

平成 22 年度

筑波大学情報学群情報科学類

卒業研究論文

題目

伸縮性を持つ紐を用いた入力インタフェース

主専攻 知能情報メディア主専攻

著者 吉川 拓人

指導教員 志築 文太郎 高橋 伸 田中 二郎 三末 和男

要 旨

道具を入力インタフェースとして用いる研究が行われている。道具に対して行う操作と同じ操作によってインタラクションを行うことができるため、すでに学習している効率的な道具の使い方をインタラクションに活用できる。

また、弾性を入力インタフェースとして用いる研究が行われている。ユーザはコンピュータに対する操作感覚を弾性による力覚フィードバックや触覚フィードバックを介して得ることができる。また、弾性体は剛体よりも柔らかい触感を持つため、ユーザに対して触ることへの興味を引きやすい。

本研究では弾性体のうちの1つである伸縮性を持つ紐を道具として用いた入力インタフェースを開発した。ユーザは、2つの姿勢と5つの操作によってコンピュータ操作を行うことができる。伸縮性を持つ紐にマーカを取り付けることで、伸縮性を持つ紐の状態を推定する。本インタフェースを用いたアプリケーションを開発した。また、操作の認識精度と手振れについて評価を行った。

目次

第1章	序論	1
1.1	背景	1
1.2	伸縮性を持つ紐	2
1.3	目的	3
1.4	本論文の構成	3
第2章	関連研究	5
2.1	道具をインタラクションに用いる研究	5
2.2	弾性をインタラクションに用いる研究	5
2.3	紐を入力インタフェースとする研究	6
2.4	紐をメタファとして用いる研究	6
第3章	ElasticCord: 伸縮性を持つ紐を用いた入力インタフェース	7
3.1	ElasticCord において活用されるアフォーダンス	7
3.2	ElasticCord における2つの姿勢	8
3.3	姿勢によるインタラクション手法の変化	9
3.4	ElasticCord における5つの操作	9
3.4.1	伸縮性を活用した2つの操作	9
3.4.2	伸縮性を持つ紐の動きを用いた3つの操作	10
3.5	ElasticCord の応用場面	11
第4章	実装	12
4.1	システム構成	12
4.2	ソフトウェア概要	13
4.3	認識部	14
4.3.1	HSV 表色系におけるマスキング	16
4.3.2	ノイズ除去、ラベリングおよび重心の決定	17
4.3.3	三角測量による座標計測	17
4.3.4	状態推定部へのデータ送信	18
4.4	状態推定部	18
4.4.1	座標変換	19
4.4.2	ポインティング位置の決定	19

4.4.3	伸縮性を持つ紐の方位と傾斜	20
4.4.4	ElasticCord における 2 つの姿勢の認識	20
4.4.5	伸縮性を活用した 2 つの操作の認識	22
4.4.6	伸縮性を持つ紐の動きを用いた 3 つの操作の認識	22
4.4.7	重心移動量差分の計算	23
4.4.8	応用部へのデータ送信	23
第 5 章	アプリケーション	25
5.1	ポインティングアプリケーション	25
5.2	音楽プレーヤ	27
第 6 章	評価実験	30
6.1	実験目的	30
6.2	実験環境	30
6.3	実験内容	30
6.4	実験結果	31
第 7 章	議論	33
7.1	操作感	33
7.2	伸縮性を持つ紐を伸ばす際の手振れ	34
7.3	弛みの認識とそれを用いた操作の実現	34
7.4	離れた場所からのインタラクション	34
7.5	複数の伸縮性を持つ紐の利用	35
7.6	マーカ認識置ける精度の向上	35
7.7	伸縮性を持つ紐の動きを用いた操作の認識方法	35
7.8	ポータビリティ	35
第 8 章	結論と今後の課題	36
	謝辞	37
	参考文献	38

目次

1.1	伸縮性を持つ紐が伸張していない状態	2
1.2	伸縮性を持つ紐が伸張している状態	2
1.3	指向性	4
3.1	指すアフォーダンス（左）と伸ばすアフォーダンス（右）	7
3.2	ポインティング姿勢（左）と非ポインティング姿勢（右）	8
3.3	ElasticCord における姿勢と操作	11
4.1	伸縮性を持つ紐	13
4.2	マーカを取り付けた紐	13
4.3	ハードウェアを設置した様子	13
4.4	認識部における処理の流れ	15
4.5	左カメラから取得した画像	15
4.6	右カメラから取得したカメラ画像	15
4.7	左カメラによる画像から抽出されたマーカ領域	16
4.8	右カメラによる画像から抽出されたマーカ領域	16
4.9	抽出された重心	17
4.10	三角測量の際の座標系	18
4.11	座標系の変換	20
4.12	操作認識における概念図	21
5.1	ポインティングアプリケーションを使用している様子	26
5.2	ポインティングアプリケーション使用中の画面	26
5.3	音楽プレーヤを使用している様子	28
5.4	音楽プレーヤ初期状態	28
5.5	Stretch 操作によりアルバムを選択している状態	28
5.6	アルバムを選択した状態	29
5.7	非ポインティング姿勢における表示	29
6.1	実験環境	31
6.2	タスク実行画面	31

第1章 序論

本章では、まず研究の背景と目的を述べる。次に、本研究において着目する伸縮性を持つ紐について説明する。最後に本論文の構成を述べる。

1.1 背景

道具¹を入力インタフェースとして用いる研究が行われている。人間は、道具の力を借りることによって、身体のみでは行えないような操作を行うことができる。また、その道具が自身の身体の一部のように感じられることがある。猿を道具を使えるように訓練すると、猿は道具を身体化し、自身の体の一部のように用いることが可能になるとも報告されている [MI04]。道具を入力インタフェースとして用いることで、より効率的にインタラクションを行える可能性がある。なぜなら、この場合にもユーザは道具という入力インタフェースを身体化することによって繊細にコントロールできるようになるからである。また、道具に対して行うのと同じ操作によってインタラクションを行うことができるため、すでに学習している効率的な道具の使い方をインタラクションに活用できる。

また、弾性をインタフェースに用いる研究が行われている。人間は「押し込む」「伸ばす」「曲げる」などの操作を普段の生活において、物体に対して度々行う。例えば、本に対して曲げる操作を行うことや、やわらかいボールに対して握る操作を行うことである。弾性をを用いることによって、ユーザはこれらの操作をインタフェースに対しても行うことができるようになる。また、ユーザが行うことのできるインタラクションを多様化することができる。弾性体はユーザの操作に対して反発力を返すため、ユーザは触覚フィードバックや力覚フィードバックを受ける。ユーザは、これらのフィードバックを介して、操作感を得ることができる。また、弾性体は剛体よりも柔らかい触感を持つため、ユーザにより触ることをアフォードする。そのゆえ、ユーザの興味を引きやすいという特徴がある。

本研究は弾性体のうちの1つである伸縮性を持つ紐を入力インタフェースとして用いる。ここでの、伸縮性を持つ紐とは、紐の素材自身の弾性により、元の長さへ戻ろうとする性質を持つ紐のことを言う。本研究は、ユーザが伸縮性を持つ紐を伸ばした量、すなわち伸縮率と伸縮性を持つ紐の位置、伸縮性を持つ紐の両端を通る直線の方角をコンピュータ操作に用いる。伸縮性を持つ紐は、今までコンピュータとのインタラクションに用いられていなかった新たな道具である。また、紐とその伸縮性を組み合わせたインタラクション手法は全く開発

¹本論文における道具とは、身体から分離可能であり、身体によって操作され、かつ電氣的なデバイスが取り付けられていない物体を言う。

されていない。

本研究の貢献は、弾性を用いるインタフェースとして、伸縮性を持つ紐を用いる入力インタフェースを新たに示したことである。このことにより、弾性を用いるインタフェースの多様性を増加させ、弾性を用いた新たなインタラクションを発見する機会を生み出すことが考えられる。

1.2 伸縮性を持つ紐

伸縮性を持つ紐について説明する。図 1.1 に伸縮性を持つ紐の例を示す。伸縮性を持つ紐は、図 1.1 に示すような伸張していない状態から、図 1.2 に示すような伸張した状態へと伸ばすことができる性質を持つ。また、伸縮性を持つ紐が伸張した状態において、手を離れたとき、伸張していない状態へ戻ろうとする性質を持つ。この2つの性質を満たす紐を、本論文では伸縮性を持つ紐と呼ぶ。次に、伸縮性を持つ紐の特徴について述べる。

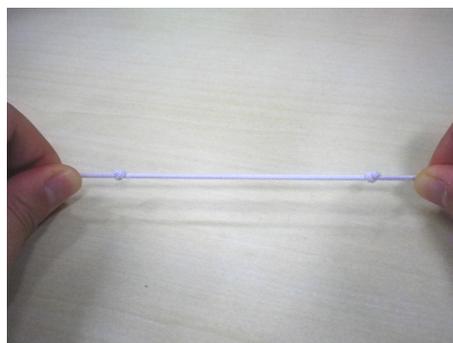


図 1.1: 伸縮性を持つ紐が伸張していない状態 図 1.2: 伸縮性を持つ紐が伸張している状態

伸縮性を持つ紐の特徴

伸縮性を持つ紐の特徴として、アフォーダンス、力覚フィードバック、指向性、そしてポータビリティが挙げられる。それぞれについて、次に詳しく述べる。

多様なアフォーダンス

伸縮性を持つ紐に対して様々な操作が可能である。例えば、伸縮性を持つ紐を「伸ばす」操作、「ねじる」操作、「結ぶ」操作や「巻く」操作が考えられる。また、弛みのないように張った伸縮性を持つ紐の両端を通る直線を用いて「指す」操作といった操作も可能である。そのため、伸縮性を持つ紐が多様なアフォーダンスを持つ。

このような性質をコンピュータ操作に用いることによって、ユーザは伸縮性を持つ紐によって行うことのできるコンピュータ操作を理解しやすくなる可能性がある。

力覚フィードバック

我々は伸縮性を持つ紐を伸ばした際、伸縮性を持つ紐からその伸縮率に対応した力覚

フィードバックを持った手を通して受け取る。この力覚フィードバックによって、我々は伸縮性を持つ紐を見ることなしにその伸縮率を感覚によって理解することができる。このような性質をコンピュータ操作に用いることによって、ある連続量を伸縮性を持つ紐の伸縮率によって操作することが可能である。その場合、ユーザは力覚フィードバックによって、その連続量の値を感じることができる可能性がある。そのため、ユーザは伸縮性を持つ紐を見ることなしにその連続量を調整できる可能性がある。

指向性

紐には指向性がある。その概念図を図 1.3 に示す。我々は、伸縮性を持つ紐を様々な方向へ伸ばすことができる。例えば、ユーザは伸縮性を持つ紐の一方の端を基準点として、もう一方の端を「右に伸ばす」、「左に伸ばす」や「右手前に伸ばす」、「左奥に伸ばす」などの操作を行うことができる。

この性質をコンピュータ操作のパラメータとして用いることができる。すなわち、伸縮性を持つ紐を伸ばす際、どの方向に伸びたかをインタラクションに用いることができる。

この性質を活用することによって、紐を用いて正確に物を指すことができる可能性がある。人間は、目を閉じている場合にも身体によって大まかな方向を指すことができる。また、目を開いている場合にはより正確に指すことが可能である。これは人間が持つ身体性による効果である。そのため、道具を身体化することができれば、道具によっても物を正確に指すことができる可能性がある。

このような性質をコンピュータ操作におけるポインティング操作に活用することができる。

ポータビリティ

伸縮性を持つ紐は小さく折りたたむことができる。そのため、かさ張らず簡単に持ち運ぶことができる。この性質のため、外出先などにおいても紐によるインタラクションを行うことが容易になる。

1.3 目的

本研究の目的は、伸縮性を持つ紐を持ったアフォーダンス、力覚フィードバック、指向性、およびポータビリティという特徴を活用したインタラクション手法を提案し、その手法を用いた入力インタフェースを開発することである。

1.4 本論文の構成

1 章では研究の背景と目的について述べた。また、伸縮性を持つ紐について述べた。2 章では、関連研究について述べる。3 章では、伸縮性を持つ紐を用いたインタフェースである

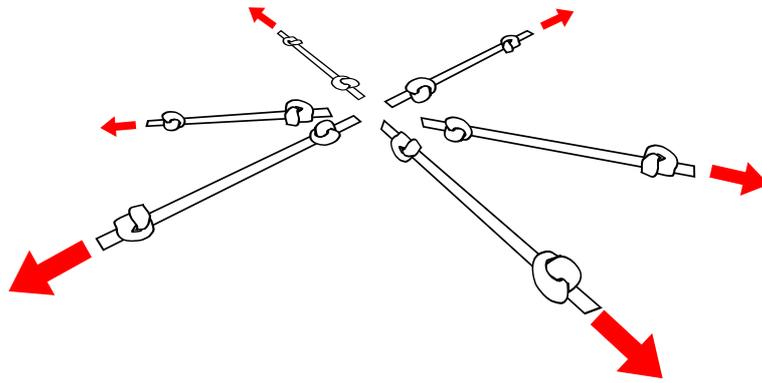


図 1.3: 指向性

ElasticCord とそのインタラクション手法について述べる。4 章では、ElasticCord の実装について述べる。5 章では、ElasticCord を用いたアプリケーションについて述べる。6 章では、評価実験について述べる。7 章では、本研究に基づいて議論を行う。そして、8 章では、本論文の結論と今後の課題について述べる。

第2章 関連研究

本章ではまず、道具をインタラクションに用いる研究について述べる。次に、弾性をインタラクションに用いる研究について述べる。最後に、紐を入力インタフェースとする研究、紐をメタファとして用いる研究について述べる。

2.1 道具をインタラクションに用いる研究

池田らは、スポイトやピンセットによってコンピュータ操作を行うデバイスを開発した [池田 02]。Cao らの VisionWand [CB03] は棒という道具を用いることによって大画面操作を行うシステムである。Lee らの HandSCAPE [LSRI00] は、メジャーによって測定した物体の長さをインタラクションに用いるものである。

本研究は道具として伸縮性を持つ紐を用いる点で、これらと異なる。道具はそれぞれ機能が異なっているため、道具の違いによってインタラクション手法も異なる。

2.2 弾性をインタラクションに用いる研究

Murakami らは、弾性を持つ立方体に対し、押す操作を加えることによって 3D オブジェクトを編集する入力デバイスを示した [MHON95]。Schwesig らの Gummi [SPM04] は、フレキシブルディスプレイに曲げセンサを取りつけることによって、面への曲げ操作をインタラクションに組み込むことを可能にした。また、Herkenrath らの TWEND [HKB08] は面に対する多様な曲げ方を認識する入力インタフェースである。Ye らの Cobra [YK10] は携帯ゲーム環境において、フレキシブルディスプレイに対する曲げ操作をゲーム操作に用いる。Sato らの PhotoelasticTouch [SMKF09] は、弾性体からなる透明なタンジブルオブジェクトを用いるテーブルトップシステムである。Vlack らの GelForce [VMK⁺05] は、弾性体の中にマーカのアレイを埋め込むことによって、弾性体面に加えられた 3 次元の力ベクトルを計測することができる。Kakehi らの ForceTile [KJS⁺08] も、弾性体の中にマーカのアレイを埋め込むことによって、弾性体面に加えられた 3 次元の力ベクトルを計測する。GelForce との違いは、ForceTile はテーブルトップ上のタンジブルオブジェクトとして用いられる点にある。Balakrishnan らは、テープ状デバイスである ShapeTape を開発した [BFKS99]。ShapeTape はテープへの曲げと捻じりを検出することが可能である。また、テープの形状を 3 次元的に復元できる。

本研究は弾性を用いる点でこれらの研究と同様である。しかし、伸縮性を持つ紐における伸縮性に着目した点で異なる。

ShapeTape は本研究において用いる伸縮性を持つ紐と形状が似通っている。ShapeTape を 3D、2D モデリング [GBS03] や、音楽の演奏 [She04] に応用する研究がなされている。ShapeTape に対して、「伸ばす」操作を行うことはできない。ShapeTape が曲げとねじりを検出する一方、本研究は伸縮性に着目する点でやはり異なっている。

2.3 紐を入力インタフェースとする研究

Schwarz らは、紐を用いた入力インタフェースである Cord Input を開発した。[SHHM10] Cord Input は紐に取りつけたデバイスを用いて、紐に対する捩じり、引っ張りを認識することが可能である。また、紐上におけるタッチ位置を検出可能である。Blasko らは、モバイル機器にメジャーのように収納できる紐を取りつけ、その紐の伸びた量と方向を用いて、機器とのインタラクションを可能にする手法を示した [BNF06]。また、Koch らは、紐をインタラクションに用いるウェアラブル機器を開発し、その機器を用いた 3 次元空間ナビゲーション手法を示した [KW08]。

本研究は、紐を用いる点ではこれらと類似しているが、伸縮性を持った紐を用いている点でこれらと異なる。また、これらの研究では、紐は一方の端が固定されていた。本研究は、両端が自由な紐を用いてインタラクションを行うこと点で異なっている。

Caglioti らは、紐の形状を写真から 3 次的に復元することによって、3D モデルを入力する手法を示した [CGMT08]。しかし、紐の形状をリアルタイムに復元することはできないため、応用は 3D モデルの入力に限られる。

本研究は紐を用いる点ではこの研究と類似しているが、紐を 3D モデルの入力ではなく、より汎用的なコンピュータ操作に用いる点で異なる。

2.4 紐をメタファとして用いる研究

佐藤らは紐を操作のメタファとした GUI 環境におけるシーケンサーを開発した [佐藤 10]。このシーケンサーは、切ることによって曲を切断することや、繋ぐことによって曲を繋ぐことができる。シーケンサーをループ状に繋ぐことによって、ループ再生を行うことができる。

本研究は、実世界の紐を用いる点で異なる。

第3章 ElasticCord: 伸縮性を持つ紐を用いた入力インタフェース

本章では伸縮性を持つ紐を用いた入力インタフェースである ElasticCord について述べる。ElasticCord において活用されるアフォーダンスについて述べた後、ElasticCord における姿勢と操作について述べる。

3.1 ElasticCord において活用されるアフォーダンス

ElasticCord は、伸縮性を持つ紐における指すアフォーダンスと伸ばすアフォーダンスを活用する。2つのアフォーダンスの概念図を図3.1に示す。それぞれのアフォーダンスについて説明を述べる。

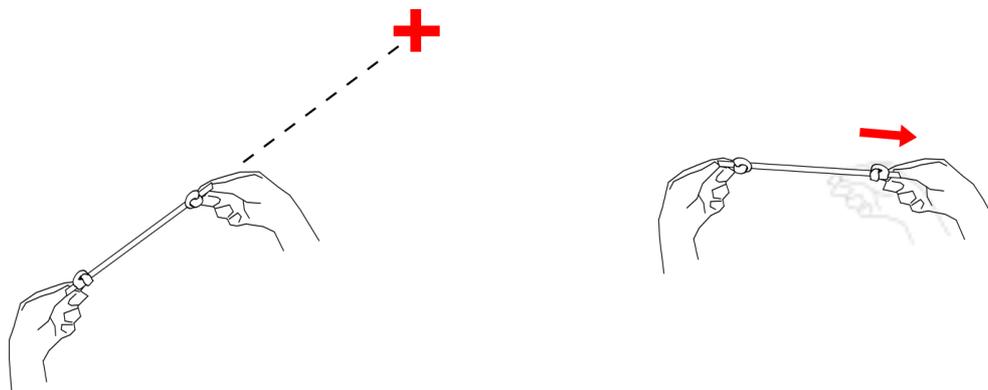


図 3.1: 指すアフォーダンス (左) と伸ばすアフォーダンス (右)

指すアフォーダンス

1.2 節で述べたように伸縮性を持つ紐は指向性を持つ。そのため、人間は伸縮性を持つ紐を用いて物を指す場合、図3.1左のように、弛みのないように張った伸縮性を持つ紐の両端を通る直線を用いて物を指すと考えられる。これは伸縮性を持つ紐が指すことをアフォードするからだと考えることが可能である。

ElasticCord においては、弛みのないように張った伸縮性を持つ紐の両端を通る直線を用いて画面を指すことによってポインティングを行う。このポインティング方法により、

ユーザは画面を直接ポインティングすることが可能である。

伸ばすアフォーダンス

伸縮性を持つ紐から人間は、図 3.1 右のように、伸縮性を持つ紐のそれぞれの端を手を持ち、伸縮性を持つ紐を伸ばすことをアフォードされる。

ElasticCord においては、伸縮性を持つ紐を伸ばすことにより連続量の調整を行う。例えば、ポインティングした 3D オブジェクトをポインティングした状態において、伸縮性を持つ紐を手前に伸ばすことによってそのオブジェクトを手前に移動させることが考えられる。ユーザは力覚フィードバックによってその連続量を理解することができる。

3.2 ElasticCord における 2 つの姿勢

ユーザは指すアフォーダンスを活用して画面をポインティングする。ユーザがポインティングを行っているか、行っていないかによって姿勢を 2 つに分類する。以降、この 2 つの姿勢をポインティング姿勢と非ポインティング姿勢と呼ぶこととする。それぞれの姿勢の概念図を図 3.2 に示す。次に、それぞれの姿勢について述べる。

ポインティング姿勢

ElasticCord が弛まずに真っ直ぐになるよう伸ばし、その状態のまま ElasticCord の両端を通る直線が画面と交差する姿勢をとるものである。

非ポインティング姿勢

ポインティング姿勢を取っていない場合を非ポインティング姿勢とする。

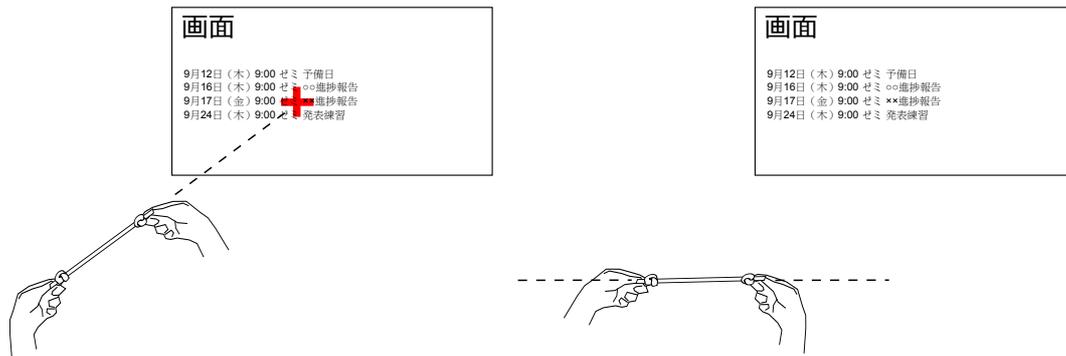


図 3.2: ポインティング姿勢 (左) と非ポインティング姿勢 (右)

3.3 姿勢によるインタラクション手法の変化

ElasticCord においては姿勢によってインタラクション手法を変化させる。これには2つの理由がある。

1つ目は、ある姿勢の場合には行いづらい操作が存在するからである。伸縮性を持つ紐を操作することのできる空間は、腕および手の可動域によって制限されるのである。例えば、ポインティング姿勢において、伸縮性を持つ紐を素早く画面方向に動かすなどの操作は難しい。

2つ目は、姿勢によってユーザのコンピュータ操作に対する意識が異なるからである。ユーザは、ポインティング姿勢をとっている際に、意識的に画面上のオブジェクトに対して選択やクリックなどの細かな操作を行う。一方、非ポインティング姿勢を取っている際は、そのようなコンピュータを能動的に操作する意識はほとんどない。だが、このような場合でも、ブラウザの戻る・進むや音楽プレーヤの曲送り・曲戻りと言ったわずかな操作を行いたいことがある。そのようなわずかなコンピュータ操作を行うためだけに、ポインティングを行い、カーソルを合わせ操作を行うことは煩わしい。非ポインティング姿勢において利用可能なインタラクション手法を提供することによって、そのような煩わしさを軽減することができる。

これらの理由から、図 3.3 に示すよう、姿勢によって行える操作を割り当てる。ポインティング姿勢においては、Stretch、Tap と言う2つの操作が、非ポインティング姿勢においては、Stretch、Tap、Push、Pull、Wind と言う5つの操作が使用可能になっている。次に、それぞれの操作について述べる。

3.4 ElasticCord における5つの操作

ElasticCord においては5つの操作が可能である。すなわち、Stretch 操作、Tap 操作、Push 操作、Pull 操作、Wind 操作である。

Stretch 操作、Tap 操作は伸縮性を持つ紐の伸縮性を活用した操作であり、その他の操作は、伸縮性を持つ紐の動きを用いた操作である。伸縮性を活用した操作を述べた後、伸縮性を持つ紐の動きを用いた操作について説明する。それぞれの操作の概念図を図 3.3 に示す。

3.4.1 伸縮性を活用した2つの操作

伸縮性を活用した操作は伸ばすアフォーダンスを活用した操作である。この操作はポインティング姿勢、非ポインティング姿勢どちらでも行うことができる。

Stretch 操作

ElasticCord がたるまない状態のまま、真っ直ぐ伸縮性を持つ紐を伸ばす操作である。

Tap 操作

伸縮性を持つ紐の片端を固定し、もう片端を素早く伸ばした後、元の位置に戻す操作である。また、Tap 操作の応用として 1.2 節で述べた指向性を活かした「指向性を持つ

Tap」がある。

指向性を持つ Tap

指向性を持つ Tap とは、伸縮性を持つ紐の一方の端を基準点として、もう一方の端を任意の方向に対して Tap 操作を行うことである。1.2 節において述べた指向性を持った操作の 1 つである。この操作によって、例えば、「左に伸ばす」操作によって、音楽プレーヤにおける「前の曲へ」操作を行うことや「右に伸ばす」操作によって、ブラウザにおける「次へ」操作を行うことができる。

3.4.2 伸縮性を持つ紐の動きを用いた 3 つの操作

伸縮性を持つ紐の動きをジェスチャのように用いた操作である。この操作は非ポインティング姿勢においてのみ行うことができる。

Push 操作

ElasticCord を画面に向かって押し出すように、素早く移動させる操作である。

Pull 操作

ElasticCord を画面から引き寄せるように、素早く移動させる操作である。

Wind 操作

ElasticCord のそれぞれの片端を回転させる操作である。図 3.3 の赤い矢印の方向に回転させる操作を奥 Wind と呼び、青い矢印の方向に回転させる操作を手前 Wind と呼ぶ。

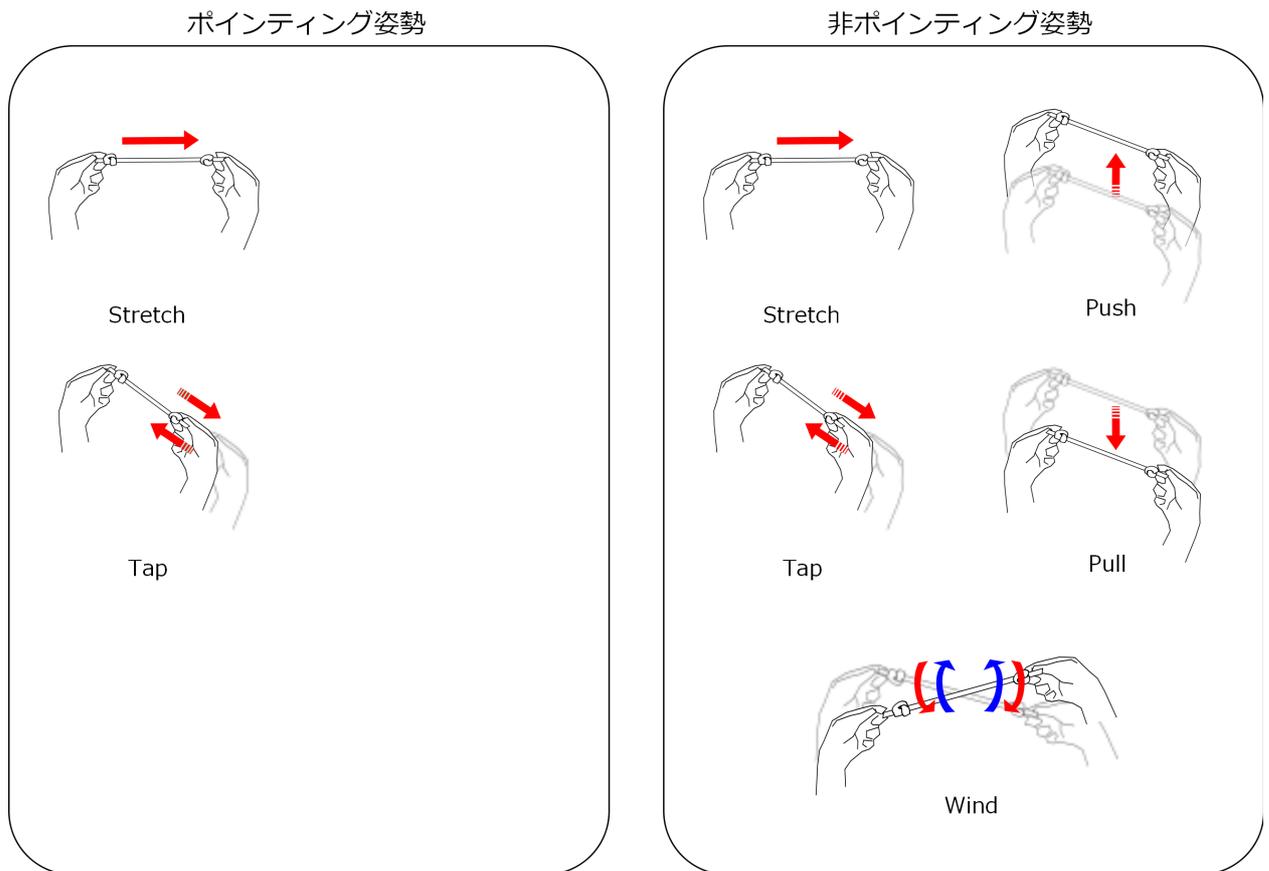


図 3.3: ElasticCord における姿勢と操作

3.5 ElasticCord の応用場面

伸縮性を持つ紐における「指す」アフォーダンスは大画面操作に対して相性が良いと考えている。それらの研究分野においては、レーザポインタや棒を大画面操作に用いる研究がなされてきた。これらの研究はレーザポインタの持つ指す機能や、棒の持つ指すアフォーダンスを活用した。伸縮性を持つ紐も同様に指すアフォーダンスを持つため、それらと同様にポインティングを行うことができる。さらに、伸縮性を活用した操作も行うことができる。このような操作を大画面操作に適応することによって、より効率的に大画面操作を行える可能性がある。

第4章 実装

本章では、ElasticCordの実装について述べる。今回実装したシステムでは、マーカを取り付けることによって伸縮性を持つ紐を認識する。システムは、認識部、状態推定部、応用部に分かれている。本章ではシステム構成について述べた後、ソフトウェア概要について述べる。次に、認識部、状態推定部について詳しく述べる。

4.1 システム構成

今回実装したシステムにおいて用いた開発環境、ハードウェア、ソフトウェアについて述べる。

開発環境

コンピュータとしてDELL社製XPS 630を用いた。OSとしてWindows Vistaを使用した。CPUはIntel Core2 Quad CPU Q9550、メモリは4GBである。

ハードウェア

伸縮性を持つ紐には図4.1に示す一般的に裁縫に用いられるゴム紐を使用した。伸縮性を持つ紐を認識するために、伸縮性を持つ紐にマーカを取り付けた。マーカが取り付けられた伸縮性を持つ紐を図4.2に示す。マーカにはオレンジマーカとグリーンマーカの2色のマーカを用いた。これらはポリエチレン製の球を蛍光塗料で塗装したものである。カメラには、Sony社製PLAYSTATIONREye(CEJH-15001)を2台用いた。画面に向かって左のカメラを左カメラ、向かって右のカメラを右カメラと呼ぶこととする。カメラはユーザをやや上方から撮影するために、約15度傾けて画面の上方奥に設置した。カメラを設置した様子を図4.3に示す。

ソフトウェア

画像処理ライブラリとしてOpenCV 2.1を用いた。



図 4.1: 伸縮性を持つ紐

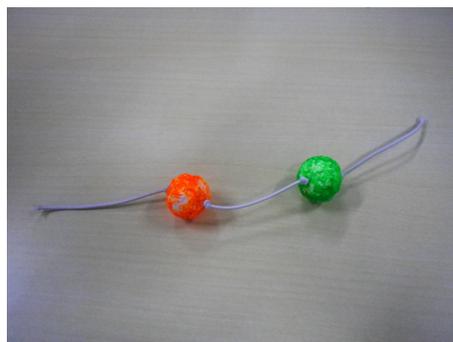


図 4.2: マーカを取り付けた紐

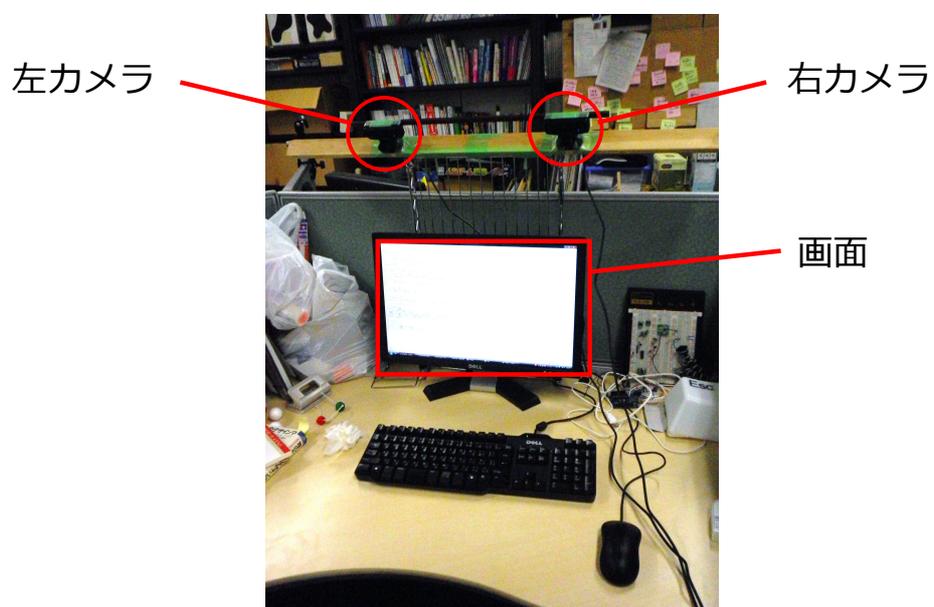


図 4.3: ハードウェアを設置した様子

4.2 ソフトウェア概要

ソフトウェア全体は次の3つに分割されている。

- 認識部
- 状態推定部
- 応用部

認識部は、三角測量を用いて伸縮性を持つ紐に取り付けられたマーカの座標を計測する部分である。また、状態推定部は認識部において得たマーカの座標から、伸縮性を持つ紐の状態を推定し、ElasticCord における姿勢と操作を認識する部分である。そして、応用部は状態推定部からポインティング位置や Tap 操作、Push 操作などの認識結果を受け取り、それらを基にコンピュータに対して命令を出す部分である。

次に、認識部、状態推定部について詳細に述べる。

4.3 認識部

認識部におけるおおまかな処理の流れを図 4.4 に示す。認識部では、伸縮性を持つ紐に取り付けられたマーカの座標を、2 台のカメラによる三角測量によって計測する。初めに、左右のカメラから画像を取得する。左カメラから取得した画像を図 4.5 に、右カメラから得られた画像を図 4.6 に示す。これらの画像から、マーカ領域を求めるため、HSV 表色系におけるマスキング、ノイズ除去を行った後、ラベリングを行う。ラベリングを行った領域のうち、面積が閾値以上の領域をマーカ領域と認識する。マーカ領域を認識した場合、それらの領域の重心を求める。そして、それらの重心に対して三角測量を行うことによってマーカの座標を得る。

次にそれぞれの処理について詳細に述べる。

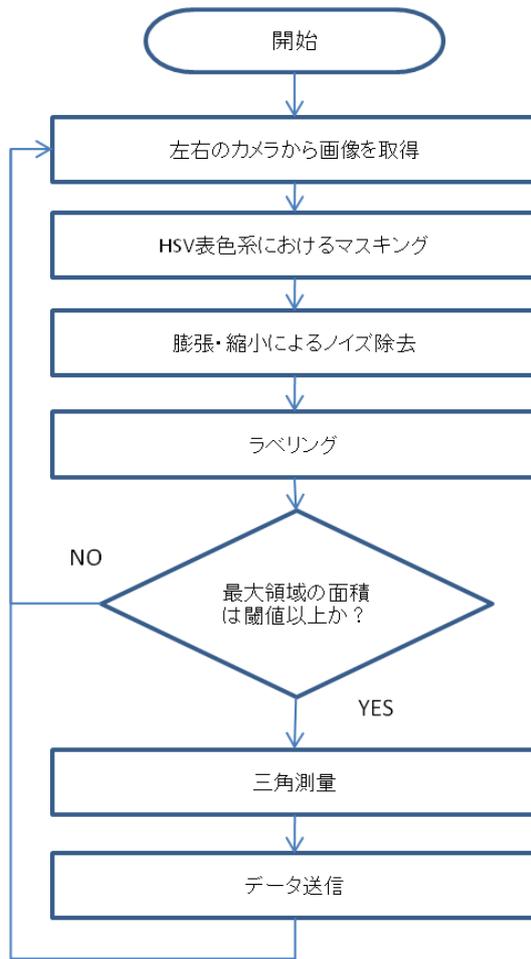


図 4.4: 認識部における処理の流れ



図 4.5: 左カメラから取得した画像

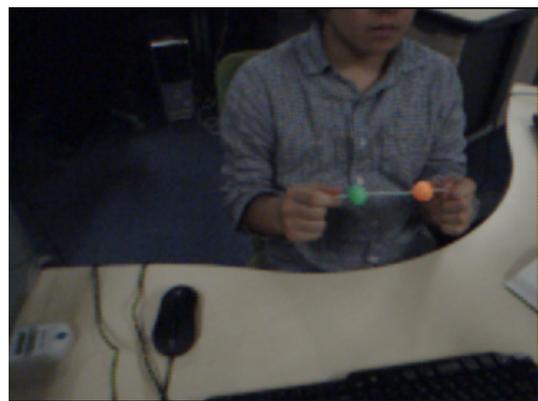


図 4.6: 右カメラから取得したカメラ画像

4.3.1 HSV 表色系におけるマスクング

カメラ画像から特定の色成分を持つ領域を抽出するためマスクング処理を行う。図 4.7、図 4.8 にマスクングした領域を表す画像を示す。まず、カメラ画像を RGB 表色系から HSV 表色系へと変換する。HSV 表色系においては、色相 (H) と明度 (V) が独立しているため、画像の明るさに対してロバストにマーカ領域の抽出を行うことが可能となる。RGB の三つの値の内、最大のものを MAX、最小のものを MIN とした時、RGB 表色系から HSV 表色系への変換は式 4.1 によって表現することができる。

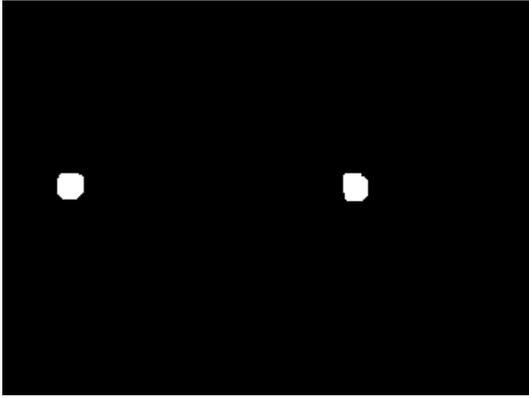


図 4.7: 左カメラによる画像から抽出されたマーカ領域

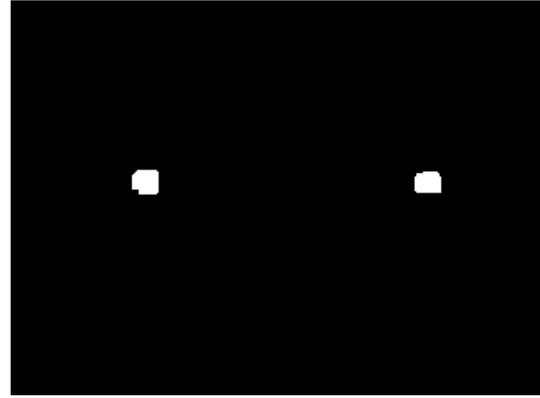


図 4.8: 右カメラによる画像から抽出されたマーカ領域

$$H = \begin{cases} 60 * \frac{G-B}{MAX-MIN} + 0, & if MAX = R \\ 60 * \frac{B-R}{MAX-MIN} + 120, & if MAX = G \\ 60 * \frac{R-G}{MAX-MIN} + 240, & if MAX = B \end{cases} \quad (4.1)$$

$$S = \frac{MAX-MIN}{MAX}$$

$$V = MAX$$

次に、式 4.1 によって HSV 表色系へ写されたカメラ画像から、式 4.2 によって特定の色成分を持つ領域を抽出する。ただし、式 4.2 において、色相の最小閾値を M_{min} 、最大閾値を M_{max} 、彩度の最小閾値を S_{min} 、最大閾値を S_{max} 、明度の最小閾値を V_{min} 、最大閾値を V_{max} 、マスクングされた画素 $P(H, S, V)$ とする。

$$P(H, S, V) = \begin{cases} 1 & if (M_{min} < H < M_{max} \wedge S_{min} < S < S_{max} \wedge V_{min} < V < V_{max}) \\ 0 & Otherwise \end{cases} \quad (4.2)$$

マーカにはオレンジマーカとグリーンマーカの2色のマーカを用いた。それぞれの領域を抽出するため、上述したHSV表色系によるマスクングを2度行う。

4.3.2 ノイズ除去、ラベリングおよび重心の決定

マスクングを行った画像に対し、膨張・縮小処理を行うことによってノイズ除去を行った後、ラベリングを行う。ラベリングにより、最大面積を持った連結成分だけを抽出することができる。もし、閾値以上の面積を持った連結成分がなければ、カメラにマーカが写っていないと認識し、これ以上認識処理を行わない。閾値以上の面積を持った連結成分があれば、その領域をマーカ領域として認識する。マーカ領域を認識した場合、その領域の重心を求める。マーカ領域から求めた重心を1枚の画像に描画した画像を図4.9に示す。それぞれの円の中心が各マーカの重心座標である。左カメラ画像から求められた重心を緑色の円で、右カメラ画像から求められた重心を青色の円で描画している。画像中において近接している2つの重心が、1つのマーカに対する重心である。

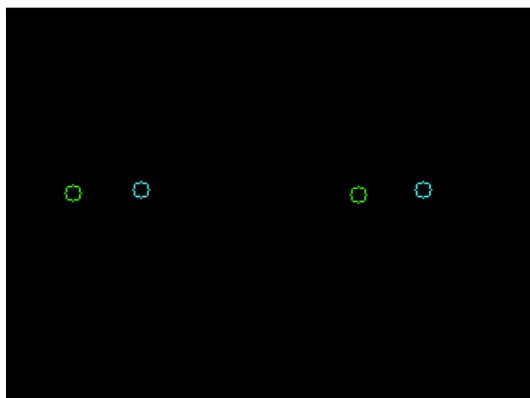


図 4.9: 抽出された重心

4.3.3 三角測量による座標計測

マーカ領域の重心に対して、三角測量を行いマーカの座標を得る。右カメラを原点とした座標系を用いているため、三角測量の際の座標系は図4.10に示す通りになる。マーカの座標は、カメラの焦点距離を f 、左カメラ、右カメラ間の距離を h とし、マーカ座標を $P(x, y, z)$ とし、左カメラにおけるマーカの座標を $M_L(x, y)$ 、右カメラにおけるマーカの座標を $M_R(x, y)$ とした時、式4.3により求めることができる。

$$\begin{cases} P_x = \frac{P_z \cdot M_{Lx}}{f} \\ P_y = \frac{P_z \cdot M_{Ly}}{f} \\ P_z = \frac{f \cdot h}{M_{Lx} - M_{Rx}} \end{cases} \quad (4.3)$$

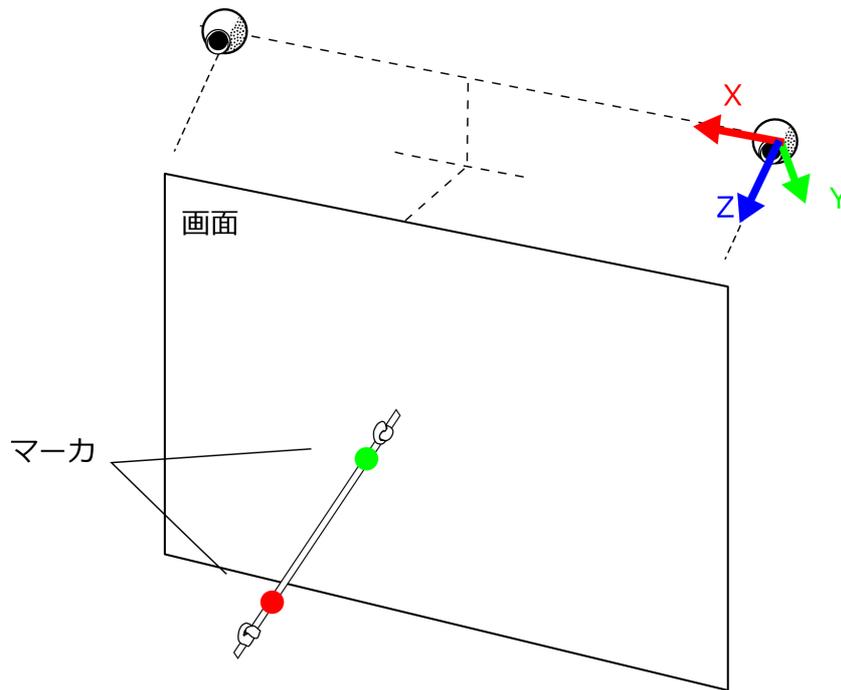


図 4.10: 三角測量の際の座標系

4.3.2 において求めた 4 つの重心に対し、同様の処理を行うことによってオレンジマーカとグリーンマーカの座標を求めることができる。

4.3.4 状態推定部へのデータ送信

状態推定部へは表 4.1 に示されるデータが送信される。通信プロトコルには Open Sound Controll(OSC)¹を用いた。

4.4 状態推定部

状態推定部では、まず座標変換によって認識部において用いた右カメラを原点とした座標系を、認識の際に便利なディスプレイの右上を原点とした座標系へと変換する。次に、変換した座標を用いて、伸縮性を持つ紐の座標、状態を推定する。また、ポインティング姿勢、Tap 操作や Push 操作と言った姿勢や操作の認識も行う。

¹<http://opensoundcontrol.org/>

表 4.1: 認識部分から状態推定部へと送信されるデータ。ただし、オレンジマーカ、グリーンマーカのうち、x 座標が大きい方をマーカ 1 とし、x 座標が小さい方をマーカ 2 とする。

型	内容
Float	マーカ 1 の x 座標
Float	マーカ 1 の y 座標
Float	マーカ 1 の z 座標
Float	マーカ 2 の x 座標
Float	マーカ 2 の y 座標
Float	マーカ 2 の z 座標

4.4.1 座標変換

変換前の座標系と変換後の座標系を図 4.11 に示す。矢印の元に描かれた座標系が変換前の座標系、矢印の先に描かれた座標系が変換後の座標系である。この変換によって、座標系と画面の右上を一致させることができる。

次に、座標変換方法について述べる。図 4.11 において、変換前座標におけるマーカの座標を点 P とする。また、Y 軸方向の座標系移動距離を dy 、Z 軸方向の座標系移動距離を dz とする。点 P を変換後座標へと座標変換した点を点 P' と置くと、点 P から点 P' への座標変換は

$$P' = CP = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -dy \\ 0 & 0 & 1 & -dz \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos & -\sin & 0 \\ 0 & \sin & \cos & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} Px & Px \\ Py & Py \\ Pz & Pz \\ 1 & 1 \end{bmatrix} \quad (4.4)$$

によって表すことができる。

この変換をそれぞれのマーカの座標に対して行う。

4.4.2 ポインティング位置の決定

姿勢、操作認識における概念図を図 4.12 に示す。色マーカを用いることによって、伸縮性を持つ紐の両端の座標を得ることができた。それぞれは図 4.12 中の赤い点 P 、緑の点 Q にあたる。この 2 点間のベクトルが紐ベクトルである。また、伸縮していない状態の紐ベクトルを基準ベクトルと呼ぶことにする。図 4.12 中の交点 c はポインティング位置を表す。また、方位 θ とは、画面に近い伸縮性を持つ紐の端の座標を原点とした時、紐ベクトルの XY 平面への射影と X 軸と成す角度である。傾斜 ϕ とは、紐ベクトルと Z 軸とが成す角度である。

画面に近い伸縮性を持つ紐の端の座標を \vec{p} とする。また、紐ベクトルを \vec{v} とする。ディスプレイ面は $Z=0$ の位置にあるため、点 \vec{p} を通るベクトル \vec{v} と $z=0$ 面との交点 c をポインティング位置とすればよい。この c の座標を求める式を式 4.5 に示す。

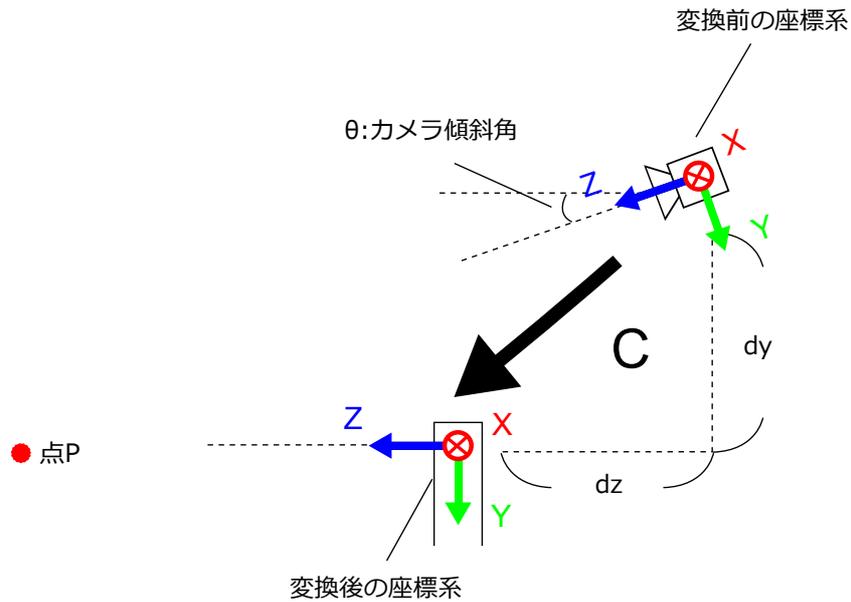


図 4.11: 座標系の変換

$$c(x, y, z) = (p_x - v_x * p_z / v_z, p_y - v_y * p_z / v_z, 0) \quad (4.5)$$

4.4.3 伸縮性を持つ紐の方位と傾斜

伸縮性を持つ紐の方位 は、伸縮性を持つ紐の XY 平面への射影の角度を求めることによって得ることができる。すなわち式 4.6 によって求めることができる。

$$= \tan^{-1}(v_y / v_x) \quad (4.6)$$

また、伸縮性を持つ紐の傾斜 は、Z 軸と伸縮性を持つ紐との角度を求めることによって得ることができる。すなわち式 4.7 によって求めることができる。

$$= \cos^{-1}(v_z / |\vec{v}|) \quad (4.7)$$

4.4.4 ElasticCord における 2 つの姿勢の認識

c を用いて 2 つの姿勢を認識する。 c がディスプレイ領域内に含まれている場合をポインティング姿勢、含まれていない場合を非ポインティング姿勢と認識する。

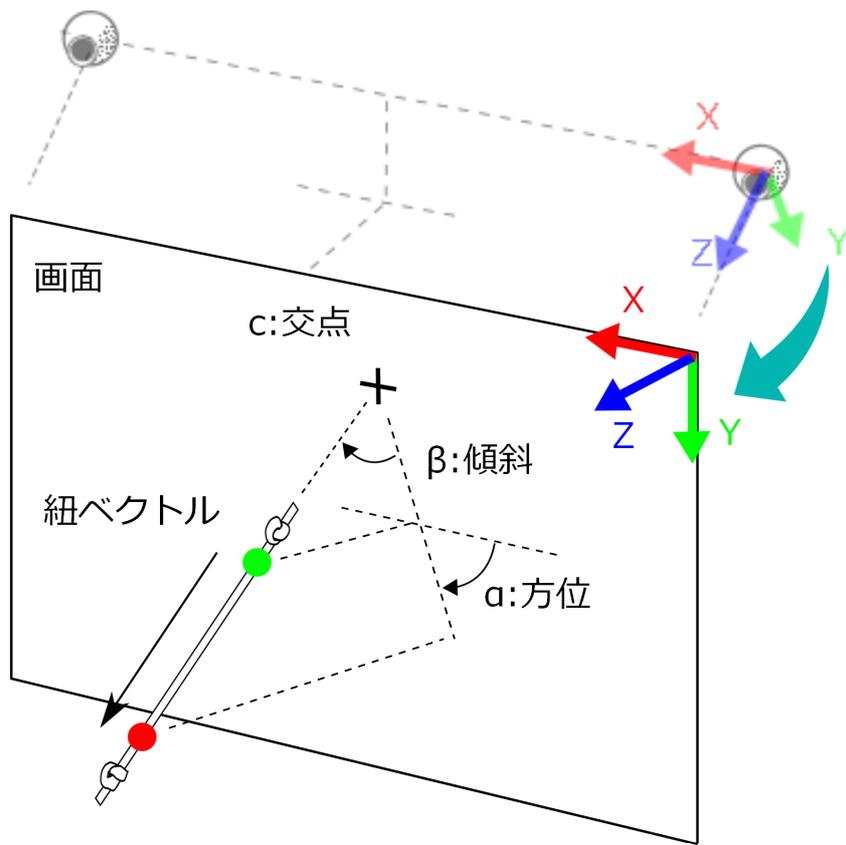


図 4.12: 操作認識における概念図

4.4.5 伸縮性を活用した2つの操作の認識

伸縮性を活用した操作である Stretch 操作、Tap 操作の認識手法について述べる。

伸縮性を持つ紐の位置、角度や方位によって、三角測量に起因する誤差が生じるため、基準ベクトルの大きさを固定にすることは難しい。そこで、紐ベクトルをトラッキングし、動的に基準ベクトルを更新し誤差を吸収する。紐ベクトルの大きさの変化が少ない場合は、基準ベクトルを紐ベクトルによって更新する。一方、紐ベクトルの大きさの変化が大きい場合は、伸びている状態と認識し、基準ベクトルを固定する。基準ベクトルの初期値にはプログラムが開始時してから最初に認識された紐ベクトルを設定する。そして、式 4.8 により伸縮率を求める。

Stretch 操作

伸縮性を持つ紐の伸縮率を求めるには式 4.8 を用いる。

$$\text{伸縮率} = \text{紐ベクトルの大きさ} / \text{基準ベクトルの大きさ} \quad (4.8)$$

伸縮率が閾値より大きい場合 Stretch 操作と認識する。

Tap 操作

Tap 操作を行った際、伸縮率は数フレームのうちに急激に上昇した後すぐに減少する。今回の実装においては、前7フレームにおいて伸縮率が閾値を交差した回数を数える。交差した回数が2回かつ、基準ベクトルと紐ベクトルとの角度差が $0.1rad$ 以下の場合 Tap 操作と認識する。また、指向性を持つ Tap は後述する重心移動量差分を、Tap 操作の付加情報として用いることで可能になる。

4.4.6 伸縮性を持つ紐の動きを用いた3つの操作の認識

伸縮性を持つ紐の動きを用いた操作である Push 操作、Pull 操作、Wind 操作の認識手法について述べる。これらの認識手法は、マーカの座標の動きを観察することによって発見的に設計した。

Push 操作

Push 操作を行った際、各マーカの Z 軸方向の移動量が数フレームのうちに急激に増加する。そのため、各マーカの移動量の総和を用いて Push 操作を認識する。今回の実装においては、まずそれぞれのマーカに対して、5 フレーム分の速度ベクトルを求める。それらの速度ベクトルを足し合わせたものの大きさが $80cm$ 以上、 $300cm$ 以下かつ、3 フレーム前と比較して Z 軸方向の移動量が $-3cm$ 以上ならば Push 操作と認識する。

Pull 操作

Pull 操作を行った際も、Push 操作と同様に各マーカの Z 軸方向の移動量が数フレームのうち

に急減に増加する。そのため、各マーカの移動量の総和を用いて Pull 操作を認識する。今回の実装においては、まずそれぞれのマーカに対して、5 フレーム分の速度ベクトルを求める。それらの速度ベクトルを足し合わせたものの大きさが $80cm$ 以上、 $300cm$ 以下かつ、3 フレーム前と比較して Z 軸方向の移動量が $3cm$ 以上ならば Pull 操作と認識する。

Wind 操作

ユーザが Wind 操作を行っている際、各マーカの速度ベクトルの大きさと回転角の分散は小さくなる。また、速度ベクトルの平均的な大きさは一定値以上になる。今回の実装においては、まずそれぞれのマーカに対して、前 10 フレームに対し速度ベクトルを求める。それらの速度ベクトルに対して、大きさの平均、分散、回転角の平均、分散を求める。大きさの平均が $1.8cm$ 以上 $20cm$ 以下、大きさの分散が $1cm$ 以下、かつ回転角の分散が $0.35rad$ 以下の場合 Wind 操作と認識する。

また、一方のマーカの 2 フレーム前の速度ベクトル、3 フレーム前の速度ベクトルとの外積を求め、その Z 軸方向の成分が $0cm$ 以下なら、手前から奥への Wind、 $0cm$ 以上なら奥から手前への Wind と認識する。

4.4.7 重心移動量差分の計算

方向性のある Tap を実現するため、伸縮性を持つ紐の重心移動量を用いる。伸縮性を持つ紐の重心は、次式により求めることができる。ただし重心を G 、マーカ 1 の座標を $M1$ 、マーカ 2 の座標を $M2$ する。

$$G = \frac{1}{2} \sum_{i=1}^2 M_i \quad (4.9)$$

現在の重心と 5 フレーム前の重心との差を重心移動量差分とする。例えば、Tap した際に、重心移動量差分の x 成分が $0cm$ よりも大きければ、左方向の方向性のある Tap であると認識できる。応用部分において方向性のある Tap を柔軟に扱えるように、状態推定部では重心移動量差分の値もそのまま応用部分に送信する。

4.4.8 応用部へのデータ送信

応用部へは表 4.2 に示されるデータが送信される。通信プロトコルには OSC を用いた。

表 4.2: 状態推定部から応用部へと送信されるデータ

型	内容
Int	ポインティング位置の x 座標
Int	ポインティング位置の y 座標
Float	重心移動量差分の x 成分
Float	重心移動量差分の y 成分
Float	伸縮率
Int	姿勢
Int	操作の種類
Int	Wind の種類

第5章 アプリケーション

本章では、今回実装した ElasticCord を用いたアプリケーションについて述べる。

ElasticCord の応用場면을 3.5 節において述べた。これらの応用場面に対して ElasticCord を適応する前に、デスクトップ環境において ElasticCord を用いたアプリケーションを実装し、伸縮性を持つ紐によるインタラクションにおける知見を得ることとした。コンピュータ操作を行うためのポインティングアプリケーションと音楽プレーヤを実装した。それぞれについて詳しく述べる。

5.1 ポインティングアプリケーション

ElasticCord を用いたポインティングアプリケーションを開発した。ユーザが ElasticCord によってポインティングアプリケーションを操作している様子を図 5.1 に示す。また、マウスエミュレータ使用中の画面を図 5.2 に示す。

ユーザはポインティング姿勢において、画面上のポインティングを行うことができる。ポインティングが行われている間は図 5.2 にも示されているように、ポインティング位置周辺に円が現れる。ユーザがポインティング姿勢において Stretch 操作を行った場合は、伸縮率に比例した拡大率によって、カーソル周辺の領域を拡大する。また、Tap 操作によって、円の中心座標においてクリックを行うことができる。ユーザが、非ポインティング姿勢を行った場合、ポインティングは行われない。しかし、奥 Wind を行った場合、下への Wind が、手前 Wind を行った場合、上への Wind が行われる。

ElasticCord における操作とコンピュータ操作との対応を表 5.1 に示す。

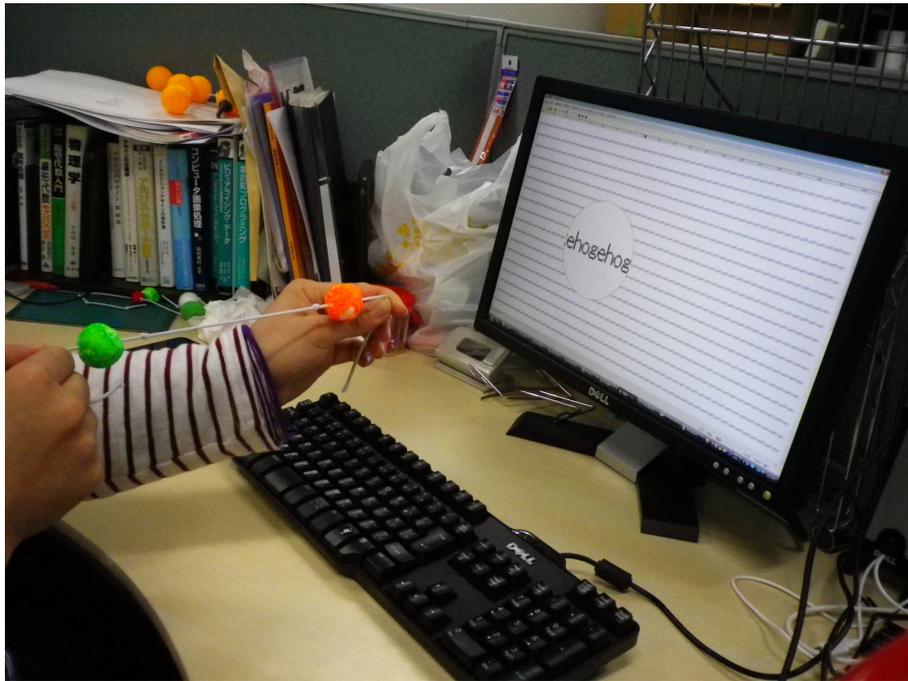


図 5.1: ポインティングアプリケーションを使用している様子

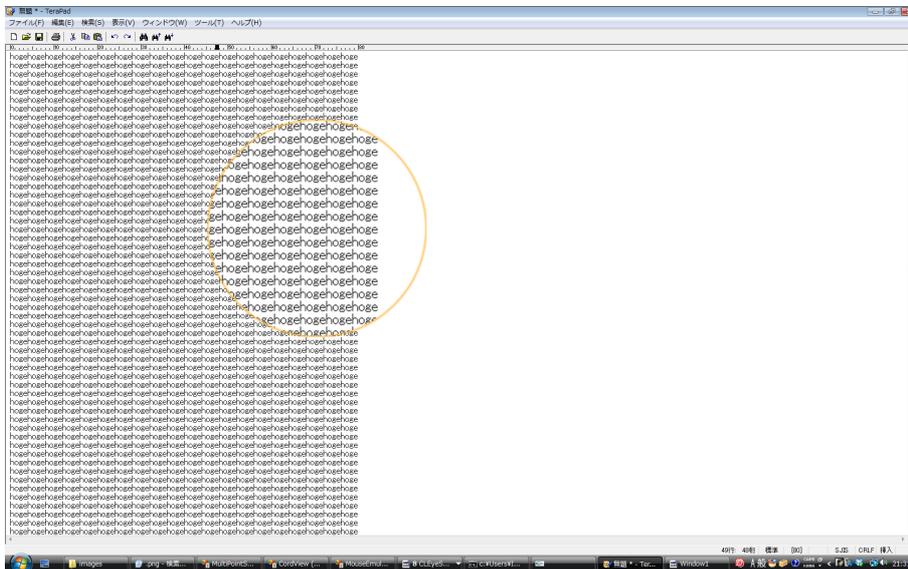


図 5.2: ポインティングアプリケーション使用中の画面

表 5.1: ElasticCord における操作とコンピュータ操作との対応図

ElasticCord における操作	コンピュータ操作
Stretch 操作	カーソル周辺の拡大
Tap 操作	クリック
Wind 操作	スクロール

5.2 音楽プレーヤ

ElasticCord を用いた音楽プレーヤを開発した。このアプリケーションでは、ユーザは音楽アルバムの「選択」、「再生」、「停止」、「曲送り」や「曲戻し」を行うことができる。ユーザが ElasticCord によって、音楽プレーヤを操作している様子を図 5.3 に示す。

ユーザはポインティング姿勢において、アルバムの「選択」を行う。このアルバム選択画面を図 5.4 に示す。画面上には、アルバムの画像が格子状に並んで表示されている。ポインティングが行われている場合は、ポインティング位置に赤い円が表示される。ユーザはアルバム画像をポインティングし、その状態のまま、Stretch 操作を行うことによってアルバムを選択できる。Stretch 操作を行った場合、図 5.5 のように選択しているアルバムの画像が拡大する。その時、伸縮率が 1.5 を超えた場合、そのアルバムを再生することができる。選択されたアルバムは、図 5.6 のように画像が赤い枠によって囲まれる。

また、非ポインティング姿勢においては、アルバムの「停止」、「曲送り」や「曲戻し」を行うことができる。非ポインティング姿勢における表示画面を図 5.7 に示す。ユーザは、Push 操作によってアルバムの停止を行う。また、右 Tap 操作によって曲送りを行い、左 Tap 操作によって曲戻しを行う。

ElasticCord における操作とコンピュータ操作との対応を表 5.2 に示す。

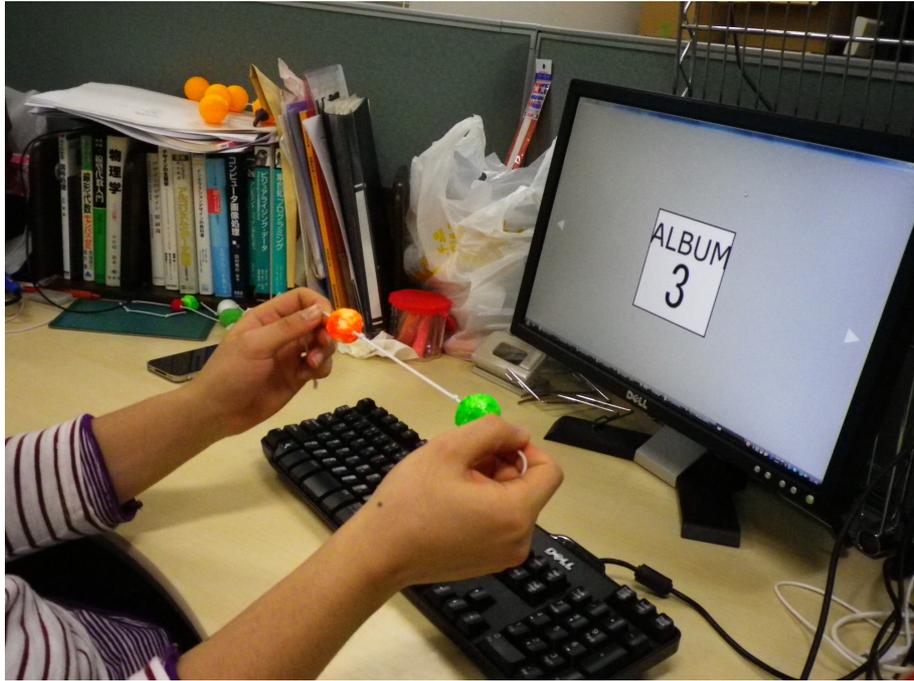


図 5.3: 音楽プレーヤを使用している様子

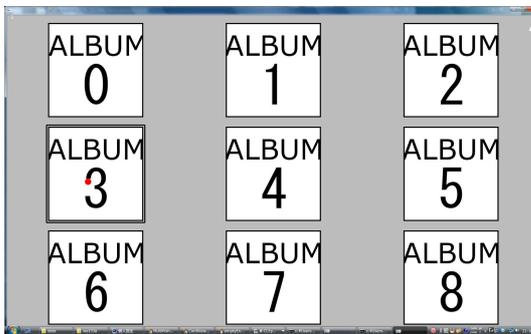


図 5.4: 音楽プレーヤ初期状態

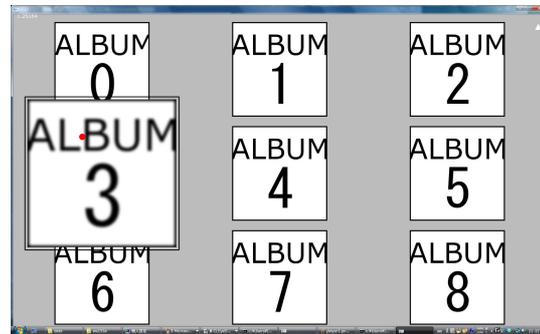


図 5.5: Stretch 操作によりアルバムを選択している状態

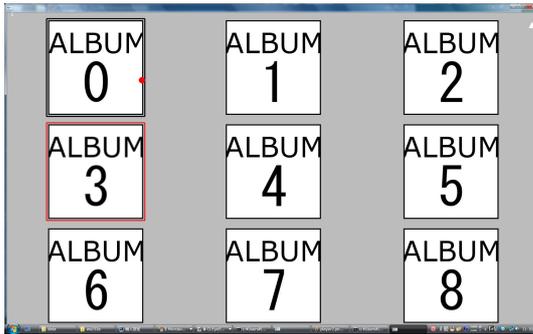


図 5.6: アルバムを選択した状態

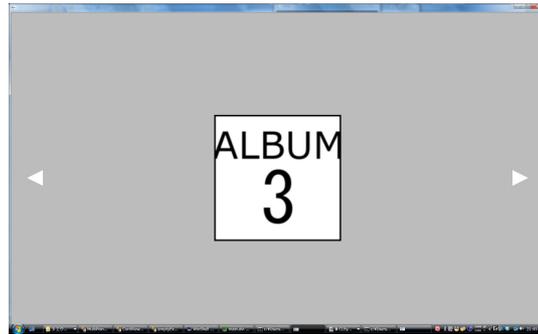


図 5.7: 非ポインティング姿勢における表示

表 5.2: ElasticCord における操作とコンピュータ操作との対応図

ElasticCord における操作	コンピュータ操作
Stretch 操作	アルバムの選択と再生
Tap 操作	次の曲へ、前の曲へ
Wind 操作	巻き戻し、早送り
Push 操作	再生中止

第6章 評価実験

伸縮性を活用した2つの操作である Stretch 操作、Tap 操作それぞれの認識率をする評価実験を行った。同時に、伸縮性を持つ紐を伸ばす際に手振れがどの程度生じるかを測定した。

手振れはインタラクション手法や GUI デザインに影響を与える。なぜなら、操作を行う距離が大きくなるほど手振れによる誤差の影響が大きくなるため、詳細な操作が困難になるからである。

6.1 実験目的

伸縮性を活用した2つの操作である Stretch 操作、Tap 操作それぞれの認識率を評価することを目的とする。また、伸縮性を持つ紐を伸ばす際に手振れがどの程度生じるかを評価することを目的とする。

6.2 実験環境

実験環境を表した図を図 6.1 に示す。画面の中心から、約 80cm の地点に椅子の座面の中心が来るように配置した。本実験は Stretch 操作、Tap 操作の認識率を確認するために行ったため、カメラがマーカを映しやすい位置に椅子を設置した。被験者はこの椅子に座り ElasticCord を用いた。

6.3 実験内容

被験者は著者 1 名であった。実験は以下の手順で行われた。

1. 画面のランダムな位置に半径 80px の円が表示される。(図 6.2)
2. 被験者は、カーソルがその円の中に入るように 3 秒間ポインティングを行う。
3. 被験者は 3 秒以内に Stretch 操作を 1 回行う。
4. システムに Stretch 操作が認識された場合、成功としてカウントする。操作が他の操作と認識された場合、もしくは 3 秒以内に何らかの操作が認識されなかった場合を失敗してカウントする。

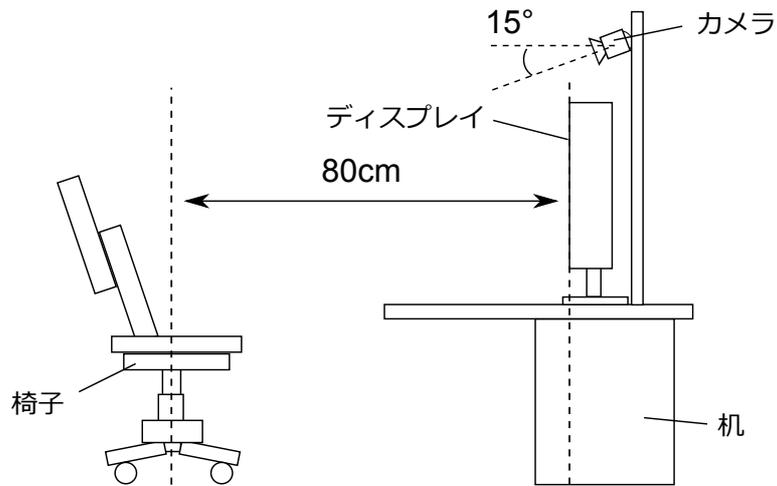


図 6.1: 実験環境

5. 1 へ戻る。1 から 4 を 20 回繰り返したら終了する。

Tap 操作についても同様に行った。

実験においては、エラー率および、タスクが成功した際の手振れ角度を測定した。手振れ角度とは、各操作が認識された際の、基準ベクトルと紐ベクトルとの角度差である。

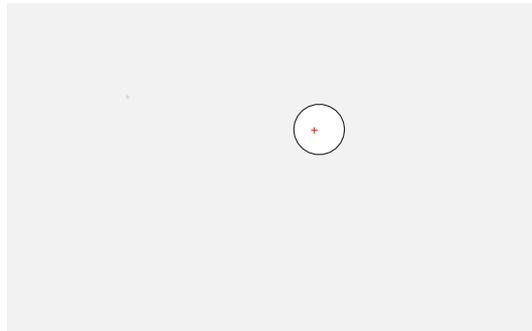


図 6.2: タスク実行画面

6.4 実験結果

実験結果を示した表を 6.1 に示す。

表 6.1: 実験結果と考察

操作の種類	成功回数	失敗回数	エラー率	手振れ角度平均 [度]	手振れ角度分散 [度]
Stretch 操作	17	3	0.15	2.52	2.80
Tap 操作	19	1	0.05	2.53	2.26

認識率について

Stretch 操作を 85% 認識することができた。また、Tap 操作の認識率は 95% とほぼ完璧に認識できた。Stretch 操作におけるすべての失敗は、Stretch 操作が Tap 操作と認識されるというものだった。これは、Stretch 操作を素早く行いすぎたため、Tap 操作と認識されてしまったことが原因だと考えられる。Stretch 操作と Tap 操作との切り分けをより明確にすることによって、認識精度を上げることができる。

手振れについて

どちらの操作においても、平均 2.5 度 程度の手振れが観測された。Stretch 操作において、手振れ角度分散は Tap 操作よりも大きい。これは、伸縮性を持つ紐を伸ばしている間に手振れすることが多いことを示している。

第7章 議論

ElasticCord について議論を行う。

7.1 操作感

筆者が試用した結果から得た ElasticCord の操作感について述べる。

当初はポインティング位置を定めるのが困難だったが、使用を通じて大まかにポインティング位置を定められるようになった。これは学習効果によって伸縮性を持つ紐を身体化することができたからだと考えられる。また、マーカ認識の精度向上によりポインティング位置の誤差が軽減すれば、より正確にポインティングが行えると考えられる。

Stretch 操作において、力覚フィードバックによって、おおまかに伸縮率を調整することが可能だった。また、伸縮率を保ったまま、ポインティング位置を変更することが可能であった。しかし、伸縮率を 1 程度に保ったまま伸縮性を持つ紐を把持することは難しかった。一方、伸縮率が高い場合のほうが、伸縮率に伴って大きくなる力覚フィードバックのために、安定して伸縮率を保つことができた。このことから、人間の伸縮率への分解能は、伸縮率 1 付近において低い一方、高い伸縮率においては高くなると考えられる。また、伸縮率を調整する際、一度伸縮率大きくした後、徐々に伸縮率を小さくしていくという行動が見られた。これは、伸縮性を持つ紐を引いて伸縮率を調整するよりも、伸縮性を持つ紐の伸縮を緩めて伸縮率を調整するほうが行いやすいからだと考えられる。

ポインティングアプリケーションの試用の際、Tap 操作において、クリックしたい位置とずれた位置がクリックされるケースが多かった。Tap 操作が認識された瞬間よりも前のポインティング位置をクリック位置とするなどの補正が必要ある。

方向性を持った Tap に関して、音楽プレーヤを用いた際に意図通りに曲送り・曲戻しを行うことができた。大まかに指向性を持った操作を実現することができた。

伸縮性を持つ紐の動きを用いた操作に関しては全般的に遅延が見られた。Push、Pull 操作については、手や腕の可動域の中で自然に使えることを確認した。しかし、伸縮性を持つ紐を把持したまま、腕を下した際などに誤認識が見られた。Wind 操作については、操作に慣れが必要だった。しかし、一度 Wind 操作が認識されると、連続して Wind 操作を行うことが可能だった。これらの問題は、操作の認識手法を変えることによって、改善できると考えている。

7.2 伸縮性を持つ紐を伸ばす際の手振れ

評価実験において、手振れが観測された。手振れの原因の1つに力覚フィードバックによる影響が考えられる。ユーザは、伸縮性を持つ紐を伸ばした際に受ける力によって、手を同じ場所の固定することが難しくなってしまう。そのため、手が振れてしまうと考えられる。

手振れによって生じるポインティング位置のずれは画面との距離に比例する。そのため、画面との距離が大きくなるほど、このような手振れによって生じるポインティング位置のずれが大きくなる。そして、ユーザは狙い通りの位置をポインティングすることが困難になってしまう。そのため、伸縮性を持つ紐の伸びをインタラクションに用いる際には、手振れ補正機能を実装すべきである。

また、力覚フィードバック以外にも手振れの原因が考えられる。著者がシステムを用いたところ、伸縮性を持つ紐が軽すぎたため、伸縮性を持つ紐を安定して把持することができなかった。粘土を用いて今回の実装で用いたマーカよりも重い球を作製し、伸縮性を持つ紐に取り付けたところ本実装で用いた伸縮性を持つ紐よりも安定して把持することができた。より重いマーカを用いることによって、手振れを軽減することが可能だと考えられる。

7.3 弛みの認識とそれを用いた操作の実現

現在は、伸縮性を持つ紐が「弛んでいる」状態を認識することができない。伸縮率が1よりも小さければ弛んでいる状態と認識することは可能に思える。しかし、三角測量における誤差を吸収するために伸縮率が減少する際にも基準ベクトルを更新している。それゆえ、弛みを認識することができない。

マーカを伸縮性を持つ紐の中央に取り付けることによって、伸縮性を持つ紐の「弛み」を認識できると考えている。こうすることによって、伸縮性を持つ紐に取り付けたマーカが直線状に並ばない時に、伸縮性を持つ紐が弛んでいると認識することが可能である。そのため、三角測量における誤差と関係なしに弛みを認識することが可能になる。

「弛み」を認識することによって、「弛み」姿勢をElasticCordにおける姿勢に追加することができる。また、「弛み」姿勢における操作を導入することによって、さらに豊かなインタラクションが実現できる。

7.4 離れた場所からのインタラクション

3.5節において述べたように、伸縮性を持つ紐の持つ「指す」アフォーダンスは10フィートUIや大画面操作などに対して相性が良いと考えている。そのためには、画面から離れた場所からのインタラクションを可能にする必要がある。これは、マーカ認識精度の向上とカメラの位置の工夫によって可能になると考えている。離れた場所からのインタラクションにおいて、伸縮性を持つ紐の伸縮性がインタラクションに与える影響を調査したい。離れた場所からのインタラクションを行う場合、手振れの影響を考慮すべきである。

7.5 複数の伸縮性を持つ紐の利用

現在の実装では、1つの伸縮性を持つ紐のみを検出可能である。複数の伸縮性を持つ紐を認識できるようにすることによって、複数ユーザによる協調作業が可能になる。例えば、ある人の伸縮性を持つ紐ともう一人の伸縮性を持つ紐をつなげることによって、可能なインタラクションを変化させることが考えられる。複数の伸縮性を持つ紐を用いる場合、近接するマーカをグルーピングし、トラッキングすることが必要になると考えられる。

7.6 マーカ認識置ける精度の向上

三角測量に起因する誤差によって、誤った伸縮率が計算される場合がある。また、一度誤差によって伸縮率が誤った値になってしまう場合、誤った伸縮率のまま認識が続けられてしまう。これらは、距離カメラを用いてマーカ座標の認識精度を向上させることによって回避できると考えている。また、伸縮性を持つ紐が弛んだ状態から張った状態へと遷移した場合に基準ベクトルを更新することによって軽減することができる。

7.7 伸縮性を持つ紐の動きを用いた操作の認識方法

今回実装したシステムでは、伸縮性を持つ紐の動きを観察することによって発見的に操作の認識手法を実装した。しかし、認識部のフレームレートに大きく依存する手法であった。そのため、ユーザの操作に対してロバストな認識が行えなかった。DP マッチングなどデータ系列の伸縮に対応した認識技術を用いることによって認識率を向上させることができると考えている。

7.8 ポータビリティ

今回の実装では、伸縮性を持つ紐の状態推定のために伸縮性を持つ紐に対して大きなマーカを取り付ける必要があった。なぜなら、色を用いてマーカ領域を抽出しているため、カメラにマーカがある程度大きく映る必要があった。その結果、伸縮性を持つ紐の性質であるポータビリティが損なわれてしまった。距離カメラなどを用いることにより、より小さなマーカを用いて状態推定を行えるようになり、ポータビリティを改善できると考えている。

第8章 結論と今後の課題

伸縮性を持つ紐を道具として用いた入力インタフェースである ElasticCord を開発した。ElasticCord は、伸縮性を持つ紐の持つ伸ばすアフォーダンスと指すアフォーダンスを活用する。その伸縮性を活用した Stretch 操作と Tap 操作が可能である。また、伸縮性を持つ紐の動きを用いた Push 操作、Pull 操作、Wind 操作も可能である。画面をポインティングしている場合、ポインティング姿勢、していない場合を非ポインティング姿勢と認識し、その姿勢によりインタラクション手法を変化させる。ElasticCord を用いるアプリケーションを開発した。また、Stretch 操作、Tap 操作の認識精度と、ElasticCord を用いる際の手振れについて評価を行った。伸縮性を持つ紐を用いるインタラクションにおいては、手振れを考慮しなければならないことが分かった。

今後の課題としては、認識精度の向上と弛みの認識があげられる。

謝辞

本論文の執筆にあたり、指導教員である田中二郎教授をはじめ、志築文太郎先生、三末和男先生、高橋伸先生には多くのご助言やご指導をいただきました。特に、志築文太郎先生には研究の方針から論文の執筆に至るまで、丁寧できめ細かいご指導をいただきました。心より感謝致します。

また、インタラクティブプログラミング研究室の皆様にも、大変お世話になりました。日頃から貴重な助言やご意見を頂きました。特に WAVE チームの皆様には、チームゼミでの意見のみならず研究生生活全体に渡っても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両親や、お世話になった多くの友人たちにも心から感謝します。

参考文献

- [BFKS99] Ravin Balakrishnan, George Fitzmaurice, Gordon Kurtenbach, and Karan Singh. Exploring interactive curve and surface manipulation using a bend and twist sensitive input strip. In *Proceedings of the 1999 symposium on Interactive 3D graphics*, I3D '99, pp. 111–118, New York, NY, USA, 1999. ACM.
- [BNF06] Gabor Blasko, Chandra Narayanaswami, and Steven Feiner. Prototyping retractable string-based interaction techniques for dual-display mobile devices. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pp. 369–372, New York, NY, USA, Apr 2006. ACM.
- [CB03] Xiang Cao and Ravin Balakrishnan. Visionwand: interaction techniques for large displays using a passive wand tracked in 3D. In *UIST '03: Proceedings of the 16th annual ACM symposium on User interface software and technology*, pp. 173–182, New York, NY, USA, 2003. ACM.
- [CGMT08] Vincenzo Caglioti, Alessandro Giusti, Lorenzo Mureddu, and Pierluigi Taddei. A manipulable vision-based 3d input device for space curves. In *Proceedings of the 5th international conference on Articulated Motion and Deformable Objects*, AMDO '08, pp. 309–318, Berlin, Heidelberg, 2008. Springer-Verlag.
- [GBS03] Tovi Grossman, Ravin Balakrishnan, and Karan Singh. An interface for creating and manipulating curves using a high degree-of-freedom curve input device. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 185–192, New York, NY, USA, Apr 2003. ACM.
- [HKB08] Gero Herkenrath, Thorsten Karrer, and Jan Borchers. Twend: twisting and bending as new interaction gesture in mobile devices. In *CHI '08 extended abstracts on Human factors in computing systems*, CHI '08, pp. 3819–3824, New York, NY, USA, 2008. ACM.
- [KJS⁺08] Yasuaki Kakehi, Kensei Jo, Katsunori Sato, Kouta Minamizawa, Hideaki Nii, Naoki Kawakami, Takeshi Naemura, and Susumu Tachi. Forcetile: tabletop tangible interface with vision-based force distribution sensing. In *ACM SIGGRAPH 2008 new tech demos*, SIGGRAPH '08, New York, NY, USA, 2008. ACM.

- [KW08] Erik Koch and Hendrik Witt. Prototyping a chest-worn string-based wearable input device. In *Proceedings of the 2008 International Symposium on a World of Wireless, Mobile and Multimedia Networks*, pp. 1–6, Washington, DC, USA, 2008. IEEE Computer Society.
- [LSRI00] Jay Lee, Victor Su, Sandia Ren, and Hiroshi Ishii. Handscape: a vectorizing tape measure for on-site measuring applications. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '00, pp. 137–144, New York, NY, USA, 2000. ACM.
- [MHON95] Tamotsu Murakami, Kazuhiko Hayashi, Kazuhiro Oikawa, and Naomasa Nakajima. Do-it: deformable objects as input tools. In *Conference companion on Human factors in computing systems*, CHI '95, pp. 87–88, New York, NY, USA, 1995. ACM.
- [MI04] Angelo Maravita and Atsushi Iriki. Tools for the body (schema). *Trends in Cognitive Sciences*, Vol. 8, No. 2, pp. 79 – 86, 2004.
- [She04] Brian Sheehan. The squiggle: a digital musical instrument. In *Proceedings of the 2004 conference on New interfaces for musical expression*, NIME '04, pp. 92–95, Singapore, Singapore, 2004. National University of Singapore.
- [SHHM10] Julia Schwarz, Chris Harrison, Scott Hudson, and Jennifer Mankoff. Cord input: an intuitive, high-accuracy, multi-degree-of-freedom input method for mobile devices. In *Proceedings of the 28th international conference on Human factors in computing systems*, CHI '10, pp. 1657–1660, New York, NY, USA, 2010. ACM.
- [SMKF09] Toshiki Sato, Haruko Mamiya, Hideki Koike, and Kentaro Fukuchi. Photoelastictouch: transparent rubbery tangible interface using an lcd and photoelasticity. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*, UIST '09, pp. 43–50, New York, NY, USA, 2009. ACM.
- [SPM04] Carsten Schwesig, Ivan Poupyrev, and Eijiro Mori. Gummi: a bendable computer. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '04, pp. 263–270, New York, NY, USA, 2004. ACM.
- [VMK⁺05] Kevin Vlack, Terukazu Mizota, Naoki Kawakami, Kazuto Kamiyama, Hiroyuki Kajimoto, and Susumu Tachi. Gelforce: a vision-based traction field computer interface. In *CHI '05 extended abstracts on Human factors in computing systems*, CHI '05, pp. 1154–1155, New York, NY, USA, 2005. ACM.
- [YK10] Zi Ye and Hammad Khalid. Cobra: flexible displays for mobilegaming scenarios. In *Proceedings of the 28th of the international conference extended abstracts on Human*

factors in computing systems, CHI EA '10, pp. 4363–4368, New York, NY, USA, 2010. ACM.

- [佐藤 10] 佐藤剛, 宮下芳明. Seek rope : 曲げて切って結べるシークバー. インタラクシオン 2010 論文集, pp. 197–200, 2010.
- [池田 02] 池田洋一, 木村朝子, 佐藤宏介. 道具の持つアフォーダンスを利用した触覚フィードバックデバイス. バーチャルリアリティ学会論文誌, Vol. 7, No. 3, pp. pp. 339 – 345, 2002.