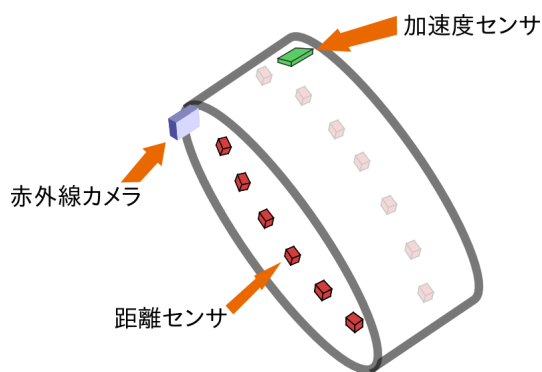


図 3.2: 設計する輪型入力装置



図 3.3: 目標とする 3D モデル

3.3 輪型入力装置を用いた 3D モデルの設計

輪型入力装置を用いた 3D モデルの設計を行う。3D モデルの設計を行うために、必要な操作を洗い出し 3D モデルの要件を定義する。そして、必要な操作に対応するインタラクション手法を設計する。

3.3.1 3D モデルの要件

3D モデルに必要な操作を洗い出すために、作成する 3D モデルの目標を blender¹を用いて作成した。作成した 3D モデルを図 3.3 に示す。この 3D モデルを作成するためには以下の操作が必要であった。

- ナビゲーション操作
 - － 視点の回転操作
- 3D オブジェクトに対する基本的な操作
 - － 原型となる球の生成
 - － 移動操作

¹<http://www.blender.org/>

- 回転操作
- 拡大・縮小操作
- スカルプティング操作
 - 掴む操作
 - スカルプティング領域の拡大・縮小

掴む操作は3D オブジェクトの一部範囲を任意のカーソル位置に引っ張ることである。また、スカルプティング領域は掴む操作を適用する範囲である。

以上の操作が設計する 3D モデラに必要な操作である。

3.3.2 インタラクション手法の設計

必要な操作に対応するインタラクション手法を設計する。「モード切り替え問題」に対するシステムの設計方針から、全ての入力手の姿勢とジェスチャを用いて行われる。

わくこんを用いた基本となるインタラクション手法を以下に示す。

1. 非利き手を用いてわくこんを把持する
2. わくこんを画面に向ける
3. 輪の内側に利き手を挿入する
4. 利き手の姿勢やジェスチャを用いて操作を行う

視点の回転 視点の回転操作を図 3.4 に示す。操作方法は以下となる。

1. わくこんを画面外に向ける
2. 利き手を輪に添える
3. 利き手を輪に添えたままわくこんを回転させる

上記の操作を行うと、わくこんを回転させる方向に視点が回転される。

原型となる球の生成 原型となる球の生成操作を図 3.5 に示す。球を生成したい座標にわくこんを向け、利き手を数回左右に振ると球が生成される。この操作を行うと、わくこんのポインティング位置に球が生成される。

移動操作 移動操作を図 3.6 に示す。操作方法は以下となる。

1. 3D オブジェクトにわくこんを向ける
2. 利き手をポインティングの姿勢にする
3. ポインティングの姿勢のままわくこんを移動させる

わくこんを移動させる方向に 3D オブジェクトが移動される。

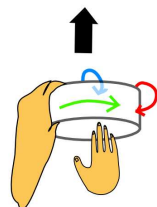
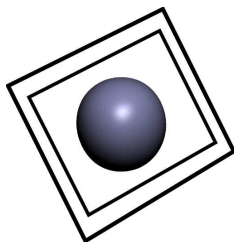


図 3.4: 視点の回転操作

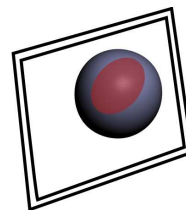
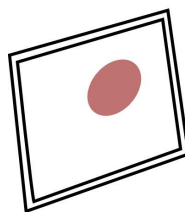


図 3.5: 原型となる球の生成操作

回転操作 回転操作を図 3.7 に示す。操作方法是以下となる。

1. 3D オブジェクトにわくこんを向ける
2. 利き手を輪に添える
3. 利き手を輪に添えたままわくこんを回転させる

わくこんを回転させる方向に 3D オブジェクトが回転される。

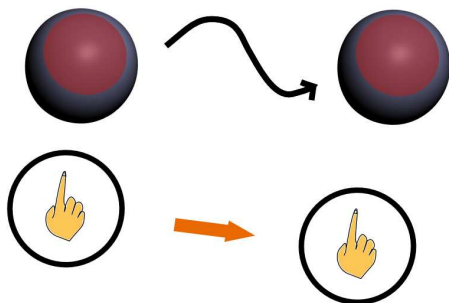


図 3.6: 移動操作

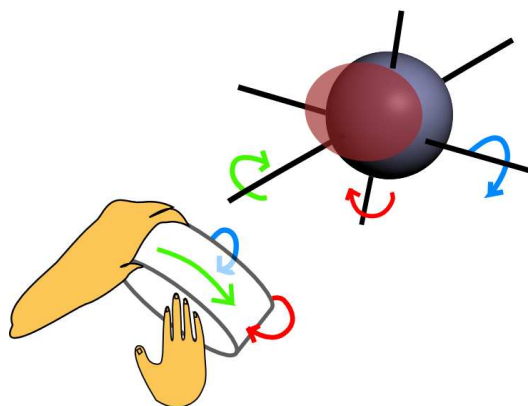


図 3.7: 回転操作

拡大・縮小操作 拡大・縮小操作を図 3.8 に示す。対象とする 3D オブジェクトにわくこんを向け、利き手を用いて輪の内側をなぞる。輪をなぞることにより 3D オブジェクトが拡大・縮小される。

掴む操作 掴む操作を図 3.9 に示す。操作方法是以下となる。

1. 3D オブジェクトの掴みたい部分にわくこんを向ける

2. 利き手を用いて掴むジェスチャを行う
3. 手を閉じた姿勢のままわくこんを移動させる

掴むジェスチャを行うことによりスカルプティングの掴む操作が行われる。

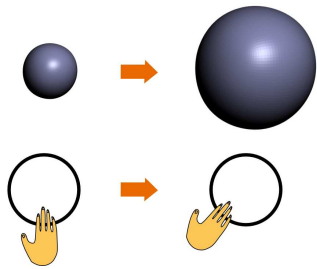


図 3.8: 拡大・縮小操作

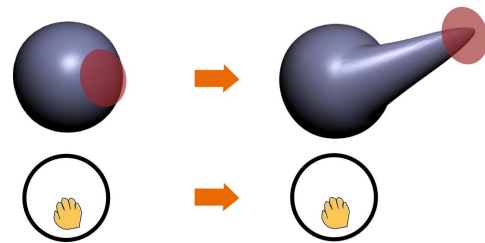


図 3.9: 掴む操作

スカルプティング領域の拡大・縮小 スカルプティング領域の拡大・縮小を図 3.10 に示す。輪を画面外に向けた状態で、輪の内側をなぞる。輪をなぞることによりスカルプティング領域が拡大・縮小される。

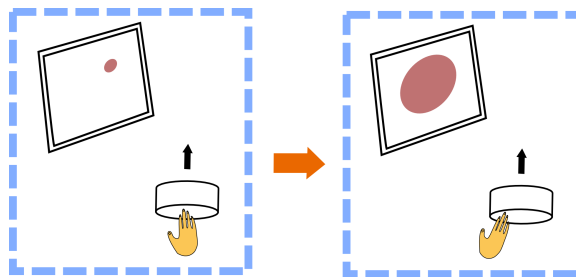


図 3.10: スカルプティング領域の拡大・縮小

第4章 輪型入力装置の試作

本章では輪型入力装置の試作について述べる。輪型入力装置には距離センサとして静電容量センサを用いたプロトタイプ1、距離センサとしてフォトリフレクタを用いたプロトタイプ2を試作した。プロトタイプ1を試作することにより、わくこんの実装に静電容量センサを使用することの妥当性を判断した。そして、プロトタイプ1の結果を受けプロトタイプ2を試作した。本章では、まずプロトタイプ1とそのセンシング特性を示す。次にプロトタイプ2とそのセンシング特性を示す。

4.1 プロトタイプ1の試作

本節では、プロトタイプ1の回路と実装について述べ、さらにプロトタイプ1のセンシング特性について述べる。

4.1.1 駆動回路

プロトタイプ1の回路図を図4.1に示す。回路図中のfoilはアルミホイルを表す。それぞれのfoilが独立した静電容量センサとして働く。静電容量センサは8個用いられる。Atmega328はArduino¹マイコンに搭載されているICを表している。

4.1.2 ハードウェア実装

回路図を基にハードウェア実装を行った。実装にはArduino、アルミホイル、布、非導電性の輪を用いた。また、Arduinoの静電容量センサライブラリとしてCapacitive Sensing Library²を用いた。試作したプロトタイプ1の概観を図4.2に示す。

アルミホイルを用いることにより静電容量のセンシング感度を上げることができる。式4.1の S をアルミホイルの面積、 d を手とアルミホイルとの距離に定める場合、センシング感度はアルミホイルの面積に比例する。

$$C = \frac{\epsilon S}{d} \quad (C \text{ は静電容量}, \epsilon \text{ は誘電率}, S \text{ は面積}, d \text{ は距離を示す}) \quad (4.1)$$

手とアルミホイルが接触することを避けるために輪の内側に布を備え付ける。式4.1に示される様に、アルミホイルと手が接触する場合、静電容量の値は発散する。そのため、布を備え付けることで静電容量の値の発散を防いでいる。

¹<http://www.arduino.cc/>

²<http://www.arduino.cc/playground/Main/CapSense>

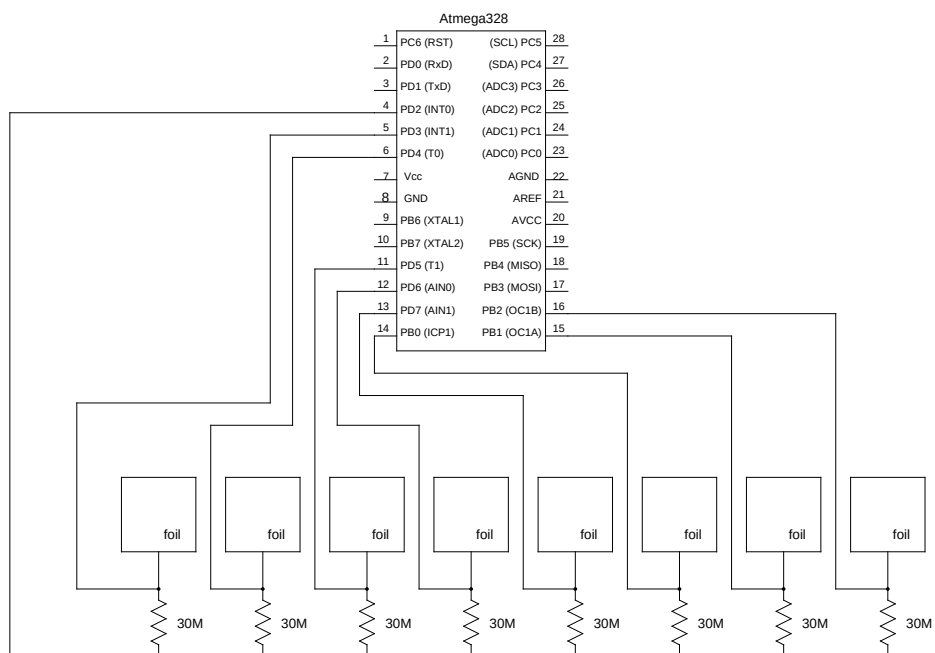


図 4.1: プロトタイプ 1 の回路図

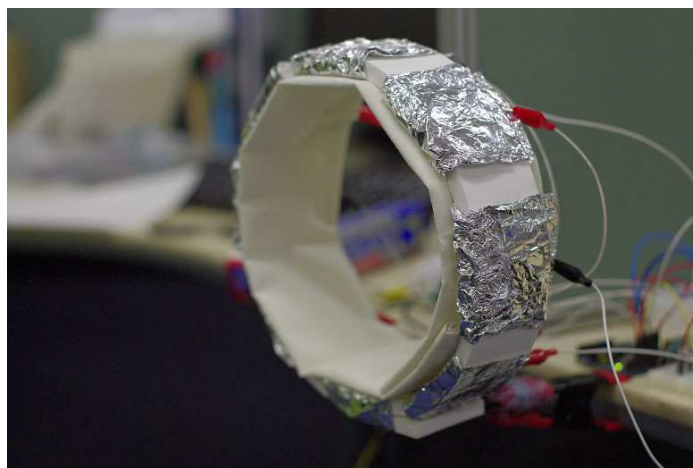


図 4.2: プロトタイプ 1 の概観

非導電性の輪を用いた理由は、輪自体が静電容量センサとなり 8 個の静電容量センサの独立性が失われることを防ぐためである。

4.1.3 プロトタイプ 1 のセンシング特性

静電容量センサを用いた実装の有効性を判断するために、手の姿勢ごとに得られるセンシング値を調べる。センシング値を調べるためのアプリケーションを実装した。このアプリケーションはセンシング値を把握しやすくするために、文字ではなくレーダーチャートとしてセンシング値を表示する。

図 4.3 に手の姿勢ごとに得られるセンシング値を示す。図 4.3a は手を入れない姿勢、図 4.3b はポインティングの姿勢、図 4.3c は手を開いた姿勢、図 4.3d は手を閉じた姿勢、図 4.3e は輪に手を添える姿勢、図 4.3f は輪の外側を掴む姿勢である。図 4.3 の a から f は実際に手を入れた写真とセンシング値のレーダーチャートを示している。

図 4.3e と図 4.3f に示されるレーダーチャートは、静電容量センサの反応が大きすぎるために画面外に出てしまった。特に、図 4.3f は反応が大きく、輪の外側を掴む手が周囲の静電容量センサに影響を与えてしまう。この様に掴む手が周囲の静電容量センサに影響を与えてしまうため、輪の外側を掴むと図 4.3a～図 4.3d のセンシング値が得られない。つまり、わくこんを把持する場合は図 4.3a～図 4.3d の姿勢を認識できない。

わくこんを用いて手の姿勢やジェスチャを認識するためには、輪の外側の手にセンサが反応してはならない。静電容量センサは距離のセンシング範囲が広く、輪の内側だけではなく輪の外側にセンサが反応する。そのため、静電容量センサの性質は、把持されて用いられるわくこんには適さないと判断した。

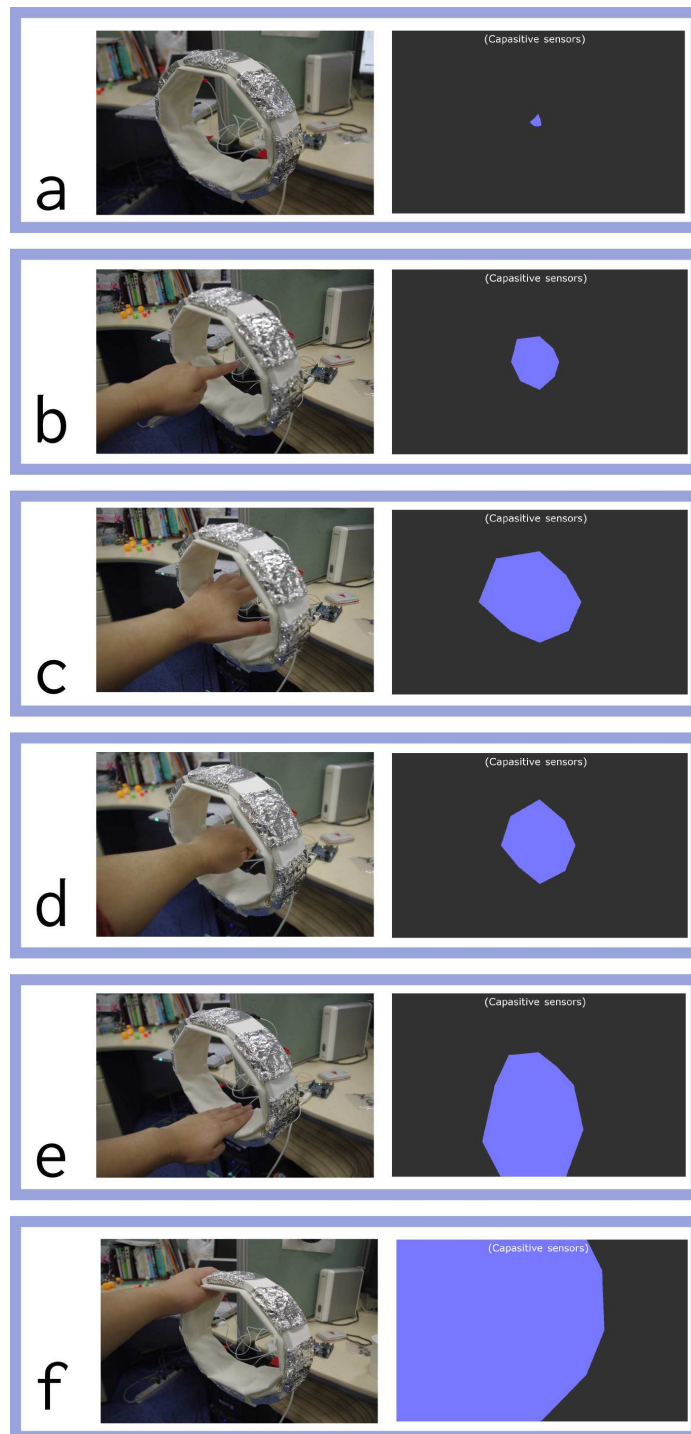


図 4.3: プロトタイプ 1 のセンシング値

4.2 プロトタイプ2の試作

プロトタイプ2では距離センサとして静電容量センサではなくフォトリフレクタを用いた。フォトリフレクタは赤外光を利用して距離を取得する。フォトリフレクタには図4.4に示す ROHM の RPR-220³を用いた。RPR-220 は 赤外線 LED と光センサから構成されている。赤外線 LED の反射光を光センサで取得することにより、距離をセンシングすることができる。RPR-220 のセンシング特性を図 4.5 に示す。プロトタイプ2 では 16 個の RPR-220 を用いた。



図 4.4: プロトタイプ2 で用いる距離センサ

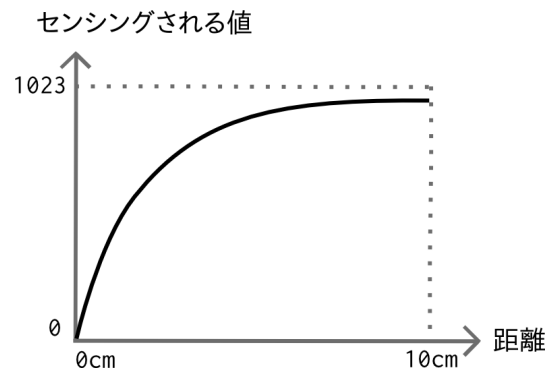


図 4.5: RPR-220 のセンシング特性

4.2.1 駆動回路

作成したプロトタイプ2の回路図を図 4.6 に示す。回路図中の LED と photoTr から成るモジュールは RPR-220 であり、4051 はマルチプレクサである。マルチプレクサを用いると、複数のアナログ入力を 1 つのアナログ入力ポートにまとめることができる。RPR-220 は 16 個用いるため、16 個のアナログ入力が必要となる。Atmega328 のアナログ入力ポートは 6 個でありアナログ入力ポートが足りない。したがって、マルチプレクサを用いる。また、トランジスタである 2SC1627 は、RPR-220 に備わる赤外線 LED のスイッチとして用いる。

4.2.2 ハードウェア実装

設計した回路図を基にハードウェア実装を行った。具体的には以下に示す作業を行った。

1. 土台となる輪の作成
2. RPR-220 の埋め込み
3. マイコンと電子部品の配線

³www.rohm.co.jp/products/databook/s/pdf/rpr-220-j.pdf

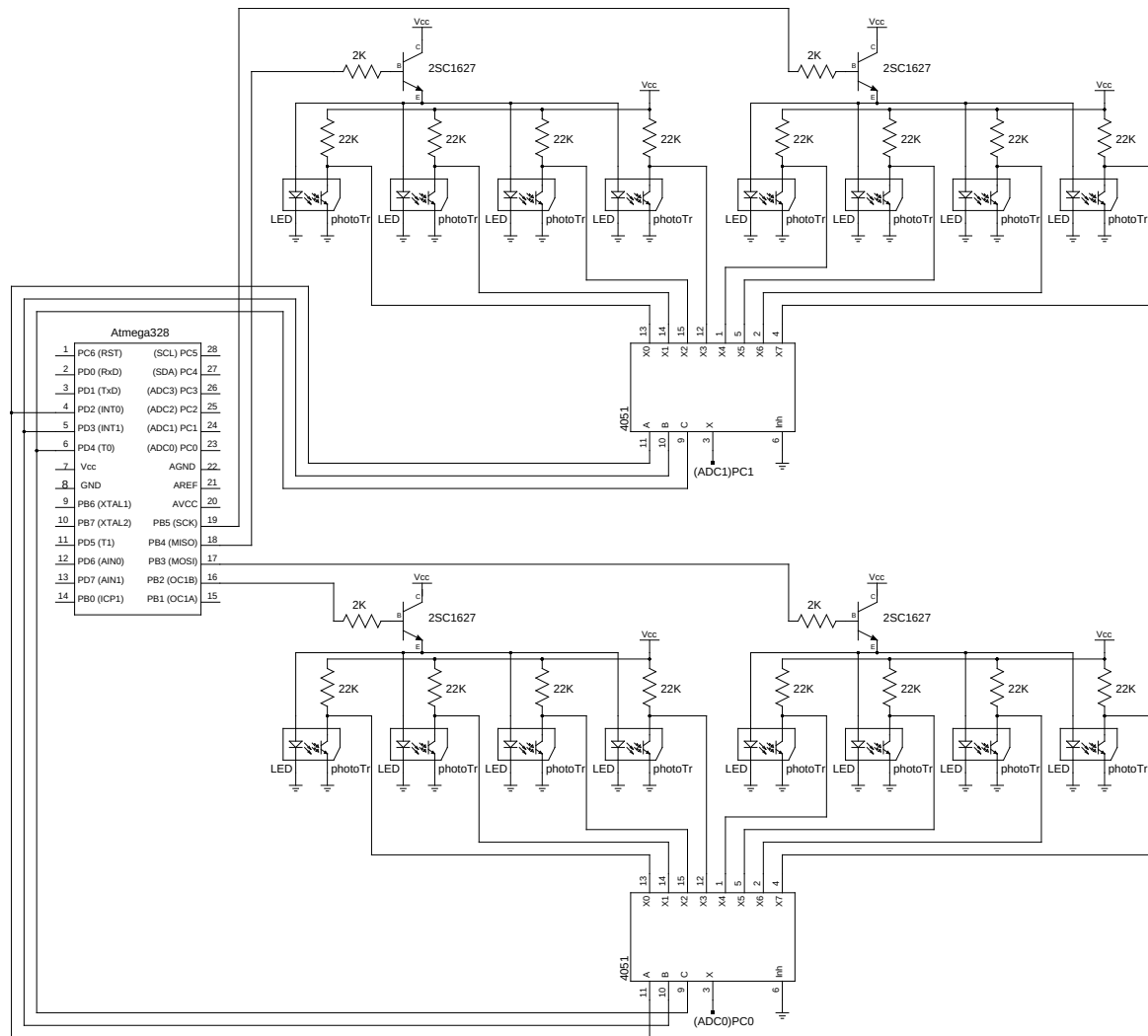


図 4.6: プロトタイプ 2 の回路図

土台となる輪の素材には紙粘土を用いた。紙粘土は加工が容易かつ一定の強度を持つので、プロトタイプ作成に適している。紙粘土を用いて整形した輪を図 4.7 に示す。この輪に RPR-220 を埋め込んだ。RPR-220 は輪の内側にセンサを向かせ等間隔に配置した。輪に埋め込まれた RPR-220 を図 4.8 に示す。



図 4.7: 紙粘土を用いて整形した輪

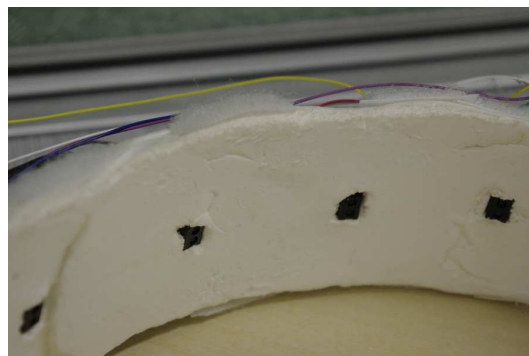


図 4.8: 輪に埋め込まれた RPR-220

RPR-220 の値を取得するためにマイコンと電子部品の配線を行う。マイコンには FIO⁴を用いた。FIO は XBee⁵ との接続が容易である。XBee を用いると計算機との無線接続を実現できる。マイコンや電子部品等を取り付け配線した後の輪を図 4.9 に示す。

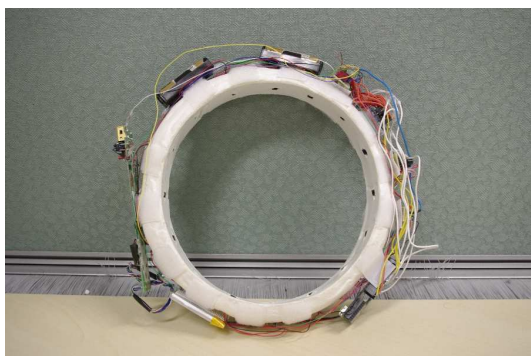


図 4.9: プロトタイプ 2 の配線後の状態

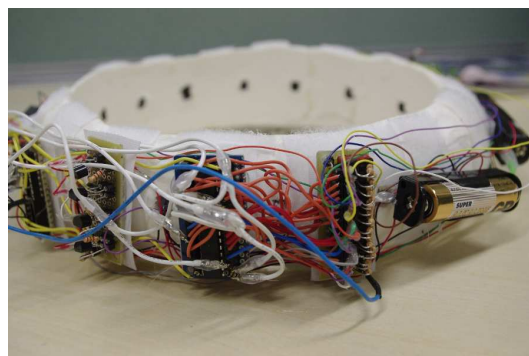


図 4.10: プロトタイプ 2 の配線を拡大した様子

配線後の輪の外周は図 4.10 に示す様に、輪を握ると配線材が断線する可能性があった。この問題を解決するため配線材を覆う様にポリプロピレンのカップを配置した。その結果を図 4.11 に示す。さらに輪を把持し易くするためにウレタンを用いて輪全体を覆った。その結果を図 4.12 に示す。

⁴<http://funnel.cc/Hardware/FIO>

⁵<http://www.digi.com/products/wireless/point-multipoint/xbec-series1-module.jsp#overview>

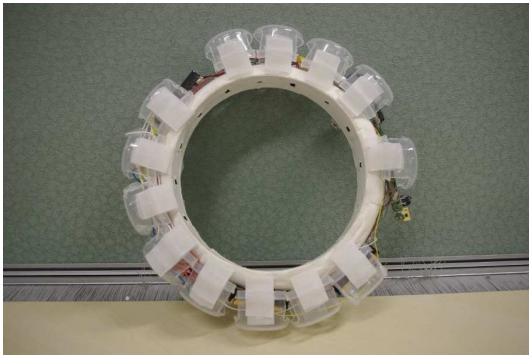


図 4.11: ポリプロピレンのカップにより覆われた様子



図 4.12: ウレタンにより覆われた様子

4.2.3 ソフトウェア実装

配線するのみでは RPP-220 のセンシング値を取得できないので、センシング値を取得するためのロジックを組む必要がある。RPR-220 のセンシング値を取得する処理を Arduino マイコンに実装する。センシング値を取得する処理の流れを図 4.13 に示す。処理は処理 a、処理 b、処理 c の 3 つから構成される。

処理 a は環境光の影響を取り除くために行われる。そのために、全ての赤外線 LED を消灯しセンシング値を得る。赤外線 LED を消灯する場合、RPR-220 の光センサは環境光のみから影響を受ける。処理 b、処理 c では光センサは環境光と赤外線 LED の影響を受ける。処理 b、処理 c のセンシング値と処理 a のセンシング値との差分を求めることで、環境光の影響を取り除くことができる。

処理 b は距離のセンシング値を取得するために行われる。処理 b では 4 つのグループごとに赤外線 LED を点灯させて 16 個のセンシング値を取得する。RPR-220 は赤外線 LED の反射光を光センサで取得し距離をセンシングするので、対面の赤外線 LED が点灯している場合正しく距離をセンシングできない。グループごとに赤外線 LED を点灯させることにより対面の赤外線 LED の影響を避ける。

処理 c は対面の影響を受けた距離のセンシング値を取得するために行われる。処理 c では全ての赤外線 LED を点灯させて 16 個のセンシング値を取得する。処理 c は処理 b と対照的に、対面の赤外線 LED の影響を利用する。対面の赤外線 LED が点灯される場合、手の姿勢によっては赤外線 LED の光がもれ特徴的なセンシング値を取得できる。つまり、処理 c を行うことで手の姿勢とジェスチャの認識に有用なセンシング値が得られる。

最終的に 32 個のセンシング値を得る。センシング値は 40msec ごとに繰り返し計算機に送信される。

4.2.4 プロトタイプ 2 のセンシング特性

プロトタイプ 1 と同様にプロトタイプ 2 の手の姿勢ごとに得られるセンシング値を調べ、プロトタイプ 2 のセンシング特性を見る。

図 4.14 に手の姿勢ごとのセンシング値を示す。図 4.14a は手を入れない姿勢、図 4.14b はポインティングの姿勢、図 4.14c は手を開いた姿勢、図 4.14d は手を閉じた姿勢、図 4.14e は輪に手を添える姿勢

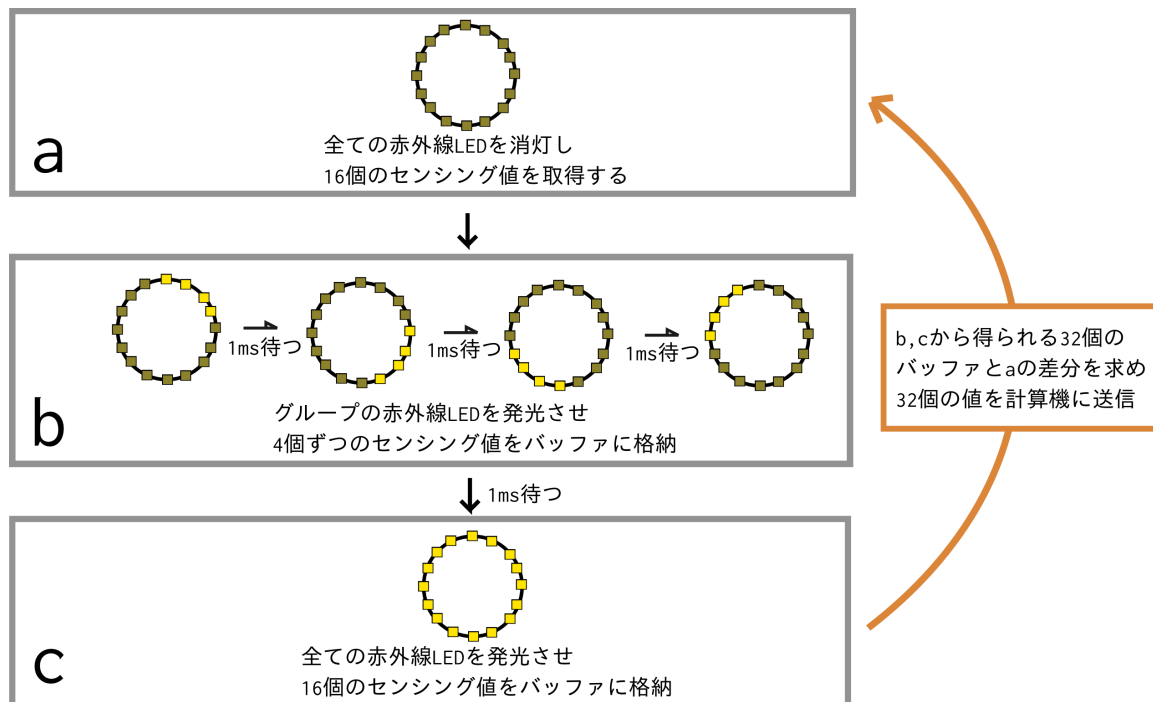


図 4.13: プロトタイプ 2 のセンシング値を取得する処理

である。アプリケーションの緑色の多角形は 4 つのグループごとに赤外線 LED を点灯させた場合のセンシング値を表し、赤色の多角形は全ての赤外線 LED を点灯させた場合のセンシング値を表す。

プロトタイプ 1 はプロトタイプ 2 と異なり、距離のセンシング範囲が狭く輪の内側の範囲内がセンシングされる。したがって、輪の外側を掴んでも距離センサに影響を与えない。この性質は把持されて用いられるわくこんに適していると判断した。

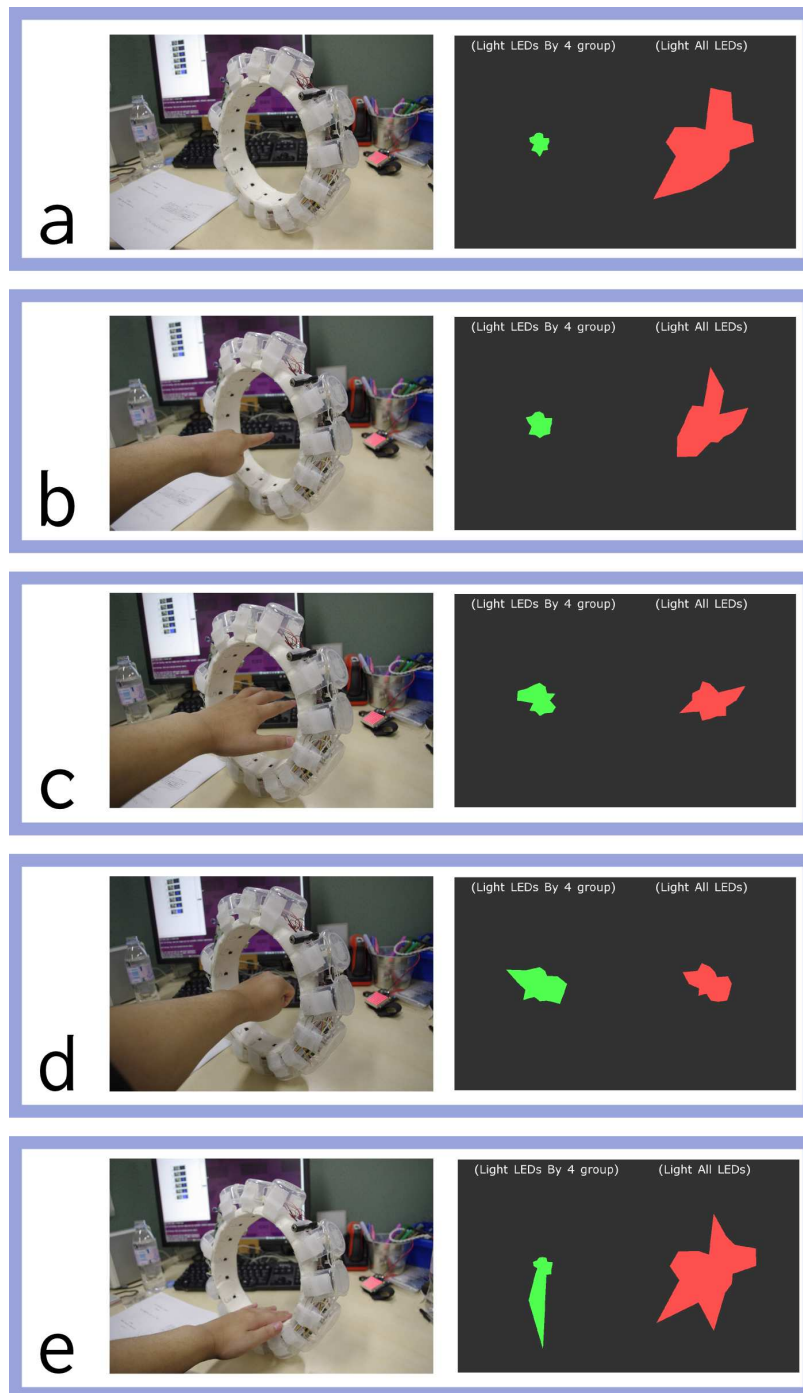


図 4.14: プロトタイプ 2 のセンシング値

4.3 回転検出部とポインティング位置検出部の実装

試作したプロトタイプ2に回転検出部とポインティング位置検出部を取り付ける。このプロトタイプでは回転検出部に加速度センサ、ポインティング位置検出部に赤外線カメラをそれぞれ用いる。加速度センサと赤外線カメラには Wii Remote⁶ に備わるものを用いる。

プロトタイプ2を用いて3次元のポインティングを実現するために、Wii Remote に備わる赤外線カメラを用いる。プロトタイプ2に取り付けられた赤外線カメラを図 4.15 に示す。Wii Remote に備わる赤外線カメラは4点の赤外線位置を検出できる。4点の内2点の赤外線位置を検出すれば、三角測量を用いて3次元のポインティング位置を検出できる。ポインティング対象であるディスプレイに2点の赤外線 LED を配置した。このディスプレイにプロトタイプ2を向けることで3次元のポインティングを行う。

プロトタイプ2を用いて3次元の回転を検出するために、Wii Remote に備わる加速度センサを用いる。Wii Remote に備わる加速度センサはx, y, z の3軸加速度をセンシングできる。この3軸加速度を用いて Pitch と Roll の2軸の回転を検出する。検出する回転方向を図 4.16 に示す。

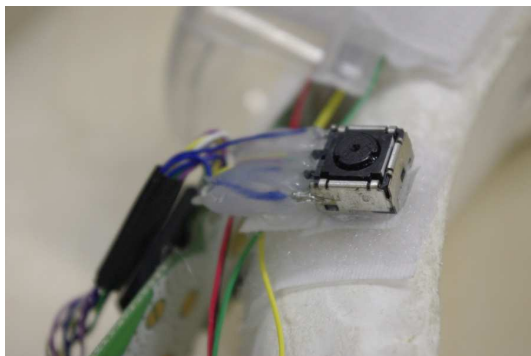


図 4.15: プロトタイプ2に取り付けられた赤外線カメラ

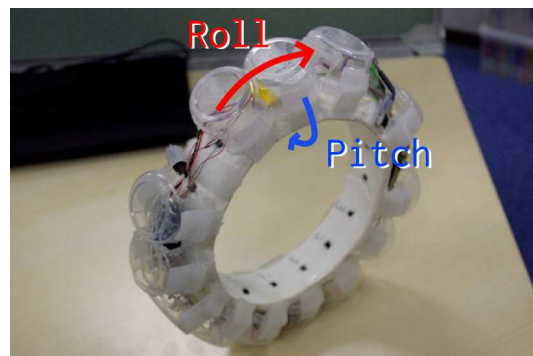


図 4.16: プロトタイプ2が検出する回転方向

⁶http://www.nintendo.co.jp/wii/features/wii_remote.html

第5章 輪型入力装置を用いた3D モデラの実装

本章では輪型入力装置を用いた3D モデラの実装について述べる。まず、手の姿勢とジェスチャを認識する認識部の実装について述べる。そして、3D モデラの実装について述べる。

5.1 認識部の実装

認識部は第4章で示したプロトタイプ2を用いて、手の姿勢とジェスチャの認識を行う。本節は手の姿勢認識の実装について述べたのち、姿勢の認識結果を利用したジェスチャ認識について述べる。

5.1.1 姿勢認識部の実装

姿勢認識部では手の大まかな姿勢を認識する。認識する手の姿勢は手を閉じた姿勢、手を開いた姿勢、ポインティングの姿勢、輪に手を添える姿勢、手を入れない姿勢の計5種である。姿勢認識にはSVMを用いた。実装にはC++とLIBSVM¹を用いた。姿勢認識はSVM学習とSVM認識から成る。

SVM 学習

SVM学習の大まかな処理を図5.1に示す。SVM学習では学習データの準備と学習モデルの作成を行う。

SVM 学習のためのハードウェア実装 図5.1bに示す様に、学習時は非利き手を用いてわくこんを把持し、利き手を用いて学習させる姿勢を作る。学習時は両手がふさがっているので、学習データを保存するトリガとして手を用いた操作を使用することができない。そこで足を用いた操作を使用する。そのために、図5.1cに示すフットスイッチをわくこんに取り付けた。このフットスイッチは学習時のみに必要であるので、容易に着脱可能にすべきであると考えた。その結果として、フットスイッチとわくこんはステレオミニプラグを用いて連結させた。

学習データの準備 わくこんに取り付けたフットスイッチを押した後に、200フレーム連続して学習データを保存する。学習データは姿勢のラベルとセンシング値から成る。1フレームに学習させるセンシング値は32個の距離センサの値である。1フレーム当たりの時間は40msecであるので、200フレームの時間は8秒となる。この8秒間手の姿勢は保持したまま、位置や角度を自由に動かした。

¹<http://www.csie.ntu.edu.tw/~cjlin/LIBSVM/>

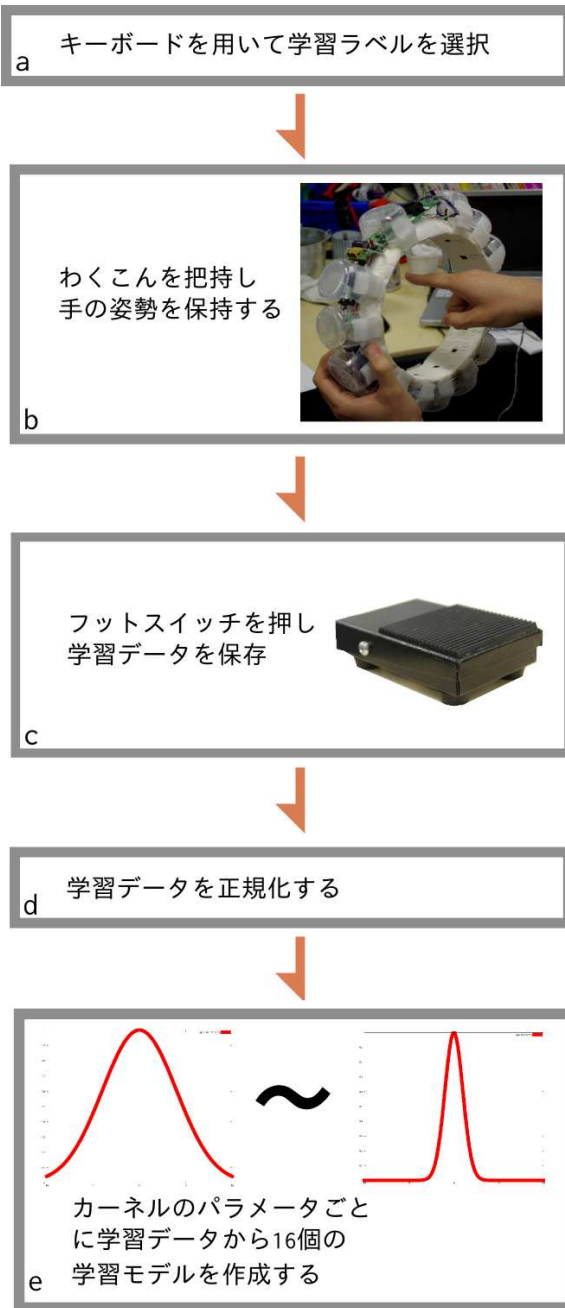


図 5.1: SVM 学習の大まかな処理

学習データの正規化 準備した学習データは、0～1023 のセンシング値である。この値を 0～1 の範囲に正規化する。

学習モデルの作成 カーネルにはガウスクーネルを用いた。ガウスクーネルにはガウス幅に関するパラメータを与える。このパラメータの初期値は 1/次元数に設定されている。つまり、わくこの場合初期値は 1/32 である。

今回、パラメータは 1/32～16/32 までの 16 個を設定し、これに対応させて学習モデルも 16 個作成した。16 個の学習モデルを作成した理由は、1 つの学習モデルを用いて SVM 認識を行うよりも、複数の学習モデルを用いて SVM 認識を行った方が有効な認識結果が得られると考えたからである。

SVM 認識

SVM 認識では SVM 学習で作成した 16 個の学習モデルを用いて、入力データを並列してラベルづける。16 個のうち最も多く認識されたラベルを認識結果として求める。入力データは 0～1023 のセンシング値であり、認識部で処理する前に 0～1 の値に正規化される。

5.1.2 ジェスチャ認識部の実装

認識するジェスチャは手を掴むジェスチャ、輪をなぞるジェスチャ、手を振るジェスチャである。

手を掴むジェスチャ 手を掴むジェスチャは姿勢認識部の結果を用いて認識する。このジェスチャ認識には 8 フレーム分の時系列的な姿勢の認識結果を用いる。8 フレームのうち前 4 フレームが開いた姿勢であり、後 4 フレームが閉じた姿勢である場合手を掴むジェスチャと認識する。

輪をなぞるジェスチャ 輪をなぞるジェスチャは掴むジェスチャと同様に、姿勢認識部の結果を用いて認識する。輪をなぞるジェスチャは 2 フレーム分の認識結果と、距離センサが最大値となる RPR-220 のインデックスを用いる。2 フレーム分の姿勢が輪に手を添える姿勢かつインデックスが変化する場合、輪をなぞるジェスチャと認識する。

手を振るジェスチャ 手を振るジェスチャの認識は FFT を用いる。FFT は 1 個の距離センサに対して時系列的に行う。具体的には距離センサのサンプルを 1 秒間つまり 25 フレーム分取得し、その時系列データに対して FFT を行い周波数領域に変換する。そして、周波数領域における最大周波数成分を求める。

FFT は 16 個の距離センサに対して並列的に行われる。最終的に求めた 16 個の最大周波数成分の平均値を求め、利き手が周期的に運動しているかどうかを認識する。実装には FFTW²を用いた。実際に取得する最大周波数成分の平均値を求めたところ、手を振るジェスチャを行わない場合は 1.0Hz～5.4Hz という結果となり、手を振るジェスチャを行う場合は 1.4Hz～7.5Hz という結果になった。この計測結果から最大周波数の平均が 6Hz 以上の場合、手を振るジェスチャであると認識する様に実装した。

²<http://www.fftw.org/>

5.2 3D モデラの実装

わくこんを用いて 3D モデリングを行うアプリケーションを実装した。実装した 3D モデラではジェスチャを用いて球を掴み変形させ、その球を配置していくことにより 3D モデリングを行う。本節では実装した 3D モデラの概観を示し、第 3 章にて述べたインタラクション手法に対する実装を示す。

5.2.1 3D モデラの概観

実装した 3D モデラの概観は球とカーソルから成る。3D モデラの概観を図 5.2 に示す。画面では掴むジェスチャを用いて球を変形させている。実際にわくこんを用いて 3D モデリングを行う様子を図 5.3 に示す。

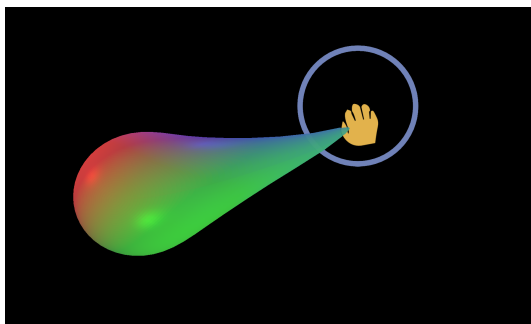


図 5.2: 3D モデラの概観

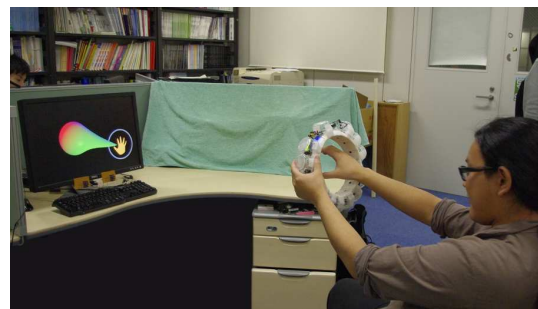


図 5.3: 実際にわくこんを用いて 3D モデリングを行う様子

変形させる球は 3D モデルのプリミティブである。球は以下のパラメータを持つ。

- 頂点集合
- 法線ベクトル集合
- 中心点
- 半径

球は $80 \times 80 = 6400$ の頂点集合から生成される。法線ベクトルは常に再計算され、変形操作を行った場合でも正しいシェーディングが保たれる。

カーソルは青い円として表示されポインティング位置を指し示す。ポインティング位置には赤外線カメラを用いて取得した 3 次元座標を用いる。3 次元座標中の (x, y) 座標はディスプレイの (x, y) 座標に対応し、 z 座標は奥行き方向に対応する。奥行き方向にカーソルが前後すると、 z 軸に沿ってカーソルが移され擬似的にカーソルの大きさが変化する。

カーソルには認識した手の姿勢も表示される。手の姿勢には SVM 認識の結果がリアルタイムに描画されるため、カーソルはユーザが手の姿勢を把握するための視覚フィードバックの役割を果たす。

スカルプティング領域はカーソルの大きさによって表現される。スカルプティング領域の拡大・縮小操作は画面外にポインティングしている状態で行われるため、画面外にポインティングしている場合

でもカーソルの大きさを分かる様にする必要がある。したがって、画面外にポインティングを行う場合は図 5.4 に示す様にカーソルを最前面の中央に表示させる。実際に行っている様子を図 5.5 に示す。

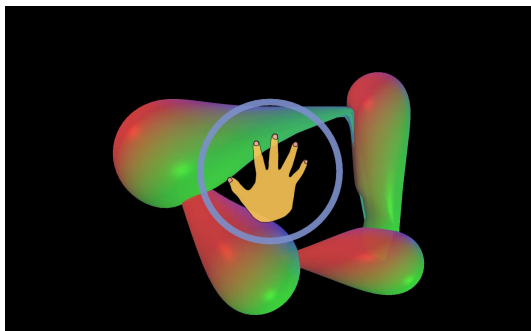


図 5.4: 画面外にポインティングする場合の表示



図 5.5: 画面外にポインティングしている様子

5.2.2 各操作の実装

3D モデラの実装には C++, openFrameworks³, Open Sound Control⁴, OpenGL⁵を用いた。以下に実装したそれぞれの操作について説明を行う。3D オブジェクトに対する操作は、カーソルと 3D オブジェクトが衝突する場合に行われる。カーソルと 3D オブジェクトは、中心間の距離が 3D オブジェクトの半径よりも小さい場合に衝突していると判定される。

原型となる球の生成 ポインティングした位置で手を振るジェスチャを行うとポインティング位置に球が生成される。生成された球は球オブジェクトのリストに動的に追加される。

移動操作 ポインティングしたカーソルの中心座標に球の中心点を移動させる。移動操作を式 5.1 に示す。式 5.1 中の $(center'_x, center'_y, center'_z)$ は移動後の球の中心点を表し、 $(cursor_x, cursor_y, cursor_z)$ はカーソルの中心座標を表す。

$$\begin{bmatrix} center'_x \\ center'_y \\ center'_z \end{bmatrix} = \begin{bmatrix} cursor_x \\ cursor_y \\ cursor_z \end{bmatrix} \quad (5.1)$$

回転操作 手を輪に添えた姿勢のままわくこんを回転させると 3D オブジェクトが回転される。3D オブジェクトの回転は式 5.2 と式 5.3 を用いて求められる。式 5.2 は x 軸周りに回転させた後、y 軸回りに回転させる行列である。式 5.3 は 3D オブジェクトの頂点を回転させる式であり、式 5.3 中の $vertices$ は 3D オブジェクトの頂点集合を表している。また、 θ はわくこんの Pitch 方向の回転、 ϕ はわくこんの Roll 方向の回転である。

³<http://www.openframeworks.cc/>

⁴<http://opensoundcontrol.org/>

⁵<http://www.opengl.org/>

$$R = \begin{bmatrix} \cos \phi & 0 & \sin \phi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \phi & 0 & \cos \phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & \sin \theta & 0 \\ 0 & -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.2)$$

$$\begin{bmatrix} v'_x \\ v'_y \\ v'_z \\ 1 \end{bmatrix} = R \times \begin{bmatrix} v_x \\ v_y \\ v_z \\ 1 \end{bmatrix} \quad (\forall v \in vertices) \quad (5.3)$$

拡大・縮小操作 わくこんの輪をなぞると 3D オブジェクトが拡大・縮小される．拡大・縮小操作は式 5.4 を用いて計算される．式 5.4 中の s は拡大率を表している．拡大率は輪を時計回りになぞる場合 1.1, 反時計回りになぞる場合 0.9 と定めた．

$$\begin{bmatrix} v'_x \\ v'_y \\ v'_z \\ 1 \end{bmatrix} = \begin{bmatrix} s & 0 & 0 & 0 \\ 0 & s & 0 & 0 \\ 0 & 0 & s & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \\ 1 \end{bmatrix} \quad (\forall v \in vertices) \quad (5.4)$$

掴む操作 この操作は掴むジェスチャにより行われる．掴む操作は領域の決定と引っ張る操作から成る．領域の決定は掴む操作を始める際に行われる．その画面を図 5.6 に示す．引っ張る操作は手を閉じた姿勢のままわくこんを動かすことにより行われる．その画面を図 5.7 に示す．

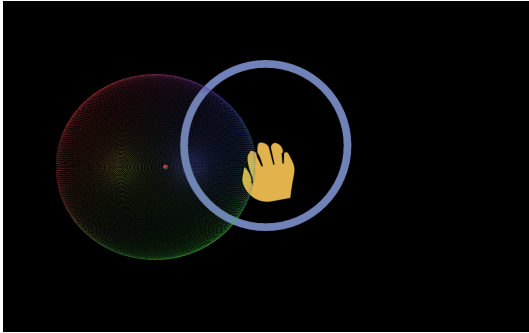


図 5.6: 領域の決定画面

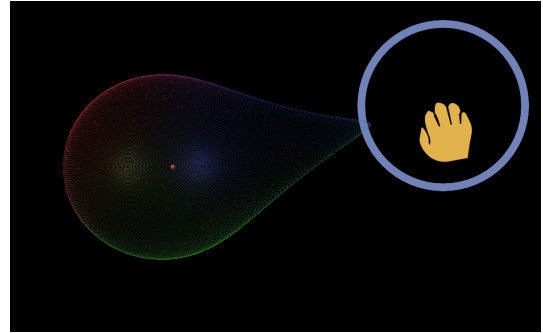


図 5.7: 引っ張る操作を行う画面

領域の決定ではカーソルの中心点と、球の全頂点との距離が計算される．距離がスカルプティング領域内に収まるものであれば、その頂点情報と頂点における重み w がリストに追加される．重み w は式 5.5 を用いて求められる． $|c - v|$ はカーソルの中心と頂点の距離を表し、 σ はスカルプティング領域の大きさを表す．

引っ張る操作では式 5.7 を用いて、引っ張られた先の頂点を計算する．式 5.7 中の (c_x, c_y, c_z) はカーソルのポインティング位置を表す．この際、式 5.6 に示されるガウス関数に引数として重み

を与え, $g(w)$ を 0.6 倍することで変形後の球が鋭く凸になることを防ぐ. 式 5.6 中の σ にはスカルプティング領域の大きさを用いる.

$$w = \frac{|c - v|}{\sigma} \quad (5.5)$$

$$g(x) = \exp\left(-\frac{x^2}{2\sigma^2}\right) \quad (5.6)$$

$$\begin{bmatrix} v'_x \\ v'_y \\ v'_z \\ 1 \end{bmatrix} = \begin{bmatrix} 0.6g(w) & 0 & 0 & 0 \\ 0 & 0.6g(w) & 0 & 0 \\ 0 & 0 & 0.6g(w) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_x - v_x \\ c_y - v_y \\ c_z - v_z \\ 1 \end{bmatrix} + \begin{bmatrix} v_x \\ v_y \\ v_z \\ 1 \end{bmatrix} \quad (\forall v \in vertices) \quad (5.7)$$

スカルプティング領域の拡大・縮小 画面外にポインティングし輪をなぞるジェスチャを行うと, スカルプティング領域が拡大・縮小される.

視点の回転操作 画面外にポインティングし, 手を輪に添えた姿勢のままわくこんを回転させると視点の回転が行われる. この操作は 3D オブジェクトの中心点の回転と 3D オブジェクトの自転から成る. 中心点の回転は式 5.8 によって求められる. 式 5.8 中の w と h は画面の幅と高さを表し, $(center_x, center_y, center_z)$ は球の中心点を表す. 式 5.8 により回転軸の平行移動, 球の中心点の回転, 回転軸を元の位置に戻すといった計算が行われる. 式 5.8 の計算に加え式 5.3 を行い視点を回転させる.

$$\begin{bmatrix} center'_x \\ center'_y \\ center'_z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & w/2 \\ 0 & 1 & 0 & h/2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} R \begin{bmatrix} 1 & 0 & 0 & -w/2 \\ 0 & 1 & 0 & -h/2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} center_x \\ center_y \\ center_z \\ 1 \end{bmatrix} \quad (\forall center \in spheres) \quad (5.8)$$

第6章 手の姿勢に関する評価実験

わくこんを用いた3Dモデリングを実現するために、手の姿勢認識は重要な役割を果たす。手の姿勢の認識結果はジェスチャの認識結果に影響を及ぼすため、ユーザビリティ全体にも影響を及ぼす。

本実験では視覚フィードバックが無い場合の手の姿勢と、視覚フィードバックが有る場合の手の姿勢に関して評価実験を行う。本実験には2つの目的がある。

目的1 視覚フィードバックが無い場合の被験者が取る姿勢とその認識率を知ること。

目的2 視覚フィードバックが有る場合の被験者が取る姿勢の認識率を知ること。

目的1を達成することで、わくこんを使用する際に被験者が取る姿勢の多様性が分かる。この情報は5.1.1節に示したSVM学習の実装に対して有益な情報である。実験によって分かる多様な姿勢は、SVMへ学習させる姿勢に対する指針となる。目的2を達成することで、現在の実装における姿勢の認識精度が分かる。わくこんを用いた3Dモデリングの操作には3Dオブジェクトの移動操作や、掴む操作など姿勢を連続的に維持させる操作がある。したがって、得られる認識精度はユーザビリティに関する指標となる。

目的1を達成するために、視覚フィードバックを与えず被験者に姿勢を指示する。指示された姿勢に対して被験者が取る姿勢を観察し、姿勢の認識率を測る。目的2を達成するために、被験者に姿勢のフィードバックを与え姿勢の認識率を測る。

被験者に示す視覚フィードバックを図6.1に示す。図6.1aは手を入れない姿勢、図6.1bは手を開いた姿勢、図6.1cは手を閉じた姿勢、図6.1dはポインティングの姿勢、図6.1eは輪に手を添える姿勢をそれぞれ表す。

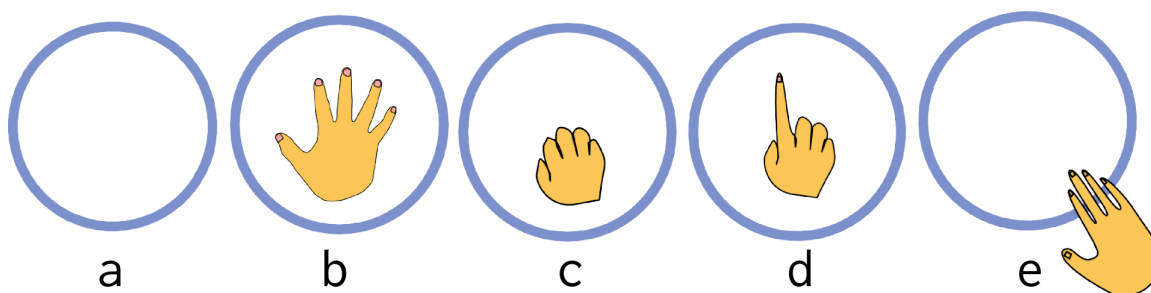


図 6.1: 被験者に示す視覚フィードバック

6.1 実験方法

実験は以下のタスクから構成される。

タスク 1 フィードバックなしの状態では被験者は5種の姿勢を取る。その姿勢は手を開いた姿勢、手を閉じた姿勢、ポインティングの姿勢、手を輪に添える姿勢、手を輪に入れない姿勢である。各姿勢に対して、被験者は姿勢を作成し終えた後5秒間保持する。被験者が姿勢を保持している間、著者は被験者が保持する手の姿勢を写真に記録する。

練習 タスク 1 とタスク 2 の間に練習を挟む。視覚フィードバックありの状態にし、被験者が5種の姿勢に対して練習を行う。

タスク 2 視覚フィードバックありの状態では被験者は5種の姿勢を取る。各姿勢に対して、被験者は姿勢を作成し終えた後その姿勢を5秒間保持する。タスク 1 と同様、被験者が姿勢を保持している間、著者は被験者が保持する手の姿勢を写真に記録する。

実験終了後、被験者は実験を通して感じたことを自由に答える。

6.2 実験結果

5名の被験者に対して実験を行った。被験者は全員が大学生か大学院生であり4名が男性で1名が女性であった。

姿勢の認識率は、式 6.1 を用いて求める。式 6.1 中の x は姿勢を認識したフレーム数を表し、125 は5秒間のフレーム数を表す。

$$\frac{x}{125} \quad (6.1)$$

6.2.1 視覚フィードバックが無い場合

視覚フィードバックが無い場合、手を開いた姿勢に対し被験者は様々な姿勢を取った。被験者が実際に取った姿勢を図 6.2 に示す。被験者 B と被験者 E はわくこんに対し垂直方向の姿勢を取った。被験者 A と被験者 D はわくこんに対し斜め方向の姿勢を取った。被験者 C はわくこんに対し水平方向の姿勢を取った。



図 6.2: 視覚フィードバックが無い場合に被験者が示した手を開いた姿勢

視覚フィードバックが無い場合の認識率を表 6.1 に示す。

表 6.1: 視覚フィードバックが無い場合の認識率

	開いた姿勢	閉じた姿勢	ポインティング	輪に添える姿勢	輪に入れない
被験者 A	100.0%	100.0%	82.4%	100.0%	100.0%
被験者 B	0.0%	100.0%	100.0%	100.0%	100.0%
被験者 C	58.4%	100.0%	0.0%	100.0%	100.0%
被験者 D	100.0%	100.0%	100.0%	100.0%	100.0%
被験者 E	0.0%	100.0%	28.0%	100.0%	100.0%

視覚フィードバックが無い場合，開いた姿勢とポインティングの姿勢の認識率が 100.0%ではない被験者がそれぞれ3名ずついた．前者の被験者 B, C, E について手を開いた姿勢の認識分布を表 6.2 に示す．後者の被験者 A, C, E についてポインティングの姿勢の認識分布を表 6.3 に示す．

表 6.2: 視覚フィードバックが無い場合の手を開いた姿勢の認識分布

	開いた姿勢	閉じた姿勢	ポインティング	輪に添える姿勢	輪に入れない
被験者 B	0.0%	100.0%	0.0%	0.0%	0.0%
被験者 C	58.4%	36.0%	5.6%	0.0%	0.0%
被験者 E	0.0%	100.0%	0.0%	0.0%	0.0%

表 6.3: 視覚フィードバックが無い場合のポインティングの姿勢の認識分布

	開いた姿勢	閉じた姿勢	ポインティング	輪に添える姿勢	輪に入れない
被験者 A	17.6%	0.0%	82.4%	0.0%	0.0%
被験者 C	42.4%	57.6%	0.0%	0.0%	0.0%
被験者 E	68.8%	0.0%	28.0%	0.0%	3.2%

6.2.2 視覚フィードバックが有る場合

視覚フィードバックが有る場合の認識率を表 6.4 に示す．

視覚フィードバックが有る場合，手を開いた姿勢の認識率が 100.0%ではない被験者が 1 名いた．また，ポインティングの姿勢の認識率が 100.0%ではない被験者が 2 名いた．

前者の被験者 C について手を開いた姿勢の認識分布を表 6.5 に示す．後者の被験者 B, E についてポインティングの姿勢の認識分布を表 6.6 に示す．

6.3 考察

視覚フィードバックが無い場合，手を開いた姿勢とポインティングの姿勢の認識率が他の姿勢と比べ低かった．視覚フィードバックが無い場合，「わくこんの奥行き方向の手の挿入位置を把握できない」

表 6.4: 視覚フィードバックが有る場合の認識率

	開いた姿勢	閉じた姿勢	ポインティング	輪に添える姿勢	輪に入れない
被験者 A	100.0%	100.0%	100.0%	100.0%	100.0%
被験者 B	100.0%	100.0%	69.6%	100.0%	100.0%
被験者 C	95.2%	100.0%	100.0%	100.0%	100.0%
被験者 D	100.0%	100.0%	100.0%	100.0%	100.0%
被験者 E	100.0%	100.0%	69.6%	100.0%	100.0%

表 6.5: 視覚フィードバックが有る場合の手を開いた姿勢の認識分布

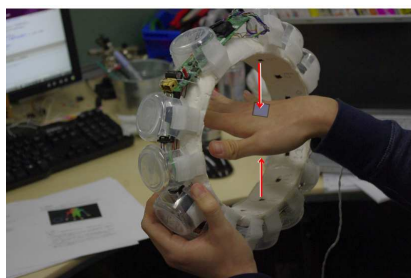
	開いた姿勢	閉じた姿勢	ポインティング	輪に添える姿勢	輪に入れない
被験者 C	95.2%	4.8%	0.0%	0.0%	0.0%

表 6.6: 視覚フィードバックが有る場合のポインティングの姿勢の認識分布

	開いた姿勢	閉じた姿勢	ポインティング	輪に添える姿勢	輪に入れない
被験者 B	30.4%	0.0%	69.6%	0.0%	0.0%
被験者 E	30.4%	0.0%	69.6%	0.0%	0.0%

という問題が生じた。この問題に関して、被験者 A は「わくの中にどの程度手を入れれば目的の姿勢となるかがフィードバックなしでは判断しづらかった」とコメントした。

この問題は手を開いた姿勢やポインティングの姿勢の認識率に影響を与える。例えば、被験者 B と被験者 E の手を開いた姿勢は 100.0% 手を閉じた姿勢であると認識された。この理由は図 6.3 に示す様に、距離センサが指先ではなく手の甲に反応したためである。距離センサが手の甲に反応すると、認識部では手を開いた姿勢ではなく手を閉じる姿勢と認識される。手を開いた姿勢と認識させるためには、センサが指の部分に反応する様に、被験者側が手の挿入位置を調整しなければならない。



被験者B



被験者E

図 6.3: 視覚フィードバックが無い場合の被験者 B, E の手を開いた姿勢

視覚フィードバックがある場合、視覚フィードバックとして表示される姿勢を取ろうとする被験者がいた。例えば、被験者 C は視覚フィードバックが無い場合、図 6.4 に示す様にわくこんに対し垂直方向のポインティング姿勢を取った。これに対し、視覚フィードバックがある場合は視覚フィードバックに表示される姿勢の様に、わくこんに対し水平方向のポインティング姿勢を取っていた。

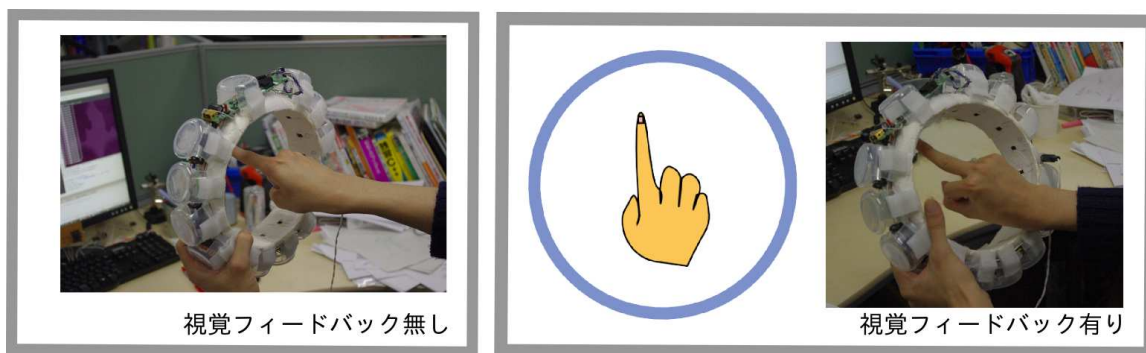


図 6.4: 被験者 C のポインティング姿勢

視覚フィードバックの有無に関わらず手を入れない姿勢、手を閉じた姿勢、輪に手を添える姿勢の認識率が 100.0%であった。100.0%となった理由には「手の挿入位置の問題」が小さいことが挙げられる。手を開いた姿勢やポインティングの姿勢は、センサが指に反応する適切な位置へ手を挿入する必要がある、この範囲は狭い。これに対し、手を閉じた姿勢は範囲が広い。また、輪に手を添える姿勢は輪に手を置くので挿入位置に関する問題が発生しない。さらに、手を入れない姿勢は問題自体が発生しない。

第7章 本システムを用いた3Dモデルの自由制作

わくこんと3Dモデラを用いた実験を行う。本実験はわくこんを用いた3Dモデリングのユーザビリティに関する問題や要求を発見するために行われる。この目的を達成するために、実際に本システムを用いて3Dモデルの自由制作を行う。

7.1 実験方法

著者1名が被験者となり本システムを用いて3Dモデルを作成する。実験の制限時間は設けない。また、実験中に発見した問題等はその都度書きとめ、必要ならばスクリーンショットを取る。

3Dモデルを作成する前に3Dモデルの制作目標を手描きした。手描きした制作目標を図7.1に示す。被験者はこの目標を適宜閲覧しながら3Dモデリングを行う。

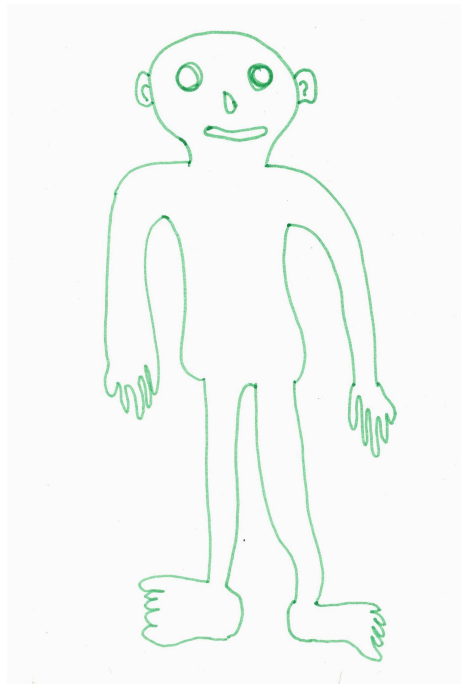


図 7.1: 手描きした制作目標

7.2 実験結果

07:47 から 08:04 までの 17 分間実験を行い 3D モデルを作成した。作成した 3D モデルを図 7.2 に示す。実験開始から 8 分後の 07:55 にわくこんを把持する左手が疲れた。その際、被験者はわくこんをデスクの上に置き、30 秒程休憩を取った後作業を再開した。

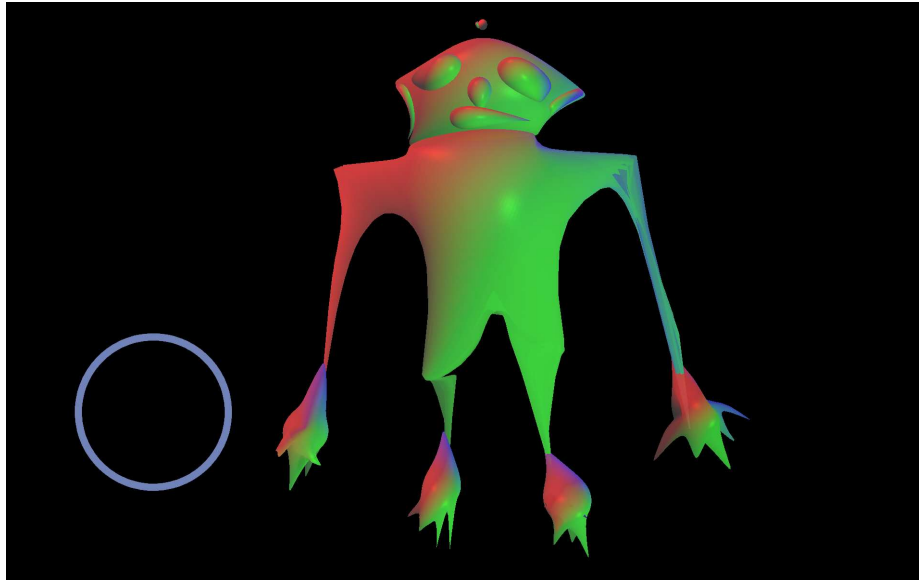


図 7.2: 実際に作成した 3D モデル

7.3 考察

実験を行うことで以下のことが明らかになった。

3次元方向の移動操作を一度にできる 実験を通して、上下左右の2次元方向に加え奥行き方向に移動可能であることを確認することが出来た。3次元方向に移動操作を行うことができたため、視点の回転操作を行うことなく手、足、顔のパーツを適切な3次元位置に配置することができた。

奥行き方向の移動操作が中断される 3次元方向の移動操作が可能であることは確認したが、同時に奥行き方向の移動操作が中断されるという問題も明らかになった。この問題は第6章にて述べた「手の挿入位置の問題」に関連する。奥行き方向の移動操作はポインティングの姿勢の挿入位置とわくこんの位置を保ったまま、前後にわくこんを動かすことで行われる。この際、ポインティングの姿勢がわくこんに対し少しでも前後に移動してしまうと、手を閉じた姿勢か手を入れられない姿勢であると誤認識される。この誤認識が原因となり、奥行き方向の移動操作が中断されてしまう。

掴む操作が楽しい 手や足を細長く引っ張る際や顔全体を変形させる際に、掴む操作が楽しいと感じた。腕の部分や足首までの部分は一つの球を変形させて作成した。この変形操作はスカルプティング

領域を小さくし、一つの球を細長く引っ張ることにより行われた。また、顔全体の形状変形は顔の各パーツを配置した後に、スカルプティング領域を大きくして行われた。顔全体の形状を変形させる際に顔の各パーツも同時に変形された。この同時変形が楽しいものであった。

カーソルの奥行き方向の位置がわからない 3D モデルのカーソルは平面的な円を用いて表現される。カーソルの奥行き方向の位置は円の大きさによって表現されるが、この表現ではカーソルの奥行き方向の位置がわからない。

この問題に付随して、3D オブジェクトとカーソルの衝突位置の把握が難しいという問題が生じた。カーソルの3次元位置を把握できない場合、3次元の衝突位置を把握することは難しくなる。衝突位置を探るために、被験者は実験中にわくこんを前後させ奥行き方向の位置を頻繁に確認した。

どちらの方向に回転操作を行っているのかわからない 視点の回転と3D オブジェクトの回転はわくこんの Pitch と Roll の回転方向に沿って行われる。Pitch か Roll のどちらか1軸に回転操作を行う際は混乱しないが、Pitch と Roll の2軸の回転を同時に行う際は回転方向への混乱が生じた。

視点の回転操作の結果として操作が難しい位置に3D オブジェクトが移動する 実験中に、3D モデルの左手部分を作成するために視点の回転操作を行った。視点の回転操作の結果として、左手部分となる球が z 軸上の手前の位置に移動した。回転操作後の画面を図7.3に示す。図7.3の右端に示される球が編集対象となる。対象がこの様に手前にある場合、掴む操作を行うことは難しい。最終的に、この位置では編集が難しいと判断し視点を元に戻した。

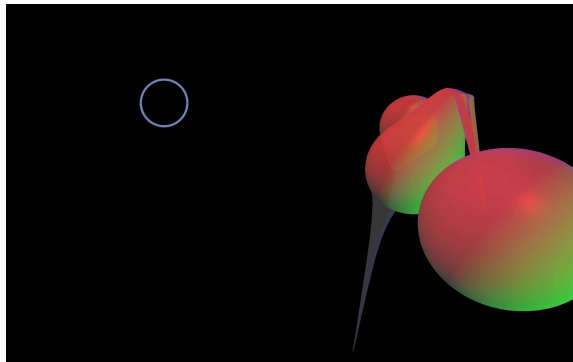


図 7.3: 視点の回転操作後の画面

視点の回転操作中にスカルプティング領域の拡大・縮小操作が行われる 視点の回転操作中に、スカルプティング領域が変更されることがあった。視点の回転は輪の一定位置に手を添え続けることによって行われる。これに対し、スカルプティング領域の拡大・縮小は輪をなぞることによって行われる。輪の一定位置に手を添え続けているつもりでも、センシング値の微妙な揺らぎから輪をなぞっているジェスチャと認識されてしまうことがあった。

カーソルが小さい場合ポインティング位置を把握しづらい スカルプティング領域を縮小させるとカーソルが小さくなる．手や足の先端部分を変形させる際は図 7.4 に示す様に，スカルプティング領域を縮小させることが多かった．図 7.4 のカーソル程小さい場合，カーソル位置の把握が難しい．

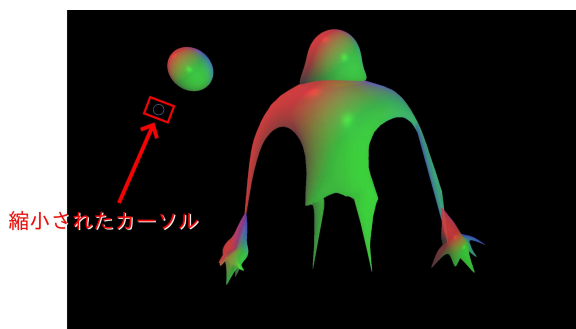


図 7.4: 足を変形させた際のカーソルの大きさ

3D オブジェクトの複製機能が欲しい 3D モデルの左手を作成し終えた後，同じ様な右手を作成することは手間であった．3D オブジェクトの複製機能がある場合，同じものを作成する負担が軽減されると考えた．

画面の縦サイズが足りない 目標となる 3D モデルは頭，胴体，腕，手，足，顔の順に作成された．足を作成する際に画面の縦サイズが足りず，本来作成したい長さの 3D モデルの足を作成することができなかった．

隅のポインティングが難しい 実験中，画面の隅のポインティングが難しいという問題が生じた．この問題はディスプレイに取り付けられた赤外線 LED とわくこんに取り付けられた赤外線カメラの位置関係が原因として考えられる．

意図せずに 3D オブジェクトの移動操作を実行してしまう 移動操作はポインティングの姿勢を取った場合に行われる．実験中，ある一瞬間ポインティングの姿勢と認識されてしまうことが多々あった．誤認識の際に，カーソルの近くに存在する 3D オブジェクトが意図しない場所へ移動されてしまった．

第8章 議論

本研究は手全体を用いて 3D モデリングを行えるようにすることを目的とした．そのためのアプローチとして，輪型入力装置とその入力装置を用いた 3D モデラの設計と実装を行った．第 6 章を通して，「手の挿入位置に関する問題」を発見した．また，第 7 章を通して 3 次元方向の移動が可能であることや，3D オブジェクトを掴む操作が可能であることが分かった．その一方で「3D モデリングのユーザビリティに関する問題」が明らかになった．本章ではこれらについて議論を行う．

8.1 手の挿入位置に関する議論

今回試作したわくこんのプロトタイプ2は手の挿入位置によって姿勢の認識結果が大きく変わる．評価実験を通して，手を開いた姿勢とポインティングの姿勢は挿入位置によって認識結果が異なることが分かった．例えば，図 8.1 に示す a, b, c は全て手を開いた姿勢であるが，それぞれ認識結果が異なる．図 8.1a の場合，手がセンサの手前にある．そのため，センサは手に反応せず手を入れない姿勢であると認識される．図 8.1b の場合，指にセンサが反応し手を開いた姿勢であると認識される．図 8.1c の場合，手の甲もしくは腕にセンサが反応し手を閉じた姿勢であると認識される．

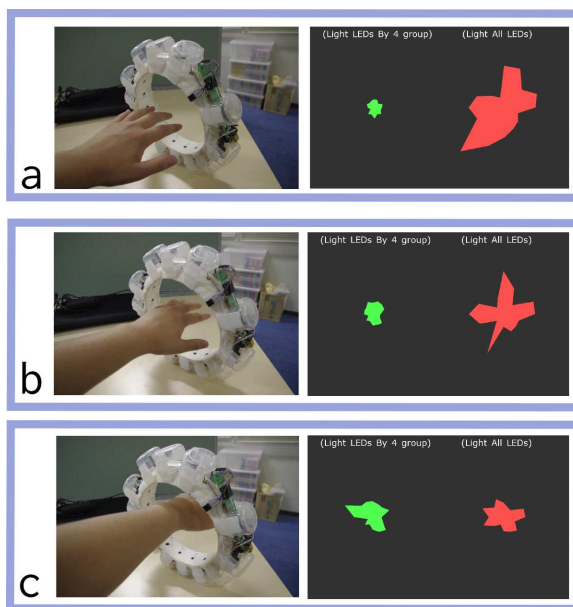


図 8.1: 手を開いた姿勢の挿入位置によるセンシング値の違い

手の挿入位置に関する問題が解決されると、幾つかのユーザビリティに関する問題が解決される。その問題には第7章の「意図せずに3Dオブジェクトの移動操作を実行してしまう」問題と「奥行き方向の移動操作が中断されてしまう」問題が挙げられる。また、挿入位置に関する問題が解決される場合、姿勢とジェスチャの認識率が向上する。したがって、手の挿入位置に関する問題の解決は掴むジェスチャの実現、さらに「手を使う要求」の実現に影響を与える。

手の挿入位置に関する問題は、以下に示す3つにより解決される可能性がある。

- 姿勢認識部の改善による解決
- ハードウェア部分の改善による解決
- 姿勢のフィードバックによる解決

本節はこれらの解決策についての議論を行う。

8.1.1 姿勢認識部の改善による解決

姿勢認識部はセンシング値を用いて手の姿勢を認識する。図8.1a, 図8.1b, 図8.1cの様に手の挿入位置によって得られるセンシング値が全く異なる場合、姿勢認識部は3つを全て異なる姿勢であると認識する。つまり、現在の距離センサを使う限り全ての挿入位置において手を開いた姿勢であると認識させることは難しい。したがって、姿勢認識部の改善では解決できないと考える。

8.1.2 ハードウェア部分の改善による解決

ハードウェア部分の改善、特に使用する距離センサを変更することにより解決を図る。プロトタイプ2では、距離センサにRPR-220を使用した。RPR-220はセンサの検出範囲が狭く、センサは平面を構築する。これに対し、静電容量センサを用いたプロトタイプ1は距離センサの検出範囲は広く、センサは3次元空間を構築する。プロトタイプ1ではその検出範囲が広過ぎるがために、把持する手を認識してしまう問題があった。距離センサの検出範囲を適切に設定することにより、図8.1aと図8.1bが手を開いた姿勢であると認識され得る。しかしながら、その場合においても図8.1cは手を閉じた姿勢であると認識されてしまうだろう。そのため、距離センサの検出範囲を適切に設定するだけでは、挿入位置に関する問題は解決できないと考えられる。つまり、RPR-220や静電容量センサの代替となる距離センサを見つけることでは問題の解決にならない。

プロトタイプ2ではRPR-220は1層の平面を構築している。この平面を2層や3層に増やし、さらに学習を行うことで挿入位置に関する問題は解決され得る。しかしながら、このアプローチでは、わくこんが厚くなり把持不可能になり得る。このアプローチを用いる場合は、輪型入力装置を把持されて用いる入力装置ではなく、据え置き型の入力装置として設計し直すことが考えられる。また、把持されて用いる入力装置に留める場合はセンサの配置を工夫する必要がある。

8.1.3 姿勢のフィードバックによる解決

姿勢のフィードバックを活用することで解決を図る。第6章にて示した様に、認識結果の視覚フィードバックを示すことで、ユーザは認識部の結果を把握し、姿勢の認識結果に合わせて手の挿入位置を調節する。結果的に、手を開いた姿勢においてその認識率は視覚フィードバックが無い状態よりも向上することを確認した。

現在のフィードバックは認識結果を視覚フィードバックとして提示するが、この提示には「図で示されている様な姿勢をユーザが取ろうとする」問題がある。

同一の姿勢でも、その姿勢は挿入角度や指の開き具合から様々な姿勢を取りうる。例えば、第6章にて示した様に、手を開いた姿勢でもユーザは様々な手を開いた姿勢を取る。現在の認識部はこの姿勢の幅を許容する様に学習されるため、ユーザはある程度自由に姿勢を取ることができる。しかしながら、現在の視覚フィードバックはユーザが取る姿勢の自由を制限する。

フィードバックとして、認識結果以外の情報もユーザに与える。例えば、認識精度に関する情報や生データを活かした情報を与える。このフィードバックは認識結果による単一の視覚表現より、姿勢に関するより豊富な情報をユーザに与え、手の挿入位置をユーザが理解する助けとなり得る。結果的に、手の挿入位置を調節する助けとなる情報をユーザに与えることで、挿入位置に関する問題が解決され得る。

8.2 3Dモデリングのユーザビリティに関する議論

第7章を通して明らかになった問題の一つに、視点の回転操作中にスカルプティング領域の拡大・縮小操作が行われる問題があった。この問題は手を輪に添え続ける操作の間に、輪をなぞるジェスチャが行われたために発生した。この問題の原因には認識部におけるジェスチャの誤認識や、輪をなぞるジェスチャと手を輪に添え続ける操作が切り分けられていないという理由が考えられる。後者の理由はシステムの設計方針に関わる問題である。

今回、モード切り替え問題に対するシステムの設計方針として、全ての操作を手の姿勢やジェスチャを用いて行うという方針を取った。全ての操作を手の姿勢やジェスチャを用いて行うと、手を用いた操作が多くなる。この際に、操作間の切り分けが不十分であると、ある操作中に別の操作が行われてしまう。

モード切り替え問題に対する設計方針には今回の方針以外の方針が考えられる。例えば、ジェスチャと音、ジェスチャと視線などのマルチモーダルな設計方針が考えられる。マルチモーダルな設計方針は、操作のパターンが増え、ユーザが実行し得る操作が増えてしまう問題が発生する。しかし、操作の切り分けが容易になるという利点がある。

第9章 結論

本研究では輪型入力装置である「わくこん」とわくこんを用いた 3D モデラの設計と実装を行った。手の姿勢に関する評価実験と本システムを用いた 3D モデルの自由制作を通して、わくこんを用いて 3 次元方向の移動操作が可能であることや、3D オブジェクトを掴む操作が可能であることを示した。一方で、3D モデラのユーザビリティに関する様々な問題や要求を発見し、これらについての議論を行うことで本システムのさらなる展望を示した。

今後の課題として、本研究を通して分かった「手の挿入位置に関する問題」を、姿勢のフィードバックの実現等により解決することで「手を使う要求」をより満たすシステムへの発展を考えている。また、3D モデラのユーザビリティに関する知見を基に、システムの設計や実装の改良を検討している。

謝辞

本論文の執筆に当たって，志築文太郎先生，三末和男先生，田中二郎先生，高橋伸先生には丁寧なご指導と貴重なご意見を頂きました。心より感謝致します。また，志築文太郎先生には日頃からきめ細かいご指導を頂き，大変お世話になりました。厚くお礼申し上げます。田中研究室のメンバーには研究生生活を通じ多くのアドバイスを頂き，さらに，研究以外の部分でも大変お世話になりました。また，WAVE チームの皆様にはチームゼミで多くのご意見を頂きました。ここに深く感謝致します。

最後に，これまで自分を支えてくれた家族や友人に感謝致します。ありがとうございました。

参考文献

- [BJKP05] Doug A. Bowman, Joseph J. LaViola Jr., Ernst Kruijff, and Ivan Poupyrev. 3D ユーザインタフェース. 丸善, 2005.
- [GK01] Mark D. Gross and Ariel J. Kemp. Gesture Modelling : Using Video to Capture Freehand Modeling Commands. In *Computer Aided Architectural Design Futures 2001: proceedings of the Ninth International Conference*, pp. 271–284, 2001.
- [LHS06] Chia-Hsun Jackie Lee, Yuchang Hu, and Ted Selker. iSphere: A Free-Hand 3D Modeling Interface. *International Journal of Architectural Computing*, Vol. 4, No. 1, pp. 19–31, 2006.
- [MN94] Tamotsu Murakami and Naomasa Nakajima. Direct and Intuitive Input Device for 3-D Shape Deformation. In *Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence*, CHI '94, pp. 465–470, 1994.
- [Nag10] Yoichi Nagashima. Untouchable Instrument “Peller-Min”. In *Proceedings of the 2010 conference on New Interfaces for Musical Expression*, NIME '10, pp. 387–390, 2010.
- [NUK98] Hiroaki Nishino, Kouichi Utsumiya, and Kazuyoshi Korida. 3D Object Modeling Using Spatial and Pictographic Gestures. In *Proceedings of the 2008 ACM symposium on Virtual Reality Software and Technology*, VRST '98, pp. 51–58, 1998.
- [SPS01] Steven Schkolne, Michael Pruett, and Peter Schröder. Surface Drawing: Creating Organic 3D Shapes with the Hand and Tangible Tools. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '01, pp. 261–268, 2001.
- [SSO04] Masatake Sato, Vladimir Savchenko, and Ryutarou Ohbuchi. 3D Freeform Design: Interactive Shape Deformations by the Use of Cyberglove. In *Proceedings of the 2004 International Conference on Cyberworlds*, CW '04, pp. 147–154, 2004.
- [STP08] Ross T. Smith, Bruce H. Thomas, and Wayne Piekarski. Digital Foam Interaction Techniques for 3D Modeling. In *Proceedings of the 2008 ACM symposium on Virtual Reality Software and Technology*, VRST '08, pp. 61–68, 2008.
- [横川 07] 横川健, 石井雅博, 唐政, 山下和也. ジェスチャインタフェースを用いた 3 次元モデリング手法の提案. *グラフィクスと CAD 研究会報告*, Vol. 2007, No. 70, pp. 41–45, 2007.