

併行可能な動作に基づく  
インターラクション手法に関する研究

鈴木 優

システム情報工学研究科  
筑波大学

2011年3月

## 概要

併行可能な動作とは、既存の操作を構成する動作と同時に、かつ同一の肢体で遂行可能な動作のことを指す。本論文では、併行可能な動作に基づくインタラクション手法の構築に関して行った一連の研究について述べる。

本研究では、併行可能な動作を可併行性動作と呼ぶ。可併行性動作は、既存の操作を構成する動作との上位互換性や身体的連続性を持つという特徴を持つ。既存インターフェースに対して可併行性動作を適用することで、その操作を拡張し、新たなインタラクション手法を創出できる。また、創出したインタラクション手法は既存操作との互換性や連続性を保持できるため、既存インターフェースの操作性を維持できる。つまり、可併行性動作の適用により、既存インターフェースの操作を失うことなく、既存インターフェースに対して新たな入力チャネルを追加することが可能になる。

可併行性動作に基づくインタラクション手法の構想を具現化するために、タッチインターフェースに対して可併行性動作の適用を試みた。タッチインターフェースとはディスプレイに直接触れて入力を行うインターフェースであり、操作に利用する媒体で二つに分類することができる。本研究ではペンを用いて入力するものをペン入力インターフェース、指を用いて入力するものを指入力インターフェースと呼び、それぞれに対して可併行性動作を適用した。

ペン入力インターフェースでは、ペンを使用する際に人間が行っている動作の中から三つの動作に着目した。まず、可併行性動作として空中での手の動作に着目し、rolling, shaking, swingingという三つのインタラクション手法を開発した。また、ペンを握る指の動作に着目し、gripping, tapping, rubbingという三つのインタラクション手法を開発した。さらに、ペンを握る力を加減する動作に着目し、この動作を利用したインタラクション手法の利用方法を検討した。これらの動作を活用することにより、今まで無視されていた動作をペン入力インターフェースのインタラクションに利用可能にした。

指入力インターフェースでは、可併行性動作として指を使い分ける動作に着目した。従来の指入力インターフェースでは、タッチする指自体に意味はなく、タッチされた座標が入力情報として意味を持っていた。一方、タッチする指を識別する動作を指入力インターフェースに適用することで、タッチする指自体に意味を持たせることができる。たとえば、既存の指入力インターフェースでは人差し指でのタッチと中指でのタッチはどちらも同じ1点のタッチとして扱われるが、それらを別の入力として扱えるようになる。このようなインタラクションのコンセプトを Finger-Specific Interaction として提唱した。

本研究では、インタラクションに可併行性動作を導入することで、今までコンピュータとのインタラクションに活用されてこなかった人間の身体的能力を利用可能にした。その結果、人間がコンピュータに入力信号を送るチャネルが増加し、コンピュータに対して人間の操作意図をより伝えやすい環境を構築することが可能になった。

# 目 次

<b>第1章 序論</b>	<b>1</b>
1.1 本研究の意義	1
1.2 本研究の目的	2
1.3 本研究の課題	2
1.4 本研究の成果	2
1.5 論文の構成	3
<b>第2章 可併行性動作に基づくインタラクション手法の構想</b>	<b>4</b>
2.1 操作と動作	4
2.2 可併行性動作	4
2.3 可併行性動作に基づくインタラクション手法	6
2.4 可併行性動作のタッチインターフェースへの適用	7
2.4.1 対象とするタッチインターフェースとその種類	7
2.4.2 ペン入力インターフェースへの適用	7
2.4.3 指入力インターフェースへの適用	9
2.5 技術的課題	9
2.5.1 デバイスレベルの課題	9
2.5.2 動作検出用ソフトウェアレベルの課題	10
2.5.3 アプリケーションレベルの課題	10
<b>第3章 関連研究</b>	<b>12</b>
3.1 人間の能力の活用を目指すインタラクション手法	12
3.1.1 マルチモーダルインターフェース	12
3.1.2 Tangible Bits	13
3.2 ペン入力インターフェースに関する研究	14
3.2.1 入力チャネルの追加	14
3.2.2 GUI の改良	16
3.2.3 ストロークの活用	17
3.2.4 本研究と類似したアプローチ	18
3.3 指入力インターフェースに関する研究	18
3.3.1 ジェスチャの使用	19
3.3.2 タッチ情報の拡張	20

3.3.3 物理ウィジェットの導入 . . . . .	21
3.3.4 身体を識別するインタラクション . . . . .	22
<b>第4章 空中での手の動作を利用したインタラクション手法</b>	<b>24</b>
4.1 ペン入力インターフェース . . . . .	24
4.1.1 ペン入力インターフェースの利点 . . . . .	24
4.1.2 ペン入力インターフェースの課題 . . . . .	25
4.2 目指すペン入力インターフェース . . . . .	25
4.3 空中での手の動作 . . . . .	25
4.4 空中での手の動作を利用したインタラクション手法 . . . . .	26
4.4.1 開発したインタラクション手法 . . . . .	26
4.4.2 扱える入力値 . . . . .	27
4.4.3 ペンメタファの導入 . . . . .	27
4.4.4 既存操作との共存 . . . . .	28
4.5 空中での手の動作を利用したインタラクション手法の実現 . . . . .	28
4.5.1 Context Sensitive Stylus の開発 . . . . .	28
4.5.2 動作の検出 . . . . .	29
4.6 空中での手の動作を利用したインタラクション手法の応用 . . . . .	30
4.6.1 割り当て方法の検討 . . . . .	30
4.6.2 実装に用いた技術 . . . . .	30
4.6.3 ペイントツール . . . . .	31
4.6.4 スクロール支援ツール . . . . .	33
4.6.5 デジタルノートの拡張 . . . . .	35
4.7 評価実験 . . . . .	36
4.7.1 評価の目的 . . . . .	36
4.7.2 評価方法 . . . . .	36
4.7.3 結果と考察 . . . . .	38
<b>第5章 ペンを握る指の動作を利用したインタラクション手法</b>	<b>41</b>
5.1 ペン入力インターフェースで扱える入力情報 . . . . .	41
5.2 目指すペン入力インターフェース . . . . .	42
5.3 ペンを握る指の動作 . . . . .	42
5.4 ペンを握る指の動作を利用したインタラクション手法 . . . . .	43
5.5 Finger Action の実現 . . . . .	44
5.5.1 Pressure-Sensitive Stylus の開発 . . . . .	44
5.5.2 Finger Action の検出 . . . . .	45
5.6 Finger Action の操作性評価 . . . . .	46
5.6.1 評価の目的 . . . . .	46
5.6.2 評価方法 . . . . .	46

5.6.3 結果と考察	47
5.7 Finger Action の応用	49
5.7.1 ペンメタファの導入	49
5.7.2 開発したペイントツール	49
<b>第6章 ペンを握る動作を利用したインタラクション手法</b>	<b>52</b>
6.1 gripping の潜在能力	52
6.2 ペンを握る動作を利用したインタラクション手法	52
6.2.1 ペンを握る動作の特徴	53
6.2.2 gripping	53
6.2.3 離散値入力の利点と課題	54
6.2.4 連続値入力の利点と課題	54
6.2.5 類似するインタラクション手法との比較	54
6.3 gripping の実現	55
6.3.1 Pressure-Sensitive Stylus の改良	55
6.3.2 ワイヤレス Pressure-Sensitive Stylus	56
6.3.3 デバイスの設計	57
6.4 実験 1: 離散値入力に関する調査	58
6.4.1 目的	58
6.4.2 被験者と実験環境	58
6.4.3 タスク	60
6.4.4 パフォーマンスの測定	61
6.4.5 結果	61
6.4.6 考察	63
6.5 実験 2: ペンを握る力と筆圧の関係に関する調査	64
6.5.1 目的	64
6.5.2 被験者と実験環境	64
6.5.3 タスクと測定	65
6.5.4 結果	65
6.5.5 考察	66
6.6 実験 3: 適切な力空間に関する調査	66
6.6.1 目的	66
6.6.2 被験者と実験環境	66
6.6.3 タスク	67
6.6.4 パフォーマンスの測定	68
6.6.5 結果	68
6.6.6 考察	69
6.7 議論	70
6.8 gripping の応用	71

6.8.1	離散値入力の応用 . . . . .	71
6.8.2	連続値入力の応用 . . . . .	73
6.8.3	gripping と筆圧を組み合わせたインタラクション手法の応用 . . . . .	75
<b>第7章</b>	<b>タッチする指を識別するインタラクション手法</b>	<b>78</b>
7.1	指入力インターフェース . . . . .	78
7.1.1	指入力インターフェースの利点 . . . . .	78
7.1.2	指入力インターフェースの課題 . . . . .	79
7.2	目指す指入力インターフェース . . . . .	79
7.3	Finger-Specific Interaction の提唱 . . . . .	80
7.3.1	コンセプト . . . . .	80
7.3.2	有効性 . . . . .	80
7.4	Finger-Specific Interaction における検討課題 . . . . .	82
7.4.1	対応付けを記憶する困難さ . . . . .	82
7.4.2	負の運動特性 . . . . .	83
7.5	Finger-Specific Interaction の実現 . . . . .	83
7.5.1	プロトタイプシステム . . . . .	83
7.5.2	タッチを検出する技術 . . . . .	85
7.5.3	指を検出する技術 . . . . .	86
7.6	アプリケーション . . . . .	88
7.6.1	大画面向けペイントツール . . . . .	89
7.6.2	ソフトウェアキーボード . . . . .	94
7.6.3	モバイル向け音楽プレイヤ . . . . .	94
7.6.4	CSCW 向けフォトビューア . . . . .	95
7.6.5	ショルダーサーフィンを防止する認証 . . . . .	97
7.6.6	運動特性を活用した確認ダイアログ . . . . .	98
7.7	実装に関する議論 . . . . .	99
<b>第8章</b>	<b>結論</b>	<b>100</b>
8.1	研究成果の概要 . . . . .	100
8.2	貢献 . . . . .	101
8.2.1	ヒューマン・コンピュータ・インタラクション分野への貢献 . . . . .	101
8.2.2	ペン入力インターフェースへの貢献 . . . . .	102
8.2.3	指入力インターフェースへの貢献 . . . . .	102
8.3	今後の課題と展望 . . . . .	103
8.3.1	全体としての課題と展望 . . . . .	103
8.3.2	ペン入力インターフェースに対する課題と展望 . . . . .	104
8.3.3	指入力インターフェースに対する課題と展望 . . . . .	104

謝辞	106
参考文献	107
著者論文リスト	119

# 図 目 次

2.1 可併行性動作と基本動作、共通動作の関係	5
4.1 採用した三つの動作	27
4.2 開発したペン型デバイス Context Sensitive Stylus	29
4.3 複合センサモジュール Cookie	30
4.4 ペイントツールと FlowButton メニュー	32
4.5 提案するインタラクション手法を用いた操作体系	33
4.6 描画色を変更し、文字を描画した際の画面遷移	34
5.1 Finger Action: 提案する五つのインタラクション手法	44
5.2 Finger Action を検出するためのペン型デバイス Pressure-Sensitive Stylus	45
5.3 使用した感圧センサ	45
5.4 各操作にかかった平均時間とその標準偏差	47
5.5 平均入力ミス回数	47
5.6 アンケート結果	48
5.7 開発したペイントツール	50
6.1 ペンを握る指の様子	53
6.2 改良した Pressure-Sensitive Stylus	56
6.3 Pressure-Sensitive Stylus のグリップ部	56
6.4 ワイヤレス Pressure-Sensitive Stylus に利用したモジュール	57
6.5 ワイヤレス Pressure-Sensitive Stylus	57
6.6 比較する四つのトリガー操作の力変化のモデル	59
6.7 実験 1 で被験者に提示されるボックスとカーソル	60
6.8 エラー率のグラフ	62
6.9 クロス数のグラフ	62
6.10 選択時間のグラフ	63
6.11 計測したペンを握る力と筆圧をプロットした散布図	65
6.12 被験者に示された Pressure Map	67
6.13 ターゲット選択時間の結果	69
6.14 カーソル移動距離の結果	70
6.15 選択が困難な四つのターゲットを示した Pressure Map	71

6.16 ペイントツールに実装した gripping ランチャ	73
6.17 gripping ソフトウェアキーボード	74
6.18 gripping で明度を調節可能なカラーパレット	75
6.19 行動推定アプリケーション	76
6.20 gripping と筆圧の組み合わせ操作が可能なペイントツール	77
6.21 地図ビューアを操作する様子	77
7.1 3点タッチ操作のイメージ	81
7.2 開発したテーブルトップインタフェース	84
7.3 ハードウェア構成	85
7.4 コ型カブセを用いてアクリル板に固定した赤外線 LED	86
7.5 タッチを検出するための画像処理手順	87
7.6 テーブル上部のカメラで撮影した映像	88
7.7 指を検出するための画像処理手順	88
7.8 ペイントツールにおける指の組み合わせと機能との対応付け一覧	89
7.9 利き手のみでの操作する様子	90
7.10 非利き手の人差し指を用いて操作する様子	91
7.11 非利き手の中指をタッチしたときに利き手に割り当てられている色	92
7.12 非利き手の薬指を用いて操作する様子	92
7.13 非利き手の人差し指と薬指を用いて操作する様子	93
7.14 非利き手の中指と薬指をタッチしたときに利き手に割り当てられている色	93
7.15 指と機能との対応関係を提示するパネル	94
7.16 大画面向けソフトウェアキーボード	95
7.17 モバイルデバイス向けソフトウェアキーボード	95
7.18 モバイル向け音楽プレイヤ	96
7.19 CSCW 向けフォトビューア	96
7.20 FSI-PIN 認証を操作する様子	97
7.21 確認ダイアログ	98

# 表 目 次

2.1 提案するインタラクション手法と各動作	11
4.1 ペイントツールにおける操作割り当て	31
4.2 スクロール支援ツールにおける操作割り当て	33
4.3 デジタルノート拡張における操作割り当て	35
4.4 実験結果の平均値と標準偏差	40
5.1 ペイントツールにおける操作割り当て	50
6.1 gripping と筆圧, バレルボタンの比較	55
6.2 地図ビューアにおける操作割り当て	77

# 第1章 序論

本論文では、併行可能な動作に基づくインタラクション手法の構築に関して行った一連の研究について述べる。併行可能な動作とは、既存の操作を構成する動作と同時に、かつ同一の肢體で遂行可能な動作のことを指す。本研究ではこのような動作を可併行性動作と呼ぶ。本章では、まず、本研究の意義と目的について説明する。次に、本研究における課題を明確にし、それに対する成果について説明する。最後に、本論文の構成を示す。

## 1.1 本研究の意義

人間がコンピュータによる支援を得るために、コンピュータにさせたい作業を何らかの方法で伝達する必要がある。Norman は人間がコンピュータにさせたい作業を「人間の目標」、処理後のコンピュータの状態を「システムの物理状態」と呼び、さらにそれらの間に存在する隔たりを「淵」と呼んだ [56]。この淵ができる限り小さくし、人間の目標とシステムの物理状態を最小限にすること、つまり両者間の情報伝達を滑らかにすることが、ヒューマン・コンピュータ・インタラクション (HCI) 分野の研究における大きな目的の一つである。

グラフィカルユーザインターフェース (GUI) の発明により、コンピュータとのインタラクションはそれまで主流だった逐次対話インターフェースから直接操作インターフェースへと大きく転換した。これにより、人間の目標からシステムの物理状態に至る「実行の淵」が小さくなり、人間の操作意図を伝達しやすくなった。このときにマウスとキーボードを用いてディスプレイに表示された GUI を操作するというインタラクションスタイルが確立され、現在多くのシステムがこのスタイルを採用している。しかしながら、このインタラクションスタイルでは人間の持つ能力を十分に発揮できていない。人間はさまざまな知的能力や身体的能力を持つにも関わらず、コンピュータとのインタラクションにおいて使用されている身体動作はマウスやキーボードを叩く等、ごく一部のものに限られている。

本研究では、人間の知的能力や身体的能力を活用したインタラクションの実現を目指し、可併行性動作に基づくインタラクション手法を構築する。人間がコンピュータとのインタラクションに利用できる能力を増幅させることで、人間とコンピュータとの間の情報伝達チャネルが増加し、コンピュータに人間の操作意図を伝えやすくなる。その結果、人間の目標とシステムの物理状態の間に存在する「実行の淵」をさらに小さくすることが可能になる。可併行性動作に基づくインタラクション手法を構築する意義は、人間がコンピュータとのインタラクションに利用できる能力を増幅させることにある。

## 1.2 本研究の目的

人間は幼少の頃から多くの道具を使い、さまざまな活動を行っている。それらの活動の中で人間は多くの知的学習、および身体的学習を行い、経験的に知識やスキルを身に付けていく。これらの経験は非常に価値があり、日常生活におけるさまざまな場面において活用している一方で、それらはコンピュータとのインタラクションにおいては十分に活用されているとはいえない。

本研究の目的は、人間の目標とシステムの物理状態を隔てている淵を最小化するために、人間が経験的に体得してきた知的能力や身体的能力をコンピュータとのインタラクションに活用可能にすることである。これにより、人間とコンピュータとの間の情報伝達チャネルが増加し、人間の意図をより伝えやすいコンピュータを実現することが可能になる。

## 1.3 本研究の課題

人間が経験的に身に付けてきた知識やスキルの数は膨大であり、それら全てをコンピュータとのインタラクションに活用することは困難である。また、インタラクションに活用することが有効な知的能力や身体的能力についても未知な部分が多い。そこで、本研究は以下の2点を課題とする。

- 人間が経験的に体得してきた知的能力や身体的能力の中から、コンピュータとのインタラクションに活用できる可能性のある知的能力や身体的能力を取り出すこと
- 取り出した知的能力や身体的能力を活用したインタラクションの実現可能性を示すこと

## 1.4 本研究の成果

本研究では、人間の知的能力や身体的能力をコンピュータとのインタラクションに活用することで、新たなインタラクションの創出を目指す。筆者の主張する本研究における成果は以下のとおりである。

### 可併行性動作の利用の提唱

本研究では、コンピュータとのインタラクションに利用可能な身体的能力として、可併行性動作という身体動作を示した。可併行性動作とは、既存の操作を構成する動作と同時に、かつ同一の肢体で遂行可能な動作を指す。可併行性動作に基づくインタラクション手法を構築することで、人間とコンピュータとの間の情報伝達チャネルが増加し、人間の意図をより伝えやすいコンピュータを実現することが可能になる。

## 可併行性動作の利用を具現化した四つのインタラクション手法の創出

可併行性動作に基づくインタラクション手法の構築を具現化した例として、ペンで操作するタッチインターフェースに関する三つのインタラクション手法、および指で操作するタッチインターフェースに関する一つのインタラクション手法を創出した。これにより、可併行性動作に基づくインタラクション手法の実現可能性、具体的には、実際に可併行性動作がインタラクションに活用できること、およびその実装方法や具体的な応用方法、有効性を示した。

## 1.5 論文の構成

第2章では、可併行性動作に基づくインタラクション手法の具体的な構想とタッチインターフェースへの適用、およびそれを実現するまでの技術的な課題について述べる。

第3章では、コンピュータとのインタラクションに人間の能力を活用するという観点から関連する分野や研究を紹介し、本研究で提唱する可併行性動作との差異について述べる。また、可併行性動作の適用対象としたタッチインターフェースについて関連する研究を紹介し、本研究との差異について述べる。

第4章から第7章では、可併行性動作を実際にインタラクションに導入した具体例として、構築した四つのインタラクション手法について説明する。具体的には、第4章では空中での手の動作を利用したペン入力インターフェース向けインタラクション手法、第5章ではペンを握る指の動作を利用したペン入力インターフェース向けインタラクション手法、第6章ではペンを握る動作を利用したペン入力インターフェース向けインタラクション手法、第7章ではタッチする指を識別する指入力インターフェース向けインタラクション手法について述べる。

最後に第8章では、研究の結論として本研究の貢献を整理し、ヒューマン・コンピュータ・インタラクション分野、およびペン入力インターフェースと指入力インターフェースの発展に向けて、今後取り組むべき課題と展望について述べる。

## 第2章 可併行性動作に基づくインタラクション手法の構想

本章では、可併行性動作に基づくインタラクション手法の構想について説明する。まず、基本的な用語の説明、および可併行性動作の定義について述べる。次に、可併行性動作に基づくインタラクション手法と、その特徴について述べる。そして、併行可能動作に基づくインタラクション手法の実現可能性を示すために可併行性動作をタッチインターフェースに適用したインタラクション手法の概要を説明する。本章の最後に、可併行性動作に基づくインタラクション手法の実現にあたって解決すべき技術的課題について述べる。

### 2.1 操作と動作

基本的な用語である、“操作”と“動作”について説明する。操作とは、インターフェースを通して人間の意志や意図をコンピュータに伝達する行為のことを指す。人間の身体活動のことを動作と呼び、操作は一つ以上の動作で構成されている。ここで、操作を構成する動作を“基本動作”と呼ぶ。インターフェースとしてマウスを例に挙げると、クリックはマウスポインタで指定した1点の座標値を入力する操作である。クリック操作を行うために、人間はマウスを握る、人差し指を動かす等の身体活動を行う。これらの身体活動がクリック操作に対する基本動作である。

### 2.2 可併行性動作

併行性とは、ある物事と同時に別の物事が行われるという性質を表す。よって、可併行性は併行可能な性質、つまり、ある物事と同時に別の物事を行うことができるという性質を指す。

本研究では、このような可併行性を持つ身体動作を可併行性動作と呼び、人間がコンピュータとのインタラクションに利用できる能力を増幅させるアプローチとして、可併行性動作の利用を提唱する。以降、可併行性動作を併行動作と記す。

併行動作とは、具体的には以下の二つの条件を同時に満たす動作であると定義する。

- 既存の操作を構成する一部、もしくは全ての動作と同時に遂行可能
- それらの動作と同一の肢体で遂行可能

前者の条件は図 2.1 のように図示できる。たとえば、操作を構成する動作（基本動作）として動作 a-c がある。動作 a-c は一つの操作を構成する動作であるため、同時に遂行可能である。併行動作である動作 d は、基本動作群の一部、もしくは全て（ここでは動作 c のみ）と同時に遂行可能な動作である。また、基本動作群のうち、併行動作と同時に遂行される動作、図 2.1 では動作群 A, B に共通する動作 c のことを共通動作と呼ぶ。

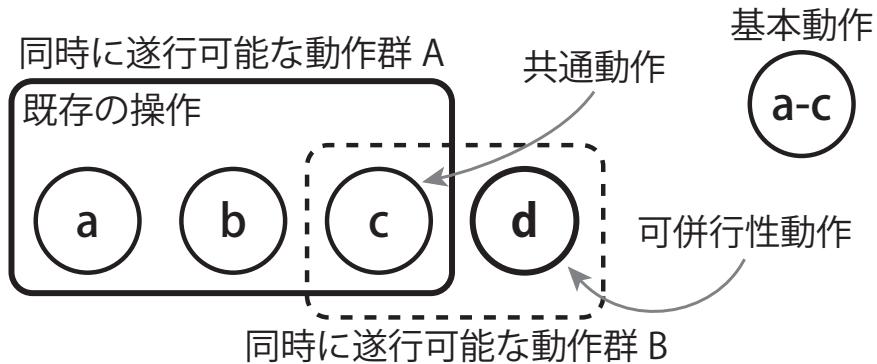


図 2.1: 可併行性動作と基本動作、共通動作の関係

併行動作は次の二つの特徴を持つ。第一に、併行動作と共通動作には互換性がある。併行動作は共通動作と同時に遂行できる動作であるが、併行動作の遂行には共通動作の遂行が必須であると捉えることもできる。よって、併行動作は共通動作に対して上位互換性を持つといえる。第二に、併行動作は共通動作と同時に遂行可能であることから、操作を構成する動作（基本動作）群と併行動作との間には身体的連続性が存在する。身体的連続性とは、複数の動作間に余計な動作が存在せず、二つの動作を滑らかに移行可能であるという性質である。基本動作群に含まれる各動作と併行動作は共通動作を介して繋がっているため、動作間を滑らかに移行することができる。

たとえば、既存の操作として、マウスを用いたポインティング操作を考える。一般的なマウスを操作する場合、人間はマウスを包み込むように握り、手首を動かしてポインティング位置を決定する。よって、マウスを用いたポインティング操作における基本動作とは、マウスを握る動作、および手首を動かす動作である。基本動作のうち、マウスを握る動作に着目すると、この動作と同時に遂行可能である同一肢體の動作として、マウスを持ち上げる動作、マウスを握る力を加減する動作等が考えられる。このような動作がマウスを用いたポインティング操作における併行動作である。これらの動作はマウスを握るという動作の基に成り立つており、併行動作が基本動作に対して上位互換性があることがわかる。また、基本動作群に含まれる各動作と併行動作は、共通動作としてマウスを握る動作を共有しており、それらの動作間には身体的連続性が存在する。つまり、基本動作群に含まれる各動作と併行動作を滑らかに移行することができる。

### 2.3 可併行性動作に基づくインタラクション手法

併行動作に基づくインタラクション手法とは、操作を行う動作に併行動作を適用したインタラクション手法である。コンピュータを操作する入力インターフェースとしてさまざまなデバイスが存在し、それぞれにおいて多くの操作があるが、既存の操作では限られた身体動作のみが利用されていることが多い。一方で、既存の多くの操作において遂行可能な身体的動作の自由度が残されている。併行動作に基づくインタラクション手法を構築することにより、既存の操作に内在する身体的動作の自由度をコンピュータとのインタラクションに利用可能にする。

図2.1では、動作群Bを用いたインタラクション手法がこれに該当する。2.2節のマウス操作の例では、マウスを持ち上げる動作や、マウスを握る力を加減する動作を利用した新しい操作を構築することが併行動作に基づくインタラクション手法の構築に相当する。

併行動作に基づくインタラクション手法を構築することはいくつかの利点があり、以下の3点に整理することができる。

#### 既存インターフェースの入力チャネルの増加

併行動作に基づくインタラクション手法を構築することにより、既存の入力インターフェースに新しい入力チャネルを追加することが可能になる。入力チャネルの増加により、そのインターフェースで行うことができる操作が増加するため、さまざまな情報を入力できるようになる。入力として扱われる情報は、離散的入力と連続的入力等、さまざまな観点で分類することができる。ほとんどの入力インターフェースは万能ではなく、扱いが得意な入力情報と不得意な入力情報を持っていることが多い。併行動作を用いて入力チャネルを追加するときに、そのインターフェースが不得意とする入力情報を利用可能にするようにインタラクション手法を設計することで、インターフェースの汎用性が高まり、さまざまなアプリケーションで活用可能になることが期待できる。

#### 既存操作の拡張

2.2節で述べたように、併行動作は共通動作に対して上位互換性を持つ動作である。よって、併行動作に基づくインタラクション手法は、既存のインタラクション手法の基本的な動作を保持しつつ、それを拡張させた手法であるといえる。つまり、併行動作に基づくインタラクション手法は既存のインタラクション手法の遂行可能なことを担保しつつ、さらにインタラクションを拡張できるといえる。

#### 既存の操作との連続性の保持

2.2節で述べたように、併行動作と操作を構成する動作（基本動作）群との間には身体的連続性が存在する。よって、併行動作に基づくインタラクション手法は、既存のインタラクション手法との連続性を保つことができる。連続性とは、併行動作に基づくインタラクション手法と既存のインタラクション手法が「身体動作として余計な動作を挟まず、両手法間を滑らかに移行できること」を指す。

## 2.4 可併行性動作のタッチインターフェースへの適用

コンピュータの利用方法が多様化する中で、マウスやキーボード等の汎用的な物理インターフェースや、物理的なデバイスを介さないジェスチャインターフェースや音声インターフェース等、さまざまな入力インターフェースが開発されている。併行動作はそれらのほとんどに適用可能である。

本研究では、併行動作に基づくインタラクション手法を具現化するために、タッチインターフェースへの適用を試みた。タッチインターフェースは専門的にも汎用的にも用いられているインターフェースである一方で、その操作性は十分ではなく、そのインターラクション設計において多くの余地が残されている。そこで、本研究では併行動作を適用する対象としてタッチインターフェースを採用した。

### 2.4.1 対象とするタッチインターフェースとその種類

タッチインターフェースとは、ディスプレイ面上の操作領域をペンや指で直接触れて操作するインターフェースである。タッチインターフェースは間接入力型と直接入力型の2種類に大きくわけることができる。間接入力型タッチインターフェースは、タッチパッドやペンタブレットのような、コンピュータの出力が操作領域とは異なる場所に表示されるデバイス、直接入力型タッチインターフェースは、タブレットPCやテーブルトップインターフェースのような、操作領域にコンピュータの出力が表示され、表示上で操作が可能なデバイスである。後者の直接入力型タッチインターフェースはタッチスクリーンと呼ばれており、本研究ではタッチスクリーン型のタッチインターフェースを対象とする。

インタラクションを行うための媒体という観点では、タッチインターフェースには二つの種類が存在する。一つは媒体としてペン型のデバイスを用いるタッチインターフェースである。これは液晶ペンタブレットやモバイルデバイス等に利用されている。もう一つは媒体として指を用いるタッチインターフェースである。これはテーブルトップインターフェースや公共の場所での端末やモバイルデバイスに採用されていることが多い。これら二つを区別するため、本論文では前者をペン入力インターフェース、後者を指入力インターフェースと呼ぶ。それぞれのタッチインターフェースについて併行動作の適用を試みた。

### 2.4.2 ペン入力インターフェースへの適用

併行動作のペン入力インターフェースへの適用として、以下の三つのインターラクション手法を創出した。三つ目のインターラクション手法は、二つ目の手法から派生し、発展したものである。これらについて表2.1にまとめた。

### 空中での手の動作を利用したインタラクション手法

人間がペンを使用する際の動作は、ペン先がディスプレイに接している状態と、接していない状態の二つに分類できる。前者は従来のペン入力インターフェースにおいて活用されている一方で、後者はほとんど利用されていない。ここでは、共通動作としてペンを握る動作に着目し、ペン先がディスプレイに接していない状態での動作、つまりペンを空中で動かす動作を併行動作として採用した。空中での手の動作を利用したインタラクション手法として、具体的に以下の三つの手法を開発した。

- **rolling** : ペンをペン軸周りに回す操作
- **shaking** : ペンをペン軸方向に振る操作
- **swinging** : ペンをペン軸と垂直方向に振る操作

### ペンを握る指の動作を利用したインタラクション手法

ペン先がディスプレイに接している状態と、接していない状態の両方において遂行可能なインタラクション手法の創出を目指した。

共通動作としてペンを握る動作に着目し、ペンを握る指の動作、つまりペングリップで指を動かす動作を併行動作として採用した。ペンを握る指の動作を利用したインタラクション手法として、具体的に以下の三つの手法を開発した。これら三つの手法には、操作する指の違いによる操作のバリエーションも存在する。

- **gripping** : ペングリップを握る操作
- **tapping** : ペングリップを軽く叩く操作
- **rubbing** : ペングリップを擦る操作

### ペンを握る動作を利用したインタラクション手法

これは、ペンを握る指の動作を利用したインタラクション手法から派生したインタラクション手法である。ペンを握る指の動作を利用したインタラクション手法を検討する中で、grippingがインタラクション手法として高い潜在能力を持つことがわかった。共通動作としてペンを握る動作に着目し、ペンを握る力を加減する動作を併行動作として採用した。本インタラクション手法に関しては、grippingについて詳細な検討を行った。具体的には、grippingによる離散値入力や連続値入力、grippingと筆圧を組み合わせたインタラクション手法について検討した。

### 2.4.3 指入力インターフェースへの適用

併行動作の指入力インターフェースへの適用として、以下のインタラクション手法を創出した。これについて表2.1にまとめた。

#### タッチする指を識別するインタラクション手法

指入力インターフェースでは他のインターフェースと比較して入力操作が少なく、操作として人が行える動作はディスプレイに指をタッチする動作以外ほとんど存在しない。一方で、指には文化的意味が含まれてあり、実生活ではそれらの意味を用いて各指を使い分ける場面も存在する。このことより、指を使い分けることも動作を見なし、そのような動作を指入力インターフェースの操作へと応用できると考えた。そこで、共通動作としてディスプレイに指をタッチする動作に着目し、指を使い分ける動作を併行動作として採用した。既存の指入力インターフェースの操作には指を用いるが、操作する指の区別はインタラクションの結果に寄与しない。タッチする指を識別するインタラクション手法では、タッチする指を区別し、その違いをインタラクションに利用可能にする。このようなインタラクションスタイルのコンセプトを Finger-Specific Interaction として提唱した。

## 2.5 技術的課題

併行動作に基づくインタラクション手法を実現するためにはいくつかの技術的課題が存在する。その技術的課題はデバイス、動作認識用ソフトウェア、アプリケーションの三つのレベルにわけることができる。

### 2.5.1 デバイスレベルの課題

併行動作をインタラクションに利用するためには、併行動作をコンピュータが検出する必要があるが、タッチインターフェースを構成する既存のデバイスでは併行動作の検出が行えない。よって、まず表2.1に示した併行動作を検出するためには必要な情報、およびその情報の取得に適当なセンサについて検討を行う必要がある。そして、それらのセンサを既存のタッチインターフェースに実装しなければならない。

また、併行動作に基づくインタラクション手法は、既存のインターフェースの操作との互換性が維持できるという特徴があるが、デバイスの設計次第ではその操作との互換性が保持できなくなる。よって、併行動作の検出を可能にするだけでなく、既存のタッチインターフェースで行える操作を妨げない設計を検討しなければならない。

### 2.5.2 動作検出用ソフトウェアレベルの課題

デバイスレベルの課題を解決し、既存のタッチインターフェースへのセンサ等の実装が完了した後の課題として、動作検出用ソフトウェアレベルでの課題がある。センサから検出できる値は数値データであるため、これらの生データを処理し、人間の動作を検出しなければならない。よって、表2.1に示したインタラクション手法の認識を行うためには、生のセンサ値から併行動作を検出する動作検出用ソフトウェアを開発する必要がある。快適なインターフェースにするためには、遅延がなく、リアルタイムな検出が行える動作検出用ソフトウェアにしなければならない。

### 2.5.3 アプリケーションレベルの課題

デバイスレベル、動作検出用ソフトウェアレベルでの課題を解決し、新しいインタラクション手法を認識可能にした後の課題として、アプリケーションレベルの課題がある。まず、併行動作により創出された新たなインタラクション手法をアプリケーションでどのように扱うかを考える必要がある。新たなインタラクション手法とアプリケーションの機能を適切に対応付けできるか否かがインタラクション手法の有用性を決定する重要な要因の一つであるため、アプリケーションのインタラクション設計ではこの点に注意しなければならない。また、新たなインタラクション手法を適用するアプリケーション自体にも注意を払う必要がある。併行動作の適用が適当なアプリケーションとそうでないアプリケーションを見極め、併行動作の適用がより有用なアプリケーションを見つけることも課題である。

表 2.1: 提案するインタラクション手法と各動作

対象インタフェース	提案するインタラクション手法	提案するインタラクション手法	共通動作	可併行性動作	創出したインタラクション手法
ペン入力インターフェース	空中での手の動作を利用したインタラクション手法	ペンを握る動作	ペンを空中で動かす動作	rolling, shaking, swinging	
ペン入力インターフェース	ペンを握る指の動作を利用したインタラクション手法	ペンを握る動作	ペンを握る動作	gripping, tapping, rubbing	
ペン入力インターフェース	ペンを握る動作を利用したインタラクション手法	ペンを握る動作	ペンを握る動作	gripping の拡張	
指入力インターフェース	タッチする指を識別するインターラクション手法	指でタッチする動作	指を使い分ける動作	Finger-Specific Interaction (指を識別する入力)	

## 第3章 関連研究

本研究の目的は、人間が経験的に体得してきた知的能力や身体的能力をコンピュータとのインタラクションに活用可能にすることである。そのアプローチとして、インタラクションに併行動作という概念を導入し、具体的な適用例としてペン入力インターフェースと指入力インターフェースへと適用し、それらの操作性向上を目指した。

本章では、まず、人間が経験的に体得してきた能力を用いたインタラクション手法という観点で関連する研究や分野について述べる。次に、ペン入力インターフェースと指入力インターフェースそれぞれについて、操作性向上を目指す研究について紹介する。これらの関連研究と本研究を比較することで、本研究の位置付けを明確化する。

### 3.1 人間の能力の活用を目指すインタラクション手法

人間は経験的にさまざまな知的能力や身体的能力を体得し、日常生活において意識的にも無意識的にもそれらを多く活用している。これらの能力を日常生活だけでなく、コンピュータとのインタラクションに積極的に活用することを目指した研究が行われている。ここでは、マルチモーダルインターフェース [109] と Tangible Bits [30] という二つの概念について紹介し、本研究で提唱する併行動作に基づくインタラクション手法との差異について述べる。

#### 3.1.1 マルチモーダルインターフェース

人間同士でコミュニケーションを行う場合、人間はさまざまなモダリティを利用する。モダリティとは、コミュニケーションを行うための経路のことを指す。

相手に物事を伝える（出力）ためのモダリティとしては、音声、表情、運動等が挙げられる。言葉によるコミュニケーションのことをバーバルコミュニケーション、言葉以外の情報伝達手段によるコミュニケーションのことをノンバーバルコミュニケーションと呼ぶ [126]。ノンバーバルな情報伝達手段として、身体動作や周辺言語、対人接触等があり [87, 115]、コミュニケーションの大半はノンバーバルな情報伝達手段によりとられている [109]。一方、相手から物事を受け入れる（入力）ためのモダリティとしては、視覚、聴覚、嗅覚、味覚、触覚の五感が挙げられる。視覚や聴覚はLCD やスピーカー等の従来のインターフェースで利用されている。触覚を利用したインターフェースとして、触覚フィードバックに関する研究 [9, 19, 64] が行われている。また、嗅覚や味覚を用いたインターフェースとして、たとえば広田ら [110] の

4種類の香料を用いた嗅覚ディスプレイや、鳴海ら [114] の視覚・嗅覚・味覚の間での感覚間相互作用を利用した味覚ディスプレイ等が開発されている。

人間はこれらのモダリティを複数同時に利用してコミュニケーションをとっている。たとえば、単純に向かい合って会話する場面でも、言葉だけでなく、相手の表情やしぐさ、声質等も利用している。つまり、人間は日常的に複数のモダリティを用いてコミュニケーションを行っている。

このように、複数のモダリティを用いてコミュニケーションを行うことをマルチモーダルコミュニケーションと呼び、これを人間とコンピュータとのインタラクションに応用したインターフェースを、マルチモーダルインターフェース [109] や、マルチメディアインターフェース [51] と呼ぶ。仮想現実 (Virtual Reality, VR) [127] や拡張現実 (Augmented Reality, AR) [95]、複合現実 (Mixed Reality, MR) 等の分野を中心に、マルチモーダルインターフェースに関する研究は数多く行われている。たとえば、Bolt の put-that-there [7] では、音声と身体動作を組み合わせたグラフィックスインターフェースを提案している。この研究では、大画面ディスプレイに図形描画を行うアプリケーションが示されている。このアプリケーションでは、ユーザは画面の前に座り、手の動きを用いて図形の描画位置を指定するためのカーソル位置を指示し、音声を用いて描画や配置等のコマンドや、図形の種類や描画色等のパラメータを指示する。また、伊賀ら [112] は頭の動きや呼気・吸気を用いて GUI を操作する環境を提案した。頭の動きを用いてマウスカーソルを動かし、呼気・吸気を用いてウィンドウの移動やアイコンのコピー・ペースト等のコマンドを実行することができる。

マルチモーダルインターフェースの研究では、全く異なる複数のモーダルを組み合わせていることが多い。たとえば、運動と音声等の組み合わせたインターフェースについて考える。音声を新たなモダリティとして追加した場合、運動だけでは行えなかったインタラクションが可能になるが、その一方で、音声は利用できる環境が限られる。たとえば、静かな場所では周囲に迷惑がかかり、騒がしい場所では使えない。異なるモーダルの利用にはこのような欠点も存在する。

併行動作に基づくインタラクション手法は、コミュニケーションを行うための経路を増やすという観点ではマルチモーダルインターフェースとも捉えることができる。本研究とその他のマルチモーダルインターフェースとの差異は使用するモーダルにある。多くのマルチモーダルインターフェースでは複数の異なるモーダルを使用する一方、本研究で提唱する併行動作は身体動作の拡張であり、単一のモーダルを使用して入力チャネルを増加させるというアプローチをとる。単一モーダル内での操作拡張であるため、既存のインターフェースが使用できる環境であれば、併行動作を用いたインタラクション手法は利用可能である。

### 3.1.2 Tangible Bits

Ishii らはコンピュータとのインタラクションに人間の知的能力や身体的能力を活用可能にすることを目指し、ヒューマン・コンピュータ・インタラクションの新しいパラダイムとして、Tangible Bits [30] を提唱した。Tangible Bits では、サイバースペース (コンピュータ内のデジタル情報) と物理世界の融合を試みている。また、キーコンセプトとして、Interactive surface

(デジタル情報とインタラクションできる物理世界の表面), Graspable objects (手につかみ, 操作できる物理オブジェクト), Ambient media (認知の周縁を活用したメディア) に焦点を当てている。これらのキーコンセプトを実現することで, 物に直接触れることで培ってきたスキルや周辺感覚によるアウェアネス等, 人間の持つ能力を活用可能にすることを目指している。

たとえば, metaDESK [86] では, アイコンやウィンドウ等のコンピュータ上の GUI 要素を実物体に付与し, コンピュータ内のオブジェクトを実体化した。つまり, 実体のない GUI を, コンピュータを操作するための物理オブジェクトに変換した。その結果として, “物を動かす”という人間が日常的に行っているインタラクションをコンピュータとのインタラクションに活用可能になった。inTouch [8] では, 遠隔地とのコミュニケーションにローラ型デバイスを用いた。遠隔地にいる人間同士がこのデバイスに手を置き, その動きを精密に伝え合うという, 触覚によるコミュニケーションを実現した。WaterLamp [14] では, ビットの流れを水滴として表現し, Pinwheels [14] では, ビットの流れを風として表現した。WaterLamp や Pinwheels はその認知に意識の集中を必要とせず, 人間は認知の周縁から情報を獲得できる。

Tangible Bits は人間が培ってきた触覚や周辺感覚等の活用を提唱する一方, 本研究で提唱する併行動作に基づくインタラクション手法は, 既存インターフェースや道具の使用において培ってきた感覚や運動能力を活用することを中心に, その他の知識等も活用することを目指し, その実現のために併行動作を活用する点で差異がある。

### 3.2 ペン入力インターフェースに関する研究

ペン入力インターフェースはマウスが発明される以前から利用されているインターフェースである一方で, その使い勝手や操作性はマウスやキーボードに劣っている部分も多い。ペン入力インターフェースは文字や絵を描く等の作業に向いており, 現在でもそのような用途では頻繁に利用されているが, 使い勝手や操作性には改善の余地がある。

ペン入力インターフェースの操作性向上のアプローチとしては, 入力チャネルの追加と GUI の改良, ストロークの活用の三つに分類できる。GUI の改良ではメニューインターフェースと文字入力インターフェースを取り上げ, それぞれのアプローチについて関連研究を紹介する。

また, 本研究のように空中での手の動作やペンを握る動作に着目したインタラクションもいくつか提案されている。これらの本研究と類似したアプローチをとる研究について紹介し, 本研究との差異を述べる。

#### 3.2.1 入力チャネルの追加

Miura らはペンをペンとしてではなく, 棒として利用するインタラクション手法, RodDirect [53] を提案した。RodDirect ではペンを手に握って操作するのではなく, ペンをその格納場所に挿入した状態で利用する。ペンを格納場所から出し入れするときの平行移動量と, 格納場所でペン型デバイスを回転させるときの回転量を新たな入力チャネルとして追加した。ペンを

巻き物の芯に見立てたメタファを採用し、スクロール操作等のインタラクションを利用する。RodDirectはペンの格納場所にイメージセンサを取り付けることで実現した。RodDirectと同様に、ペンでは通常行われない動作をインタラクションに利用にするデバイス、Tilt-Stick&Spin-Stick [111] があり、デバイスの倒立状態や正立状態等を用いたインタラクションが提案されている。

Siiolは文鎮メタファを利用したインタラクション手法 [73] を提案した。彼らは人間が紙に文字を書く際に紙を手のひらで押さえながら書くという動作に着目し、その動作をコンピュータとのインタラクションに応用した。具体的には、ユーザの手のひらがPDAやタブレットPCのディスプレイ下部に触れているか否かという状態を新しい入力チャネルとし、それをモード切り替え等の操作に応用した。手のひらの検出はタッチセンサを用いて実現した。

Biら [5] はペンを回転させる動作を利用した入力操作について詳細な検討を行った。この動作は本研究において rolling として提案したインタラクション手法 [80] であるが、彼らは回転速度や回転角度等、ペンの回転動作に関する属性について調査を行った。回転動作を検出するために Vicon motion tracking system を利用した。

Peephole Display [100] はペン入力インターフェースで操作する端末の位置を利用するインタラクション手法である。PDA等の3次元位置や移動量と、ペン入力操作を組み合わせた操作が可能である。たとえば、PDAの平面上の移動を利用した仮想大画面の操作や、PDAを持ち上げる操作によるズームアウト等が行える。ペン型デバイス側ではなく、ディスプレイ側に入力チャネルを増やした研究である。Rekimotoはペンを持ち上げて下ろすインタラクション手法、Pick-and-Drop [67] を提案した。これはペン先がディスプレイに接したままでないと操作できないというペン入力インターフェースの欠点を補う技術であり、複数のコンピュータ間でデータの送受信が簡単に行えるようになる。Subramanianら [76] はディスプレイ上部の空間に仮想的な複数のレイヤーを導入し、そのレイヤーにおいてディスプレイ面とは別のインタラクションを可能にした。たとえば、レイヤーでの操作を画像の移動等に応用した。これらの研究では、デバイス上部もしくは周辺の3次元位置を新しい入力チャネルとして追加したといえる。

ワコム社が開発しているタブレットでは、ペン先のXY座標だけでなく、筆圧やペンの傾きを検出することができる。その筆圧やペンの傾きを新しい入力チャネルとして応用した研究も存在する。Ramosら [65] は筆圧を用いて離散的なターゲット選択等への応用を試みた。彼らは筆圧を離散的入力として利用するために必要な設計空間について調査した。具体的には、人間が筆圧で使い分けることが可能な離散レベル、および離散値入力の決定に適したトリガー操作について調査するための実験を行った。その結果、最大6段階の離散値を利用でき、筆圧を素早く弱くする操作がトリガー操作として適していることがわかった。その他、筆圧はメニュー選択との組み合わせも検討されている。Mizobuchiら [54] も人間が加えることが可能な筆圧に関する調査を行っている。Adaptive Hybrid Cursor [70] というターゲット選択手法が提案されており、この手法では筆圧に応じてカーソルサイズを変更してターゲット選択を行う。Tianら [84, 85] はペンの傾きをメニュー選択に応用できるインターフェースを開発した。メニューには放射状に並んだメニュー [29] を採用し、メニュー項目の数や傾き方向が与

える影響等のパラメータについて調査を行った。

本研究では併行動作として、空中での手の動作、およびペンを握る指の動作に着目し、それらを用いたインタラクション手法を提案する。既存のペン操作との互換性や連続性の保持が担保されるため、従来のペン入力インターフェースの使い勝手を損なうことなく、新たな入力チャネルを追加できる。

また、上述した研究は一部を除いてペン型デバイスに入力チャネルを追加するものであつたが、ペン自体に出力チャネルを追加する研究も存在する。wUbi-Pen [43] はペンでの GUI 操作や描画操作に対してフォースフィードバックを返すデバイスである。また、Pen de Touch [33] はペンで仮想オブジェクトに触れた力を提示する触覚デバイスである。本研究では、ペン入力インターフェースの入力チャネルを増加させているが、これらは出力チャネルを増加させている点が大きな差異である。

### 3.2.2 GUI の改良

マウスでの操作を前提としたメニューインターフェースでは、メニューバーやポップアップメニューのようにメニュー項目は直線状に並んでいる。これに対して、ペン入力インターフェース向けのメニューインターフェースとして、放射状にメニュー項目が並んだ Pie Menu [29] がある。メニュー項目を放射状に並べることで、位置ではなく方向でメニュー項目を記憶することができる。従来の直線状メニューでは、ユーザはメニュー項目の位置を確認する必要があったが、Pie Menu では位置ではなく方向でメニュー項目を選択することができる。操作に慣れると、メニューを見ることなく項目を選択できるようになるという利点も持つ。Pie Menu と線形メニューの比較実験では、選択時間やエラー数に関して Pie Menu の方が有利であるという結果が示されている [10]。

放射状のメニューとして、Marking Menu [41, 42] が提案された。これは Pie Menu のメニュー表示がないもので、ペンの動きだけでメニュー項目を選択するメニューインターフェースである。階層化された Marking Menu を操作する場合、ジェスチャのような感覚でメニュー項目を選択できるという利点がある一方、Marking Menu ではジェスチャとメニュー項目とのマッピングを記憶する必要があるため、初心者にはほとんど使えないという欠点も存在する。FlowMenu [22] は Pie Menu と同様に放射状にメニュー項目を配置したメニューであるが、その選択方法が Pie Menu とは異なる。Pie Menu ではメニュー項目をタップすることでその項目を選択するが、FlowMenu ではメニュー項目を選択した後に開始した位置まで戻すようなストロークにより選択するという特徴を持つ。Control Menu [62] は FlowMenu と同様に、メニュー項目の選択にストロークを採用するが、Control Menu はメニュー項目の選択と、選択したメニューのパラメータ変更を 1 ストロークで行うことができる。パラメータはストロークを開始した点からの距離によって決定される。

これらのメニューインターフェースを応用した研究も行われている。Marking Menu を応用した研究として、Hinckley らは Pigtail を用いて Marking Menu を表示するシステム、Scriboli [26] を開発した。Pigtail とはペンストロークの最後に描くブタのしっぽのような形状のことを指し、これを描くことで何らかのトリガー操作とする手法である。Scriboli では操作対象を囲む

ストロークの後に Pigtail を描くことで Marking Menu を表示させ、実行するコマンドを選択する。

また、これらのメニューを文字入力インターフェースへ応用する研究もある。Quikwriting [60] は FlowMenu を用いた文字入力インターフェースであり、メニューの周囲に配置されたアルファベット等を FlowMenu と同じ操作方法で選択して文字を入力する。FlowMenu を用いた日本語入力インターフェース、Popie [113] では、メニューの周囲に配置された子音を FlowMenu と同じ操作方法で選択し、提示される言葉の候補の中から適当なものを選択することで文字入力をを行う。ペン入力インターフェースで文字入力をを行う場合にはソフトキーボードがよく用いられており、一文字ごとにペン先をディスプレイから離す必要があるが、これらの手法を用いるとペン先をディスプレイから離すことなく、1ストロークである程度の長さの文章を入力できるようになる。その他にも、T-Cube [88] や Cirrin [49] 等の文字入手法も提案されている。

これらの研究は GUI の改良によりペン入力インターフェースの操作性向上を目指している。一方で、本研究はペン入力インターフェースに新たな入力チャネルを追加するというアプローチをとる。しかしながら、筆者もメニューインターフェースの重要性は認識しており、本研究では新たな入力チャネルでの操作に適したメニューインターフェースの開発も行っている。

### 3.2.3 ストロークの活用

ペン入力インターフェースはマウス等と比較して、視差や手ぶれ等の影響により、ある1点を正確にタップする操作を苦手としている。一方で、ペン先をディスプレイに触れた状態でペンを動かすストロークを得意する。ペン先をディスプレイに押しつけることで動作が安定し、ストロークを行う過程で視差等に起因するズレも修正できる。

ストロークを活用したインタラクション手法として、Accot らの提唱した crossing [1] がある。crossing とはターゲットを横切るストローク操作である。crossing の有効性は彼らの実験により示されており、ペン入力インターフェース向けの有用なインタラクション手法として注目されている。crossing を応用した研究として、Apitz らはドローリングツール、CrossY [2] を開発した。CrossY では、あらゆる操作を crossing で行う。たとえば、ラジオボタンやスクロールバー等の GUI 部品を crossing によって操作する。

Smith らはスクロール操作を行うためのツール、the radial scroll tool [74] を提案した。このツールでは、ユーザがディスプレイをタップするとその位置にガイドが表示され、そのガイド上で円形のストロークを描くことでスクロールが行える。タップした位置がガイドの中心位置となり、そこから円を描くストロークを行う場所までの距離によってスクロール量が変化する。ガイドの中心付近で円を描くとスクロール量が多くなり、ガイドから離れた場所ではスクロール量が少なくなる。これをさらに改良したインタラクション手法として、Curve Dial [46] が提案された。Curve Dial は the radial scroll tool と同様に円形のストロークを描くことでスクロールを行うツールであるが、ジェスチャの描き方が異なる。the radial scroll tool では最初にタップした位置を常に中心にして円形のストロークを描く必要があったが、Curve Dial では円の中心を移動させながら円形のストロークを描くことができるため、ストローク操作に常に視線を向ける必要がないという利点がある。

その他にも、3.2.2節で紹介したMarking Menu [41, 42] やFlowMenu [22] 等のメニューインターフェースもストロークを活用するといえる。

これらの研究は、ペン入力インターフェースで従来から行うことができる操作であるストロークの使い方を拡張するというアプローチである。本研究も既存インターフェースの拡張ではあるが、使い方の拡張ではなく、入力チャネルを追加するという拡張を行っている点で異なる。

### 3.2.4 本研究と類似したアプローチ

本研究で提案する空中での手の動作利用したインタラクション手法で用いている動作と同じ動作を利用した研究として、Biら [5] はペンを回転させる動作を利用した入力操作について詳細な検討を行った。彼らはペン先をディスプレイに接した状態でのペンの回転動作を利用している一方で、本研究ではペン先がディスプレイに接していない状態での動作に着目している。

本研究のように、物体を握る力を利用したインタラクションの研究はいくつか存在する。Srinivasanら [75] は力の制御に関する人間の能力について調査した。ハンドヘルドPCを握るインターフェース手法 [24] や、マウスを握る力を利用するインターフェーション [13, 38] も提案されている。また、SqueezeOrb [61] というインターフェーションに握る力を利用するデバイスも存在する。

また、本研究と同様にペンの把持情報に着目した研究もいくつか行われている。ペンの把持情報とは、ペンを握る動作に関する特徴を表す情報で、たとえばペンを握る力や筆圧、把持位置等がある。山本ら [107] はペンの把持情報の中でもペンを握る位置に関して調査し、それをペンストロークのパラメータ調整へと応用した。本研究は、デバイスとしてペンに着目し、さらに把持情報の中でもペンを握る力を利用するという点でこれらの研究とは異なる。本研究と同様にペンを握る力に着目した研究も存在する。川端 [101] は各指にかかる無意識的な圧力の時間変化を調査し、個人認証への応用を目指した。一方、本研究はペンを握る力を用いてペン入力インターフェースの新しいインターフェーション手法を創出することを目的としており、研究の目的という観点において差異がある。

## 3.3 指入力インターフェースに関する研究

指入力インターフェースは、駅の券売機や銀行のATM等の公共の場所における端末から、スマートフォンやPDA等のモバイルデバイス、会議等に利用されるテーブルトップインターフェースに至るまで、さまざまな利用場所や利用形態で採用されているインターフェースである。特別なデバイスを用いることなく指だけで操作できることが指入力インターフェースの大きな利点であり、このように広く普及する要因の一つになっている。一方で、指でしか操作できないという入力チャネルの少なさが指入力インターフェースの使い勝手や操作性の制限てしまっている。

ここでは、指入力インターフェースの操作性を向上させることを目的とした研究について紹介する。操作性向上のためのアプローチは、ジェスチャの使用、タッチ情報の拡張、物理ウィジェットの導入の三つに分類できる。それぞれのアプローチについて関連研究を紹介する。

また、本研究のように身体を識別するインタラクションもいくつか提案されている。これらの研究について紹介し、本研究との差異を述べる。

### 3.3.1 ジェスチャの使用

指入力インターフェースでは、基本的に指をタッチする座標のみが入力情報として利用されている。ジェスチャはこの座標値の時間変化を利用して、さまざまな入力を可能にするというアプローチである。

Wu ら [99] は指と手全体を用いたジェスチャインタラクションを提案した。彼らは人間が実際のテーブル上で行っている動作を利用したジェスチャセットを作成し、1本指でのジェスチャ、2本指でのジェスチャ等いくつかのカテゴリに分類した。それらのジェスチャを家具配置アプリケーションに実装し、応用として示した。Malik ら [48] は大画面ディスプレイとのインタラクション手法として、両手の指によるジェスチャ操作を提案した。ジェスチャとして手を握った状態、指を1本伸ばした状態、2本伸ばした状態や、ピンチジェスチャ、手を握るジェスチャ等を認識し、これらを大画面ディスプレイ上のオブジェクトとのインタラクションやズーミング操作等に応用した。このシステムでは、Visual Touchpad [47] というデバイスを基にしてジェスチャの認識を行っている。Olwal ら [58] は指入力インターフェースにおける正確で高速な選択操作を実現するために、ディスプレイを擦るような rubbing ジェスチャとタップを利用した。Freeman らの ShadowGuides [18] も同様に指と手のジェスチャを用いたシステムである。さらに、SmartSkin [68] を用いた両腕による操作等 [108] も検討されている。また、物理世界におけるインタラクションを指入力インターフェースに導入することを目指す研究 [20, 40, 66, 98] でもジェスチャが導入されている。

ディスプレイ平面上だけでなく、ディスプレイ上の空間でのジェスチャも提案されている。Hilliges ら [25] はテーブルトップ上の空間を利用した、コンテンツとの3次元的なインタラクションの実現を目指した。たとえば、ディスプレイに表示された物体をつまみ、持ち上げるという操作を可能にした。Z-touch [104] はディスプレイ面近傍における物体の高さを検出できるシステムである。ディスプレイ面付近の指の3次元姿勢を検出し、それを用いた地図のズーミング等が行われている。その他にも BiDi Screen [28] や TouchLight [97] 等でも3次元ジェスチャが検討されている。

静電容量式のタッチパネルの普及に伴い、マルチタッチ操作が可能なデバイスも増加してきた。特に、モバイルデバイスは小型化のためにキーボード等の物理的なボタンを多く搭載することは難しく、入力として利用できる情報が少ない。マルチタッチはこれを解決し、かつ直感的な操作感を演出するための一つのアプローチとして広く利用されている。一方、マルチタッチとは異なるアプローチを試みる研究もある。MicroRolls [71] では親指でのジェスチャが提案された。MicroRolls は親指をディスプレイにタッチした状態で指を動かさず、指の接触面をすらすようなジェスチャである。具体的には、上下左右の4方向のジェスチャ、

および指の時計回りと反時計回りの2方向の回転ジェスチャからなる。これらのジェスチャはコピー、ペースト等の頻繁に利用する機能や、モード切り替え等に利用された。Slide Rule [35]では、視覚障害者のための指入力インターフェースについて検討された。具体的にはジェスチャ入力と音声フィードバックを利用し、ディスプレイを視認できなくても指入力インターフェースの操作を可能にすることを試みた。また、モバイルデバイスの表面と裏面を使ったジェスチャ [45]も提案されており、両面を掴むジェスチャや両面をドラッグするジェスチャ等が検討された。指入力インターフェースの両面を使ったインタラクションは多数提案されており、たとえば HybridTouch [77] や LucidTouch [96] 等がある。

ジェスチャは、指をタッチする座標の時間変化を用いているため、空中でのジェスチャ等を除いた多くのジェスチャはハードウェアに特別な仕組みを用意する必要がないという利点がある。一方で、ジェスチャのみによる入力情報の追加には限界がある。本研究ではタッチする指を識別することによる入力情報の追加を目指す。本アプローチはタッチする指を識別することであり、その時間変化を利用することは容易である。よって、ジェスチャと本研究でのアプローチは相性が良く、組み合わせて利用することでより多くの入力情報を創出できる可能性を持つ。

### 3.3.2 タッチ情報の拡張

上述したように、指入力インターフェースでは基本的に指をタッチする座標のみが入力情報として利用されている。しかしながら、単純に指をタッチする動作にもタッチする強さや指の向き等、さまざまな属性が隠されている。つまり、指の接触面情報にはインタラクションに利用されていないさまざまな情報が内在しているといえる。タッチ情報の拡張とは、これらの内在する情報を指入力インターフェースのインタラクションに活用するというアプローチである。

Benko らは SimPress [4] というクリック操作を提案した。SimPress とは、指の先端部分でのタッチと指の腹でのタッチの使い分けを可能にする手法である。指の先端部分でタッチした場合はホバー操作となり、指の腹でタッチした場合はドラッグ操作となる。これら二つの状態の識別には指の接触面情報を利用している。ShapeTouch [11] では、接触面の形状を利用したインタラクションが提案されている。この研究では接触面の形状に加えて、オプティカルフローを用いてその移動を追跡した。これを、人間が実世界にて日常的に行っている動作を指入力インターフェースでのインタラクションへと応用した。Wilson ら [98] も接触面の情報を活用したが、彼らは手や指の接触面だけでなく、ディスプレイ面に接する物体の接触情報を利用することを試みた。彼らは指入力インターフェースでの物体を用いたタンジブルな入力を可能にした。

Wang ら [93] はマルチタッチインタラクションにおいて、指の接触面が持つさまざまな属性について調査を行った。その結果、たとえばターゲット選択タスクは人差し指、中指、薬指の方が親指と小指と比較して正確に行えることや、接触面の形状と大きさ、方向が有用な属性であることがわかった。それらの結果を基に、彼らは指の方向を活用したインタラクション [92] を提案した。接触面形状の静的な情報から指の向きを検出し、さらに指がディスプレ

イに触れるまでの動的な接触面の形状変化を利用して指先の方向を検出するアルゴリズムを開発した。彼らはこのアルゴリズムを、指の向きでターゲットを絞り込むターゲット選択方法や、操作するユーザの位置推定等のアプリケーションへと応用した。

透明弾性体を用いて、指の角度や姿勢等のタッチ情報を検出する手法も提案されている。Photoelastic Touch [72] では、液晶ディスプレイと偏光フィルム、透明弾性体を利用したタッチディスプレイを開発した。水平に置いた液晶ディスプレイに偏光フィルムを貼り付け、その上に透明弾性体を配置した。さらに、液晶ディスプレイの上空には、レンズに偏光フィルムが貼り付けられたカメラが設置してある。このようにデバイスを配置することで、偏光フィルムの効果でカメラの映像には液晶ディスプレイの映像は遮断されるが、透明弾性体に力を加えることで偏光の向きが変化し、その周辺から光が漏れる。そして、漏れた光から指の向きや圧力を検出する。同様に透明弾性体を用いた研究として、ForceTile [32] や GelForce [89] 等も存在する。

Moscovich [55] はウィジェットの操作に指の接触面を利用する手法を提案した。指の接触面から計算した 1 ピクセルを用いてウィジェットを操作するのではなく、指の接触面の下にある全てのウィジェットを操作するという方法をとった。この方法では意図しないウィジェット操作が発生してしまうため、タッチとスライドにより操作を行う Sliding Widgets を開発した。ウィジェットが狭い範囲に多く並んでいる場合、スライドの向きを工夫して配置することで隣り合うウィジェットの誤操作を防ぐことができる。

これらのタッチ情報を拡張する研究では、デバイスや取得方法は異なるが、ほとんどの研究においてタッチする指の向きや強さ、接触面の形状を属性として利用している。一方、本研究ではタッチ情報をマクロに捉え、“指の違い”というタッチ情報を利用する。指の違いを利用するアプローチと接触面の情報を利用するアプローチは排他的ではなく、組み合わせて使うことも可能である。これらを同時に利用することで、より多くの入力情報を創出できる可能性を持つ。

### 3.3.3 物理ウィジェットの導入

指入力インターフェースは指だけで操作するため、入力に利用できる情報が少ない。また出力面を考えると、多くの指入力インターフェースではディスプレイ面に映されたオブジェクトを指で操作するだけであるため、その操作には物理的フィードバックが欠けている。この入出力の問題に対して、操作体系に物理ウィジェットを導入する研究が行われている。

Fitzmaurice ら [17] は Graspable User Interface のコンセプトとして、指入力インターフェース上で物理オブジェクトを利用して行うインタラクションを示した。彼らは物理オブジェクトとして Bricks と呼ばれる物体を開発し、それを利用してディスプレイに表示されたオブジェクトの移動やスケーリング等を行う応用例を示した。

VoodooSketch [6] では、指入力インターフェース上で使用するボタンやスライダが開発された。その具体的な利用シーンとして、これらのウィジェットを組み込んだドローリングツールが示された。SLAP Widgets [94] も同様に指入力インターフェース上で使用するウィジェットを提案した。SLAP Widgets は半透明であり、ウィジェットのラベルを動的に変更できる点で

VoodooSketch と異なる。SLAP Widgets はスライダとノブ、キーボード、ボタンで構成される。たとえば、ディスプレイ上にボタンウィジェットを配置すると、動的にボタンのラベルが表示され、キーボードウィジェットを配置すると、同様にキーラベルが動的に表示される。ユーザは通常の物理的なボタンやキーボードと同じ感覚でそれらのウィジェットの操作が行える。SLAP Widgets の底面には赤外線を反射するマーカが貼り付けられており、このマーカと Diffuse Illumination (DI) 技術 [50] を用いてウィジェットを認識している。また、Fiebrink ら [16] は物理コントローラとテーブルトップ上の仮想オブジェクトを動的にマッピングするシステムを開発した。その他にも指入力インターフェース上で物理ウィジェットを使用する研究としては、Tangible Workbench [36] や DataTiles [69]、TangibleTiles [91] 等が存在する。

Audiopad [59] は楽曲演奏のためのシステムである。位置と向きを検出できる“puck”型のコントローラをディスプレイ上で用いて操作することで、リアルタイムに楽曲を合成できる。その他にも、指入力インターフェースを用いた楽曲演奏のためのシステムとして reacTable [31] がある。

物理ウィジェットを導入するアプローチは入力情報を増加させるだけではなく、指入力インターフェースに欠けている物理的なフィードバックの提供を可能にするという利点がある。一方で、指だけで操作可能であるという指入力インターフェースの大きな特徴を損なってしまう可能性がある。本研究で提案する、指の違いを利用するというアプローチは指だけで操作するという特徴を保持できる。よって、従来の指入力インターフェースの使い勝手や操作性を保持したまま、入力情報を増加させることが可能である。また、本アプローチは、物理的フィードバックがないためディスプレイを視認しなければ操作できないという従来の指入力インターフェースの欠点を補うことも可能になる。これについては、7.3.2 節にて詳細を述べる。

### 3.3.4 身体を識別するインタラクション

FSI は身体のうち指を識別するというコンセプトであるが、本研究のように身体を識別するインタラクションはいくつか提案されている。DiamondTouch [15] ではタッチする人の識別を行う。複数人が同時にタッチしてもそれぞれのユーザの操作は保持され、ユーザ間で操作の衝突は発生しなくなるという利点を持つ。FSI は指の識別を行うため、DiamondTouch の利点を包含するといえる。

また、いくつかの研究では本研究と同様に指自体に着目している。たとえば、タッチする指の数を利用して線形メニューの操作に応用する研究 [3] や、ジェスチャ操作に指の組み合わせを組み込んだメニュー操作に関する研究 [44] がある。FingeRing [116] はあらゆる場所で指先でのタッチ入力が可能な装着型のキーボードがあるが、これもタッチする行為自体が意味を持つという観点では FSI のコンセプトに近い。また、Sugiura らにより、指紋を利用したインターフェース、Fingerprint User Interface(以下、FUI) [78] が提案されている。FUI は各指にコマンドやデータを割り当てるというインターフェースであり、インタラクションには単一の指を利用する。FUI は操作対象として物理的なボタン等を対象としており、操作に複数の指を利用することを想定していない。

本研究で提案する FSI は、タッチする指を識別するというインタラクションであり、これらの研究を包含するコンセプトであるといえる。FUI は物理的なボタン等を対象とした、単一の指によるインタラクションを提案している一方で、本研究で提唱する FSI は指入力インターフェースを扱っており、単一の指だけでなく、複数の指を用いたインタラクションとその応用について検討を行っている点で差異がある。

## 第4章 空中での手の動作を利用したインタラクション手法

併行動作のペン入力インターフェースへの適用例として、空中での手の動作を利用したインタラクション手法 [79, 80, 81, 119, 120, 121] を創出した。本章では、まず、ペン入力インターフェースについて説明する。次に、併行動作として採用した動作である、空中での手の動作について述べる。そして、その動作から成る操作として、三つのインタラクション手法を説明し、それらのインタラクションを実現するために開発したペン入力インターフェースについて述べる。次に、空中での手の動作を利用したインタラクション手法の応用例としていくつかのアプリケーションを紹介する。最後に、空中での手の動作を利用したインタラクション手法の操作性に関する評価実験について述べる。

### 4.1 ペン入力インターフェース

ペンは誰もが幼少のころから利用してきた道具であり、もっとも使い慣れた道具の一つである。多くの人はペンを与えられると、迷うことなくペンを正しく使うことができる。ペン入力インターフェースはペンを模したペン型デバイスを用いてディスプレイ面とインタラクションを行うインターフェースであり、そのインタラクションスタイルは実際のペンと紙の利用スタイルを模している。

#### 4.1.1 ペン入力インターフェースの利点

実際のペンと紙の利用スタイルを模しているため、ペン入力インターフェースは多くの人にとってマウスやキーボード等のコンピュータ独自のデバイスと比較して、学習にかかる時間や労力が少なく、扱いやすいインターフェースであるといえる。

また、ペン入力インターフェースの最大の利点は、ストローク入力のしやすさである。マウス等の入力インターフェースと比較して、ペン入力インターフェースを用いることで、ユーザは容易に意図通りのストロークを入力することができる。よって、コンピュータを利用して手書きで文字を書く、絵を描く等の創造活動にはペン入力インターフェースが頻繁に利用されている。

#### 4.1.2 ペン入力インタフェースの課題

ペン入力インタフェースには欠点や課題も存在する。ペン入力インタフェースは1点を正確にタップするような操作は得意としている。これは、ペンをディスプレイに近づけた際にペン先によってユーザの視線が遮られてしまうことや、ペン先をディスプレイ面に下ろす過程でペン先の位置がずれてしまうこと等に起因する。この“ズレ”は、ディスプレイパネルの厚さによる視差や、手首や肘が固定されていないことによる手ぶれ等が原因である。よって、マウス等の入力インタフェースでは容易に操作可能なメニューバー形式の階層メニューや、アイコン形式のメニューの操作はペン入力インタフェースでは容易ではない。つまり、このようなメニューを用いて行われるコマンド選択やパラメータ入力等は苦手な操作であるといえる。また、ペン入力インタフェースは大画面ディスプレイ用の入力インタフェースとしても利用されている。たとえば、教育現場では黒板代わりに、会議等ではホワイトボード代わりに利用されている。このような大画面ディスプレイでは、メニュー表示がディスプレイの端に表示されることが多いため、ユーザの立ち位置によっては、メニュー操作は非常に煩わしく感じられる。

このように、メニュー操作に意識や集中力を奪われることで、主となる作業が妨げられており、ペン入力インタフェースの操作性の低下を招いている。

### 4.2 目指すペン入力インタフェース

ペン入力インタフェースの操作性を向上させるために、以下のようなペン入力インタフェースを目指す。

#### 既存のペン入力インタフェースの使いやすさを維持する

既存のペン入力インタフェースで可能な操作を妨害すると、その使いやすさを失ってしまう。よって、既存のペン入力インタフェースで可能な操作を維持するべきである。これを実現するためには、既存のペン入力インタフェースを拡張するというアプローチが有効である。

#### ペン入力インタフェースのみでさまざまな入力が行える

これを実現するためには、ペン入力インタフェースで扱える入力操作の数を増加させることが必要である。単純にペンに多くのボタンを付加することでもこれを実現することは可能である。しかしながら、このような方法では操作の度にペンを握り直さなければならず、従来のペンの使い方を逸脱してしまい、主となる作業が妨げられてしまう。

### 4.3 空中での手の動作

ペン入力インタフェースの使いやすさを向上させるために、まずペンの形状や使われ方を考慮した新しいペン入力インタフェースの操作方法を考えた。人間がペンを使う際の動作に着目すると、以下の二つに分類することができる。

- ペン先がディスプレイに接している状態での手の動作
- 空中（ペン先がディスプレイに接していない状態）での手の動作

前者はタップとストローク等の操作として、既存のペン入力インターフェースに採用されている。一方で、後者のペン先が紙に接していない状態、つまり空中での手の動作は今までほとんどインタラクションには活用されていない。しかしながら、たとえばシャープペンシルの芯を出すときにペンを振る動作は行われており、空中での手の動作は文房具としてのペンでは有効活用されている。そこで、これらの動作をペン入力インターフェースでのインタラクションに利用することを考えた。

空中での手の動作にはさまざまなものがある。たとえば、ペンまわし、ペンの上部をノックする動作等がある。これらの空中での動作もインタラクションに利用することも可能であると考えられる。しかしながら、これらの動作はペンを持ち替える必要があり、ペンの使い方を大きく逸脱してしまう。また、インタラクションとして利用するため、動作を行うことによりユーザにかかる身体的な負担は最小限に抑えることが望まれる。よって、肩や腕を動かす動作は採用せず、手元を動かすだけで完結する動作に限定すべきであると考えた。

このことを考慮に入れると、インタラクション手法の動作として採用する上では、以下の条件を満たすべきであると考えた。

- ペンを握ったまま行える動作
- 人間にとって無理のない動作

これらの条件を満たす動作として、以下の三つの動作を併行動作として採用した。

- (a) ペンをペン軸回りに回転させる動作
- (b) ペンをペン軸方向に振る動作
- (c) ペンをペン軸に垂直な方向に振る動作

## 4.4 空中での手の動作を利用したインタラクション手法

### 4.4.1 開発したインタラクション手法

採用した三つの動作をインタラクション手法として利用する。それぞれ、インタラクション手法として以下のように呼ぶ。

- (a) rolling
- (b) shaking
- (c) swinging

三つのインタラクション手法を図示したものを図 4.1 に示す。rolling と swinging は回転方向や振る方向の区別が可能である。そこで、rolling は時計回り / 反時計回りの二つの入力、swinging は north, south, east, west の四つの入力としても利用できるように設計した。swinging はさらに細分化して 5 種類以上にすることも可能であるが、細分化することで得られる入力操作の数とユーザビリティを比較して考えると、4 種類が妥当ではないかと考えた。4 種類にすることで一つの入力に対して割り当てる角度が 90 度になり、ユーザは方向を直感的に理解しやすくなる。5 種類以上に細分化して割り当てる角度を小さくすればするほど、ユーザにとって swinging の使い分けが困難になる。

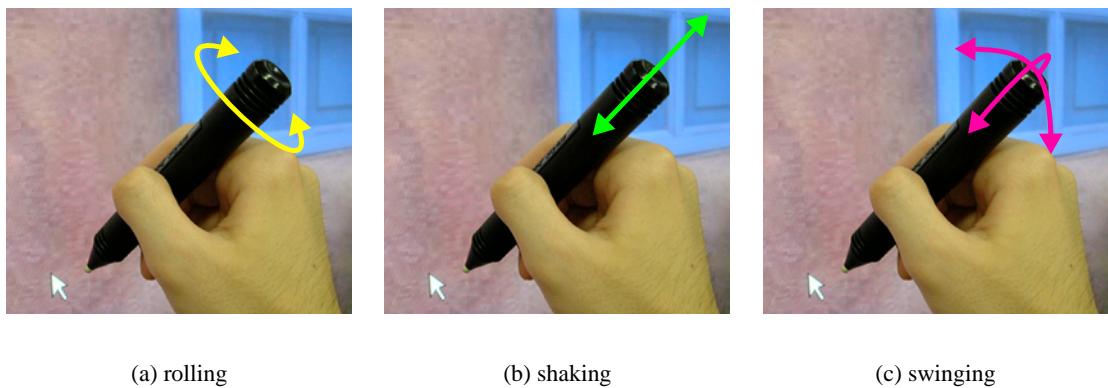


図 4.1: 採用した三つの動作

#### 4.4.2 扱える入力値

各インタラクション手法はその動作の性質上、扱うことが得意な入力と不得意な入力がある。rolling はペンを回転させる動作であり、人間は回転量を自在に制御できる。この回転量を入力として扱うと、連続値の入力に応用できる。また、具体的な回転量は扱わず、回転したという状態変化のみを扱えば、離散値の入力としても応用できる。shaking と swinging は、振る向きは異なるが、両者ともペンを振る動作である。ペンを回転させる動作とは異なり、振り幅を自在に制御することはそれほど容易ではない。よって、振り幅を用いた連続値入力として応用するよりも、離散値入力として応用する方が適切であると思われる。

#### 4.4.3 ペンメタファの導入

これらのインタラクション手法の設計に当たって、ペンメタファというアイデアを導入した。具体的には、振り子式シャープペンシルメタファと 4 色ボールペンメタファという二つのペンメタファを採用した。

振り子式シャープペンシルとはペンを上下に振るだけで芯を出すことができる機構を備えたものである。これはペンを上下に振ることに意味付けがされていると考えることができる。本インタラクション手法では、この特徴を振り子式シャープペンシルメタファとして取り入れ、shaking の設計に応用した。振り子式シャープペンシルはすでに普及しており、多くの人が利用した経験があると考えられる。さらに、このシャープペンシルを使い続けてペンを振って芯を出すことを繰り返すことにより、この行為が習慣化され、無意識的に行えるようになる。このような事象から、本インタラクション手法も使い続けることにより無意識的なインタラクションを実現できることが期待できる。

また、4色ボールペンとはペンの上部に四つのノック部分があるものである。これはペンの4方向に“色”という意味付けがされているといえる。よって、swinging は4色ボールペンメタファを導入していると考えることができる。

#### 4.4.4 既存操作との共存

提案するインタラクション手法の応用の一つとして、メニューバー形式のメニューや小さいアイコン形式のメニューを置き換えることを考えている。これらのメニューは、コンテキストに依らず一定の項目を提示してくれるメニューであるが、4.1.2節で述べたようにペン入力インターフェースでは使いにくい。そこで、提案手法とポップアップメニューを用いて、コンテキストに依存しないメニュー提示や、メニュー操作を行うことを考えた。

従来の操作方法でも、たとえば、バレルボタンと呼ばれるペングリップに付加されたスイッチを押しながらタップする操作や、タップ&ホールド等によりポップメニューの提示は可能である。しかしながら、これらの操作はコンテキストメニューを提示するための操作方法として既に重要な役割を担っているため、既に“予約された”操作方法として扱う。

提案手法とバレルボタン等の両方を用いた操作により、コンテキストに非依存のメニュー提示とコンテキスト依存のメニュー提示の両方が行えるようになる。これにより、ペン入力インターフェースの操作性向上が期待できる。

### 4.5 空中での手の動作を利用したインタラクション手法の実現

提案した三つのインタラクション手法、rolling、shaking、swinging を実現するための実装について説明する。空中での手の動作の取得を可能にするデバイス、Context Sensitive Stylus、および三つの動作を検出するためのソフトウェアを開発した。

#### 4.5.1 Context Sensitive Stylus の開発

空中での手の動作を取得するために、図 4.2 に示すデバイス、Context Sensitive Stylus ( 以下、CS Stylus ) [80] を開発した。CS Stylus はペンの上部に加速度センサを付加したペン型デバイスである。



図 4.2: 開発したペン型デバイス Context Sensitive Stylus

加速度センサとして,Nokia Research Center Tokyo が開発した複合センサモジュール, Cookie を利用した. 図 4.3 に示すように, Cookie はコインサイズのデバイスであり, 3 軸の加速度センサ以外にも, Bluetooth モジュールやバッテリー等から構成される. Cookie が搭載する 3 軸加速度センサは 1 mG 単位で  $\pm 3$  G の範囲の加速度を, 時間分解能 100 ms で検出することができる. Bluetooth モジュールは Bluetooth v1.1 Serial Port Profile に準拠しており, PC と Cookie とのワイヤレス通信を実現する.

ペンの上部に加速度センサを付加することで, ペンの重量バランスが変化することが懸念されるが, Cookie は軽量であるため, ペン型デバイスとしての使用感に与える影響は小さい. CS Stylus はバッテリを内蔵しており, 完全にコードレスな実装が可能である. ケーブル類がなく, 重量もほとんど変化しないため, 通常のペン型デバイスと同等の使用感を得ることができる.

#### 4.5.2 動作の検出

CS Stylus に付加した加速度センサから取得できる 3 軸の加速度値の時間変化を利用して三つの動作の検出を行った.

空中での手の動作が rolling, shaking, swinging, それ以外のどれに該当するかを識別する必要がある. 識別方法として, 加速度値の時間変化の単純な差分比較の利用等, 差分と閾値を用いて動作の識別を試みた. しかしながら, これらの方法では誤認識が多発した.

そこで, 動作の識別にパターンマッチングを応用した. パターンマッチングの手法の中でも, 汎用性が高く, 比較的高精度のマッチングを期待することができる DP マッチングを用いた. DP マッチングとは, 動的計画法を用いて二つのパターンの類似度を計算する方法である. DP マッチングではあらかじめ登録しておく登録パターンと, 照合対象となる照合パターンの

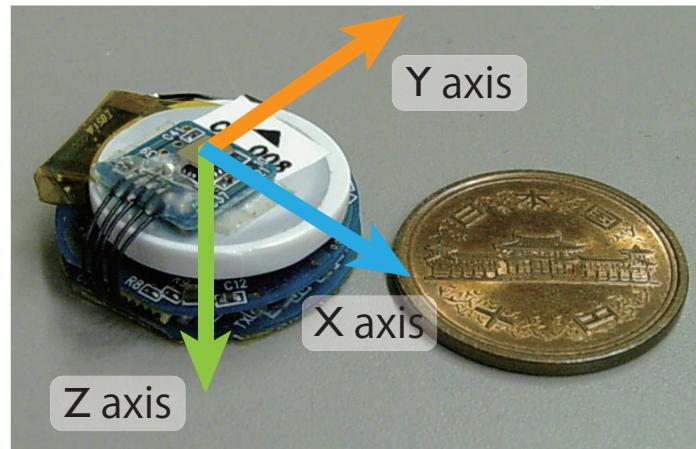


図 4.3: 複合センサモジュール Cookie

二つの数値セットが必要である。登録パターンには全ての動作を登録する必要があるため、登録パターンは rolling は二つ、shaking は一つ、swinging は四つの合計 7 パターンである。DP マッチングを用いた場合の認識精度は、shaking が約 90%、rolling と swinging が 60~70% であった [81]。

## 4.6 空中での手の動作を利用したインタラクション手法の応用

空中での手の動作を利用したインタラクション手法の応用するためには、まず、rolling、shaking、swinging の三つの新しいインタラクション手法をどのように利用するかを検討する必要がある。よって、まず三つのインタラクション手法の特徴を基に、それらの割り当て方法について検討する。そして、開発した三つのアプリケーションを紹介する。

### 4.6.1 割り当て方法の検討

4.4.2 節で述べたように、rolling、shaking、swinging の各インタラクション手法には扱える入力値に得手不得手がある。rolling は連続値入力を得意としている。よって、パラメータ量等を入力する操作に向いている。一方、shaking と swinging は離散値入力を得意としている。よって、コマンド選択等を入力する操作に向いている。rolling は離散値入力も可能であるため、shaking や swinging と同等の操作割り当てにも適している。以下で紹介するアプリケーションでは、原則的にこれらの操作割り当ての指針に従ってインタラクションの設計を行った。

### 4.6.2 実装に用いた技術

全てのアプリケーションの開発には、プラットフォームとして Java をベースとしている。アプリケーション開発に利用したライブラリは以下のとおりである。

- Java<sup>TM</sup>Platform, Standard Edition 6
- Java<sup>TM</sup>Communications API
- Java<sup>TM</sup>APIs for Bluetooth (BlueCove)
- The Standard Widget Toolkit (SWT)

Java<sup>TM</sup>Platform, Standard Edition 6 は Java の標準ライブラリである。Java<sup>TM</sup>Communications API は Java でシリアル通信を行うためのライブラリである。Java<sup>TM</sup>APIs for Bluetooth (BlueCove) は Java で Bluetooth を扱うためのライブラリである。この二つのライブラリを用いて Cookie との通信を実現した。The Standard Widget Toolkit (SWT) は Java 用の GUI ツールキットで、標準ライブラリではサポートされていない UI や機能等を提供する。

#### 4.6.3 ペイントツール

本ペイントツールは描画色の変更、ペンの種類の変更、描画モードの変更等、基本的なペイント機能を備えたアプリケーションである。本ペイントツールのスクリーンショットを図 4.4 に示す。本ツールのメニュー操作インターフェースとして、メニューバーと小さいアイコンを用いた従来形式のメニューに加えて、ペン入力インターフェースでの操作を想定して開発したインターフェース、FlowButton [121] を実装した。FlowButton とはコンテキストメニューの一種であり、図 4.4 の中央に表示されているパネル型のメニューである。FlowButton はメニューがポインタ付近に浮き上がるよう表示されるため、ディスプレイのユーザが操作している場所にメニューが表示される。大画面ディスプレイを用いる場合、従来形式のメニュー表示ではメニュー操作のためにユーザが立ち位置を変更する必要があり、かつメニュー操作自体もペン入力インターフェースでは行いにくかった。一方、FlowButton を用いる場合、ユーザは立ち位置を変更することなくメニュー操作を行え、かつメニュー操作自体もペン入力インターフェースで簡単に行える。

本ツールでは、提案手法を用いて FlowButton の表示やメニュー項目の選択等を行う。各インタラクション手法とそれに対応するメニュー操作を表 4.1 に示す。

表 4.1: ペイントツールにおける操作割り当て

インタラクション手法	割り当てた操作
rolling ( 時計回り )	カラーパレットの表示
rolling ( 反時計回り )	ペンパレットの表示
shaking	トップメニューの表示
swinging ( north )	コピー実行
swinging ( south )	ペースト実行

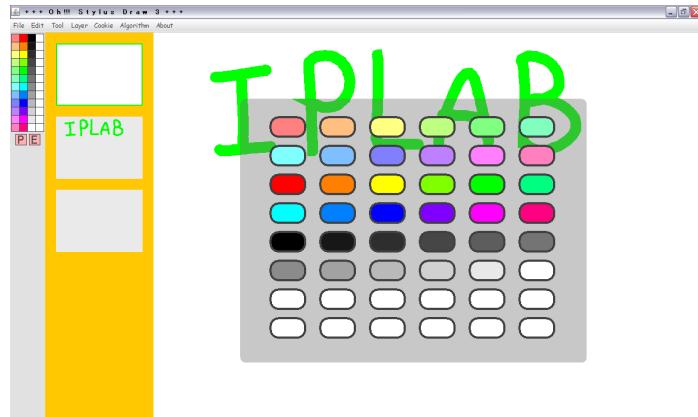


図 4.4: ペイントツールのスクリーンショット . 中央に表示されているパネルが FlowButton メニュー .

本ツールの操作体系について述べる . メニューバー形式のメニューは一般的なメニューインターフェースと同様に , メニューバー , サブメニュー , メニュー項目の順番でタップしていく . 提案手法を用いる場合 , 三つのインタラクション手法とタップを組み合わせて操作を行う . 操作の全体的な流れは図 4.5 のようになる . 基本的な操作手順は以下のとおりである .

1. shaking によりトップメニュー FlowButton を表示する
2. rolling , もしくはタップによりサブメニューを表示する
3. タップによりメニュー項目を選択する

例として , 描画色を変更する操作方法について説明する . 画面遷移は図 4.6 のようになる .

1. shaking によりトップメニュー FlowButton を表示する ( 図 4.6(b) )
2. 時計回りの rolling , もしくは COLOR ボタンのタップによりカラーパレット FlowButton を表示する ( 図 4.6(c) )
3. タップにより色を選択する

従来のメニュー操作で同様の操作を行う場合 , ディスプレイの左端に表示されたツールパネルまで手を伸ばさなければならず , カラーパレットを選択する操作に本来必要のない動作が含まれる .

前述したように , 本ツールでは FlowButton がポインタを中心とした場所に表示される . よって , 大画面ディスプレイで操作を行う場合 , ユーザは立ち位置によってメニュー操作が行いにくくなるという使いにくさが解消されている . また , shaking と rolling により FlowButton の表示やメニュー選択が行えるため , バレルボタンの使用等の煩雑な操作を行うことなく , あらゆるメニュー操作が可能である . その結果 , メニュー操作に余計な意識や集中力を奪われず , 本来の作業に集中することができるようになる .

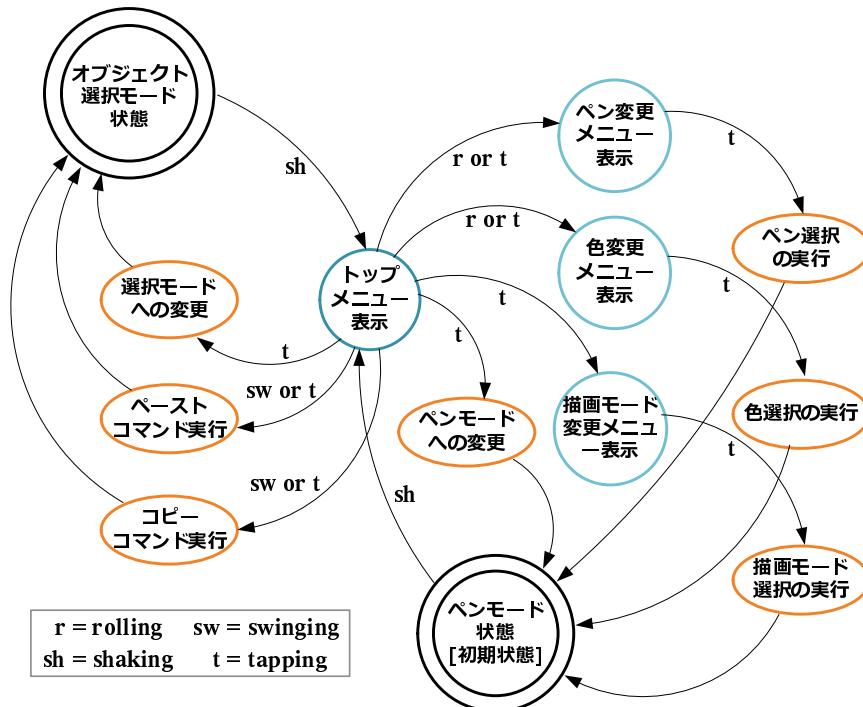


図 4.5: 提案するインタラクション手法を用いた操作体系

#### 4.6.4 スクロール支援ツール

ペン型デバイスにはマウスホイールのような機構が備わっていないため、スクロールを行うためにはスクロールバーを操作する必要があるが、この操作はユーザに細かいポイントティングが要求する。4.1.2節で述べたように、ペン入力インターフェースはこのような操作を得意としていない。また、大画面ディスプレイの場合、ユーザの立ち位置によってはスクロールバーのある位置までユーザ自身が移動する必要がある。このように、ペン入力インターフェースによるスクロール操作は行いやすいとはいえない。

本ツールはペン入力インターフェースでのスクロール操作をより簡単にするためのスクロール支援ツールである[119]。各インタラクション手法とそれに対応する操作を表4.2に示す。

CS Stylusを時計回りにrollingさせると、下方向へスクロール、反時計回りにrollingさせる

表 4.2: スクロール支援ツールにおける操作割り当て

インタラクション手法	割り当てた操作
rolling (時計回り)	下方向へのスクロール
rolling (反時計回り)	上方向へのスクロール
shaking	コンテキストメニューの表示



図 4.6: 描画色を変更し , 文字を描画した際の画面遷移

と上方向へのスクロールとなる . 本ツールを用いると , ユーザはスクロールバーを直接操作しないため , 立ち位置を気にせずにスクロール操作を行うことができる .

また , shaking によりコンテキストメニューを表示することができる . 本ツールを用いない場合 , コンテキストメニューを表示するにはペン先をディスプレイに触れた状態を維持する , もしくはパレルボタン + タップという操作をする必要がある . 一方で本ツールを用いる場合 , ペン入力インターフェースを振るだけでコンテキストメニューを表示することができる .

さらに , このアプリケーションはスクロールバーに限らず , スライダやページ送りの操作等 , マウスホイールによって操作可能なウィジェット操作や機能実行を実現できる . 本ツールにより , マウスで使うことを前提として設計された GUI をペン入力インターフェースでも容易に操作できるようなインターフェースを提供できるようになる .

#### 4.6.5 デジタルノートの拡張

手描き入力が行いやすいというペン入力インターフェースの特徴を活かすアプリケーションとして、デジタルノートが開発されている。デジタルノートとは、電子的に手描きのメモを取るためのアプリケーションである。

デジタルノートとして、たとえば Microsoft Office OneNote( 以下 ,OneNote )がある。OneNote では手書きのメモだけでなく、画像や音声等のデジタル情報を付加することができ、それらを一力所に収集・管理することができる。これにより、すべてのデジタル情報を持つリポジトリを作成することができ、あらゆるデジタル情報の再利用を可能にしている。このように、OneNote は創造活動の生産性を向上させるためのさまざまな機能を持つツールである。

OneNote はマウスを用いて操作することも可能であるが、手書き文字認識やデジタルインク等のペン操作を想定した機能が多く備わっている。しかしながら、インターフェースとしては従来の WIMP インターフェースをベースとしているため、必ずしも創造活動に適したインターフェースであるとはいえない。たとえば、メニュー項目の選択にはウインドウ上部にあるメニューバーからツリー構造になっている階層メニューを辿っていくという操作や、ツールバーにある小さいアイコンをクリックするという操作が必要である。このような操作はディスプレイを注視しながら細かいカーソル移動操作を行わなければならないため、操作に意識を奪われてしまう。さらに、ペン入力インターフェースではディスプレイ上の操作対象を直接ポイントティングする必要があるため、ディスプレイが大きくなればなるほど操作が困難になる。これらは創造活動の障害となりうる。

そこで *rolling* ,*shaking* ,*swinging* を用いて OneNote のインターフェース拡張を行った [79, 120]。具体的には、OneNote の操作体系にこれら三つのインタラクション手法を表 4.3 のように組み込んだ。

*rolling* を行うことで、OneNote に標準で備わっている 13 種類のペンを変更することができる。時計回りの *rolling* により、現在選択しているペンの一つ下のペンが選択され、反時計回りの *rolling* により一つ上のペンが選択される。*shaking* を行うことで、ペンモードと選択モードを切り替えることができる。ペンモードとは、ストロークにより文字や図形等を描くモードであり、選択モードとは、ノート上のオブジェクトを選択するモードである。*shaking* を行うごとに二つのモードが切り替わる。*swinging* を行うことで、コピーとペーストを行うことができる。*north* 方向の *swinging* にコピー、*south* 方向の *swinging* にペーストコマンドを割り当てた。

表 4.3: デジタルノート拡張における操作割り当て

インタラクション手法	割り当てた操作
<i>rolling</i>	ペンの変更
<i>shaking</i>	ペンモード / 選択モードの切り替え
<i>swinging</i> ( north )	コピーコマンド
<i>swinging</i> ( south )	ペーストコマンド

たとえばオブジェクトを複製する場合、まずモードを選択モードに変更する。次に、コピーコマンドを実行し、そしてペーストコマンドを実行するという手順をとる。標準のOneNoteの操作体系では、メニューバーもしくは小さなアイコンメニューからメニュー項目を選択することで、これらのモード変更やコマンド実行を行う。一方で、本インタラクション手法により拡張した操作体系では、これらを shaking と swinging で行うことができる。この拡張したOneNoteインターフェースを利用することで、メニューバーや小さいアイコンを操作することなく、OneNoteの利用頻度の高い機能やコマンド選択を実行できるようになる。その結果、煩わしい操作に伴う思考の中断が減少し、創造活動の効率性向上が期待できる。

## 4.7 評価実験

### 4.7.1 評価の目的

4.1.2節で述べたように、ペン入力インターフェースにおけるメニュー操作等では、意識や集中力を奪われ、主となる作業が妨げられることで、ペン入力インターフェースの操作性の低下を招くことがある。そこで、空中での手の動作を利用するインタラクション手法を提案し、ペン入力インターフェースの操作性向上を目指した。本実験は、“提案するインタラクション手法は従来手法と比較して本来の作業に集中できるか否か”を検証することを目的とする。この検証のために、まず以下の二つの仮説を立て、それらの実証を試みた[81]。

#### 仮説1

従来手法と比較して、提案したインタラクション手法を用いた操作は操作ミスが少ない。  
操作ミスとは、操作時にユーザが起こしたミスである。

#### 仮説2

従来手法と比較して、提案したインタラクション手法を用いた操作は不要な動作が少ない。  
不要な動作とは、移動等の本来の作業やメニュー操作以外の動作である。

### 4.7.2 評価方法

#### 被験者と実験環境

被験者は5人の大学生および大学院生である。22~24歳の男性で、全員右利きであった。また、全員ペン入力インターフェースを操作した経験はあるが、その操作に慣れた被験者はいない。

実験は4.6.3節で紹介したペイントツールを用いて行った。その他、1280×768ピクセルの画面解像度を持つ50インチのタッチスクリーン付きプラズマディスプレイパネル(スマートボード), CS Stylus, 従来のペン型デバイスを用いて実施した。

ペイントツールはフルスクリーンで表示した。トップメニューFlowButtonのサイズは320×274ピクセルで、FlowButton内に表示されるメニュー要素のサイズは96×41ピクセル、カラーパ

レット FlowButton のサイズは  $640 \times 487$  ピクセルで，FlowButton 内に表示されるメニュー要素のサイズは  $64 \times 36$  ピクセルである。FlowButton のサイズは大きすぎるとキャンバスのほとんどが隠れてしまいツールとして使いにくくなり，小さすぎるとタップでの選択操作に支障をきたすため，これらのサイズは画面サイズと使いやすさを考慮に入れて決定している。

### 比較する操作方法

本実験では，三つのメニュー操作方法の操作性を比較した。

- 従来のメニューインターフェースからメニュー操作を行う方式 (従来方式)

図 4.4 の左端に表示されているツールパネルを用いて操作を行う。ツールパネルでは各色のセルをタップすることで，その色に描画色を変更することができる。セルの大きさは一般的なペイントツールとほぼ同サイズにした。

- 提案手法を 1 回用いてメニュー操作を行う方式 (シングル方式)

まず，shaking でトップメニュー FlowButton (図 4.6(b)) を呼び出す。そして，タップでカラーパレット FlowButton (図 4.6(c)) を呼び出し，タップで色の変更を実行する。

- 提案手法を複数回用いてメニュー操作を行う方式 (マルチ方式)

まず，shaking でトップメニュー FlowButton (図 4.6(b)) を呼び出し，rolling でカラーパレット FlowButton (図 4.6(c)) を呼び出す。そして，タップで色の変更を実行する。

### タスクと手順

タスクは 2 種類ある。両タスクとも，あらかじめ描かれた二つで 1 組の矩形を，それと同色の線で繋ぐという作業である。矩形は全部で 5 組あり，それぞれ異なる色で描かれている。タスクにより，5 組の矩形が表示される位置が異なる。タスク 1 ではディスプレイの左側，つまりツールパネルにアクセスしやすい場所に表示され，タスク 2 ではディスプレイの右側，つまりツールパネルにアクセスしにくい場所に表示される。被験者はこれらのタスクを三つの方式でそれぞれ 1 回行う。それぞれの試行において，被験者がタスク達成に要した時間，操作の認識エラー回数，操作の認識エラーによる遅延時間，操作ミスの回数，操作ミスによる遅延時間，操作中の移動歩数を計測した。これらを比較することで，仮説の実証を試みる。

評価実験の手順について説明する。まず，被験者にはペイントツールの使い方やインターフェースの説明を行った。そして，被験者が CS Stylus，および提案手法の扱いに慣れるまで自由に使用してもらった。練習時間に制限を設けなかったが，どの被験者も概ね 10 分程度の練習を行っていた。全被験者の実験の様子はビデオカメラで撮影し，実験終了後に操作の誤認識の回数や遅延時間等を計測した。2 種類のタスクの順番と，それぞれのタスクで行うメニュー操作の順番はそれぞれの被験者にあらかじめ指示をした。なお，これらの順番はカウンターバランスを考慮して決定している。また，実験終了後には操作性に関するアンケートに回答してもらった。

### 4.7.3 結果と考察

表4.4に操作方法及びタスクごとの実験結果を示す。値はすべて5人の被験者の平均値、括弧内は標準偏差である。これらの値は小数点第二位で四捨五入している。

まず、操作ミス回数について議論する。操作ミス回数とは、具体的にはツールパネル上のカラーパレットやFlowButtonから一つのアイテムを選択するときにユーザの意思とは異なるアイテムが選択された回数である。実験の結果、従来手法ではタスク1で4.2回、タスク2で6.4回の操作ミスが発生している。タスク1よりもタスク2で操作ミスが多く発生しているが、これはユーザからツールパネルまでの距離が原因であると考えられる。タスク2では、ディスプレイの右側での作業であるため、ツールパネルを操作するためにはディスプレイの前を移動する必要があり、それにより毎回操作する位置が変化する。よって、安定的な操作が行えず、より多くの操作ミスが発生したと考えられる。また、シングル方式とマルチ方式では操作ミスは全く発生しなかった。これはメニューインターフェースにFlowButtonを採用したことがその要因であると考えられる。FlowButtonではメニューアイテムそれぞれのサイズが大きいため、間違ってアイテムを選択することがない。また、FlowButtonを表示するためにディスプレイに常時表示してあるボタンを使うのではなく、shakingを利用していているため、FlowButtonを表示するときに誤操作する恐れがない。このように、従来方式に比べて提案手法を用いた方式であるシングル方式とマルチ方式の方が、操作ミスが減少していることがわかる。この結果より、仮説1を実証することができた。

次に、移動歩数について議論する。ビデオカメラによる実験後の検証の結果、従来手法でタスク2を行ったときに平均で5.6歩ディスプレイの前を歩いていたことがわかった。ディスプレイの前を歩くという動作は色を変更するという操作には明らかに不要な動作である。一方、提案手法を用いた二つの方式では、いずれの場合も移動歩数は0歩であった。この結果より、仮説2を実証することができた。

以上で二つの仮説を実証することができたが、その他の実験結果についても検証を行う。認識エラー回数について議論する。認識エラー回数とはシステムが認識した動作が、被験者が行った動作と異なっていた、もしくは被験者が動作を行ったときにシステムがそれを動作と認識しなかった回数である。動作の認識を行っていない従来手法では認識エラー回数は0回である。実験の結果、シングル方式ではタスク1と2の平均で4.7回、マルチSA方式では10.0回の動作認識エラーが発生した。認識エラーが発生する原因は二つ考えられる。まず一つ目は被験者がCS Stylusに慣れていないことである。実験後に行ったアンケートの結果、被験者はrollingに戸惑いを感じたことがわかった。rollingはペンを握ったまま行いやすい動作ではあるが、文房具としてのペンでは通常はrollingのような動作は行わない。よって、戸惑いを感じる被験者もいたと考えられる。しかしながら、この問題はCS Stylusの扱いにある程度慣れることで解決できると思われる。二つ目の原因是システムの動作の認識精度である。本研究では動作の認識にDPマッチングを用いているが、マッチングを行うパターン要素数を多く設定することができない。それは加速度情報を取得するために用いているCookieセンサの時間分解能が100 msであり、各動作を行うために必要な時間は高々500 msである。つまり、マッチングに用いることができる加速度情報は5点程度である。この要素数の少なさ

が、マッチング精度が低い原因の一つであると考えている。本研究では高い認識精度を追及するのではなく、あくまで空中での手の動作を用いたインタラクション手法の創出に主眼を置いているため、この問題を完全な解決は行わない。しかしながら、この問題はセンサの性能向上により解決することができるため、将来的な解決は可能であると考えられる。

最後に認識エラーによる遅延時間を除いたタスク達成時間について議論する。遅延を除いた時間についてタスク毎に着目すると、従来手法での試行に比べてシングル方式とマルチ方式の方が、所要時間が短くなっていることが読み取れる。タスク2における従来方式とシングル方式の間には有意水準10%で有意な傾向が見られた( $p = 0.106$ )。この結果より、提案手法を用いることにより、操作速度が向上することが期待できる。本インタラクション手法の目的はペン入力インターフェースの操作性を向上させることであり、操作スピードは追求していないが、本実験により将来的に操作速度の向上も見込まれる結果となった。

表 4.4: タスク及び操作方法ごとの実験結果の平均値・[]内の値は標準偏差。

手法ヒタスクの種類	所要時間(秒)	認識エラー 回数(回)	認識エラー の遅延時間 (秒)	操作ミス 回数(回)	操作ミスの遅 延時間(秒)	移動歩数 (歩)	認識エラーの遅 延時間を除く所 要時間(秒)
従来手法 (task1)	26.7 [10.8]	0.0 [0.0]	0.0 [0.0]	4.2 [1.8]	2.2 [1.9]	0.0 [0.0]	26.7 [10.8]
従来手法 (task2)	33.4 [14.6]	0.0 [0.0]	0.0 [0.0]	6.4 [3.2]	5.2 [4.8]	5.4 [0.5]	33.4 [14.6]
シングル (task1)	31.3 [4.6]	5.2 [1.3]	8.2 [2.6]	0.0 [0.0]	0.0 [0.0]	0.0 [0.0]	23.1 [2.5]
シングル (task2)	29.4 [6.1]	4.2 [2.7]	5.7 [3.6]	0.0 [0.0]	0.0 [0.0]	0.0 [0.0]	23.7 [4.7]
マルチ (task1)	46.3 [12.9]	11.6 [4.6]	19.7 [13.6]	0.0 [0.0]	0.0 [0.0]	0.0 [0.0]	26.6 [3.4]
マルチ (task2)	44.2 [7.5]	8.4 [3.0]	18.4 [8.4]	0.0 [0.0]	0.0 [0.0]	0.0 [0.0]	25.8 [3.9]

## 第5章 ペンを握る指の動作を利用したインタラクション手法

本章では、併行動作のペン入力インターフェースへの適用例として創出した、ペンを握る指の動作を利用したインタラクション手法 [82] について紹介する。第4章では、併行動作として利用した空中での手の動作を利用したが、この動作はタップやストローク等の従来のペインティングインターラクションと同時に遂行することが困難であるという欠点を持っていた。一方、ペンを握る指の動作を利用したインタラクション手法は、従来のペインティングインターラクションと同時に遂行可能な手法である。本章では、まず、ペン入力インターフェースで扱える入力情報について整理する。次に、併行動作として採用したペンを握る指の動作について述べる。そして、その動作を用いたインタラクション手法、Finger Action を紹介する。Finger Action を実現可能にするために開発したペン型デバイスについて説明し、提案するインタラクション手法の応用したアプリケーションについて紹介する。最後に、提案するインタラクション手法の操作性に関する評価実験について述べる。

### 5.1 ペン入力インターフェースで扱える入力情報

一般的なペン入力インターフェースで行える入力操作は以下の三つである。

- タップ（ペン先でディスプレイ面に触れる）操作
- バレルボタンの押下操作
- 筆圧を加える操作

これら三つの中で最も基本的な操作はタップ操作である。タップでは、ペン先がディスプレイ面に接触した状態、およびそのときのペン先のディスプレイ上での2次元座標が入力情報として利用されている。バレルボタンはほとんどの場合、バイナリスイッチとして実装されており、押下されているか否かという状態を入力情報として利用されている。筆圧はペン先でディスプレイ面に力を加える操作である。筆圧の感知が可能なペン入力インターフェースは限られてはいるが、最大で2048段階の感度で筆圧を検出され、それが入力情報として利用されている。バレルボタン操作と筆圧操作はタップ時以外、つまりペンが空中にある状態では動作しない。つまり、従来のペン入力インターフェースでは、ペン先がディスプレイに接していない状態でのインタラクションはできなかった。

第4章で紹介した、空中での手の動作を利用したインタラクション手法は、ペン先がディスプレイに接していない状態でのインタラクションを可能にし、ペン入力インターフェースが持つこの制限を取り払った。

本章では、ペン先がディスプレイと接しているか否かというペン先の状態に関わらず行えるインタラクション手法の拡充を目指し、併行動作を用いたペン入力インターフェースの拡張を行う。

## 5.2 目指すペン入力インターフェース

ペン入力インターフェースの操作性を向上させるために、具体的には以下のようなペン入力インターフェースを目指す。

### 既存のペン入力インターフェースの使いやすさを維持する

既存のペン入力インターフェースで可能な操作を妨害すると、その使いやすさを失ってしまう。よって、既存のペン入力インターフェースで可能な操作を維持するべきである。これを実現するためには、既存のペン入力インターフェースを拡張するというアプローチが有効である。

### ペン入力インターフェースのみでさまざまな入力が行える

これを実現するためには、ペン入力インターフェースで扱える入力操作の数を増加させることが必要である。単純にペンに多くのボタンを付加することでもこれを実現することは可能である。しかしながら、このような方法では操作の度にペンを握り直さなければならず、従来のペンの使い方を逸脱してしまい、主となる作業が妨げられてしまう。

### ペン先の状態に依存せずに利用できる

ペン先がディスプレイに接している状態、および接していない状態の両方でインタラクションが行えることにより、応用可能なアプリケーションや利用シーンが増加する。これを実現可能な併行動作を選択する必要がある。

## 5.3 ペンを握る指の動作

ペンを扱う人間の動作に着目すると、文字や絵を描くためにペンを走らせる動作以外にもさまざまな動作を行っていることがわかる。たとえば、シャープペンシルの芯を繰り出すために、ペンの上部をノックする動作が行われている。このような指の動作は実際のペンでは頻繁に行われているが、ペン入力インターフェースにおけるインタラクションにはほとんど活用されていない。そこで、このような指の動作を利用することを考えた。

指の動作にもさまざまな動作が存在する。ペン入力インターフェースの利点を維持するためにはペンの使い方を逸脱せず、ユーザに身体的負荷をかけない動作であることが望ましい。また、ペン先がディスプレイに接しているか否か関わらず利用可能なインタラクション手法の創出を目指している。よって、採用する動作として以下の条件を満たすべきであると考えた。

- ペンを握ったまま行える動作
- 人間にとって無理のない動作
- ペン先の状態に依存せず遂行可能な動作

これらの条件を満たす動作として、ペングリップを握った状態で遂行可能な以下の三つの動作を併行動作として採用した。

- (a) ペングリップを握る動作
- (b) ペングリップを軽く叩く動作
- (c) ペングリップを擦る動作

ペングリップを握る動作、およびペングリップを軽く叩く動作は実際のペンを使用する場面でも活用されている。筆ペンを使用する場合、インクの量を増やすためにペングリップを強く握る動作が行われている。また、特殊な機構を持つシャーペンシルを使用する場合、芯を繰り出すためにペングリップを軽く叩く動作が行われている。よって、これらの動作は違和感なく受け入れられる動作であると思われる。

#### 5.4 ペンを握る指の動作を利用したインタラクション手法

上述した三つの動作をインタラクション手法として利用する。それぞれ以下のように呼ぶ。

- (a) gripping
- (b) tapping
- (c) rubbing

tapping と rubbing は親指と人差し指の両方で行える。そこで、これらを異なるインタラクション手法として利用する。両者を区別するために、親指での操作を thumb tapping, thumb rubbing と呼び、人差し指での動作を index finger tapping, index finger rubbing と呼ぶ。よって、図 5.1 に示すように、合計で五つのインタラクション手法となる。これらの五つのインタラクション手法を合わせて Finger Action と呼ぶ。

第4章で述べた空中での手の動作を利用したインタラクション手法は、ペン先がディスプレイから離れた状態でのみ行える操作である一方、Finger Action はペン先がディスプレイ面に接した状態、および離れた状態において行えるという特徴を持ち、この点で両者は異なる。Finger Action はペンをディスプレイ面に触れた状態でも行えるため、従来のタップ操作やストローク操作を行いながら遂行できる。よって、特にペイントツール等の手描きを積極的に利用するようなシーンで有用であると期待できる。

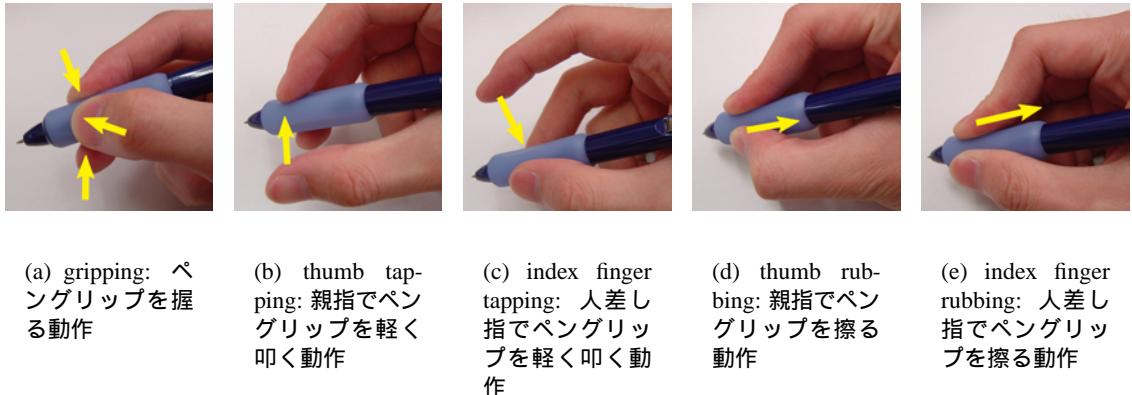


図 5.1: Finger Action: 提案する五つのインタラクション手法

## 5.5 Finger Action の実現

Finger Action を実現するためには、ペンを握る指の動作を検出する必要がある。そこで、ペンを握る指の動作の検出を可能にするデバイス、Pressure-Sensitive Stylus を開発した。また、提案する五つの動作を検出するためのソフトウェアを開発した。

### 5.5.1 Pressure-Sensitive Stylus の開発

人間はペンを握る際に、親指、人差し指、中指の3本の指を使用する[106]。よって、Finger Action を実現するためにはこれら3本の指の動作を検出する必要がある。また、rubbing はペングリップを擦る動作であるため、親指と人差し指の2本の指は縦方向の動作も検出する必要がある。

これらの指の動作を検出するために、Pressure-Sensitive Stylus (以下、PS Stylus) を開発した。PS Stylus の外観を図 5.2 に示す。各指にかかる力を検出するために、インターリンク製の半導体抵抗式圧力センサ (以下、感圧センサ), FSR402 を利用した。図 5.3 に示すように、FSR402 は直径約 18 mm の円形の感圧領域を持ち、その感圧領域に圧力を加えることで電気抵抗が変化する。

PS Stylus のグリップ部には感圧センサが五つ付加されている。ペンを握ったときに親指、人差し指、中指の各指が接触する場所に一つずつセンサを取り付けた。これら三つのセンサにより gripping と thumb tapping, index finger tapping の検出が可能になる。さらに、親指と人差し指に対応するセンサの上部に一つずつセンサを取り付けた。下部と上部のセンサにより、thumb rubbing と index finger rubbing の検出が可能になる。

感圧センサの電気抵抗を AD 変換するために、秋月電子通商の PIC マイコン、PIC-BASIC を利用した。時間分解能 200 ms で AD 変換を行い、PC にデータが送信される。センサと PIC-BASIC を接続する導線として LAN ケーブルを利用した。PS Stylus には感圧センサが五つ付

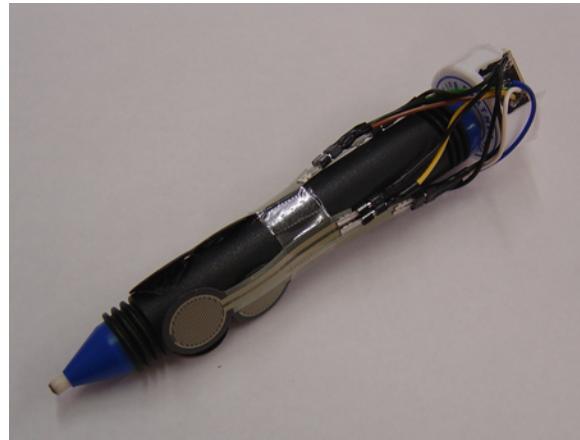


図 5.2: Finger Action を検出するためのペン型デバイス Pressure-Sensitive Stylus

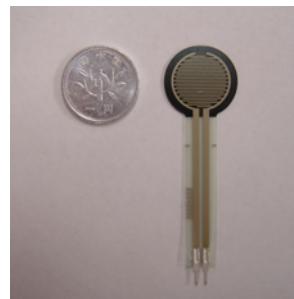


図 5.3: 使用した感圧センサ

加してあるが，グラウンドをまとめることができるため，導線は最低 6 本必要となる。LAN ケーブルは 8 本の導線を持ち，軽量で柔らかいため，これを使用することにした。PIC-BASIC と PC は RS-232C で接続し，シリアル通信でデータを送信する。

### 5.5.2 Finger Action の検出

Finger Action の検出のために，簡単な予備実験を実施した。予備実験では PS Stylus を用いて gripping, tapping, rubbing を行ってもらい，各感圧センサの抵抗値の変化を観測した。その結果，gripping では，三つのセンサの抵抗値がそれぞれほぼ線形に変化し，tapping では，tapping を行う指に対応するセンサ値が大きく変化することがわかった。また，rubbing では，rubbing を行う指に対応する上下のセンサ値が連動して変化することが確認できた。

これらの結果を基に，感圧センサの抵抗値の時間変化と閾値を用いた指の動作検出用ソフトウェアを開発した。このソフトウェア開発には BASIC，および Java を利用した。

## 5.6 Finger Action の操作性評価

### 5.6.1 評価の目的

Finger Action の五つのインタラクション手法 ,gripping ,thumb tapping ,index finger tapping ,thumb rubbing ,index finger rubbing の動作自体は容易に遂行可能であるが , これらがペン入力インターフェースのインタラクション手法として適切か否かはわからない . そこで , Finger Action のインタラクション手法としての実用性を調査するための評価実験を実施した .

### 5.6.2 評価方法

人間にとての Finger Action の行いやすさと , 人間にかかる身体的な負担を測る実験を行った .

#### タスク

実験は二つのタスクから構成される . タスク 1 は , Finger Action の五つの入力操作を行うことだけに集中して操作を行うタスクである . 実験が始まると , 数秒後にスピーカーから短いトリガー音が流れる . 被験者はその音を聞くとすぐに入力操作を行う . 入力操作が成功すると , それを示す音が鳴る . トリガー音は 1 回の試行につき 10 回鳴る . トリガー音が発せられる間隔は 3~6 秒でランダムに決定される . これは , トリガー音の間隔が全て同じであった場合に被験者が音の間隔を学習し , 音のタイミングを予測できるようになることを防ぐためである . この試行を一つの入力操作に対して 5 回ずつ行う . よって , 一つの入力操作につき , 50 回の操作を行うことになる . タスク 2 は , 文字や絵を描きながらタスク 1 と同様の実験を行うタスクである . タスク 1 では操作を単独で行う一方 , タスク 2 では Finger Action とは別の作業を行いながら 5 つの入力操作を行うことになる . つまり , タスク 1 では各操作自体の行いやすさを , タスク 2 では各操作の実用場面での行いやすさを調査する .

二つのタスク終了に , 被験者は操作性や疲労に関するアンケートに答えた . アンケートでは , 操作の行いやすさ , 操作による身体的負担について 5 段階で評価してもらった . 操作の行いやすさは 5 が最も良く , 1 が最も悪い , 身体的負担は 5 が最も楽で , 1 が最も疲れるという評価である .

#### 被験者と実験環境

実験の被験者は 20~25 歳の男性 7 人 , 女性 1 人の合計 8 人である . 全員 Finger Action に関する知識を有していないなかったため , 実験の前に 5 つの入力操作について 5 分程度教えた . その後 , 全操作を数回程度練習してもらった . 実験は一つのタスクを行うために 20 分程度の時間を費やすため , 被験者の疲労を考えて二つのタスクの間には適宜休憩をとってもらった .

実験では , 開発した PS Stylus , および画面解像度が 1280×768 ピクセルの 50 インチのタッチスクリーン付きプラズマディスプレイパネル (スマートボード) を利用した .

### パフォーマンス測定

動作の行いやすさは操作にかかった平均時間と平均操作ミス回数、およびアンケートから評価する。身体的負担はアンケートから評価する。

#### 5.6.3 結果と考察

各入力操作の操作にかかった平均時間とその標準偏差を図 5.4 に、操作ミスの回数を図 5.5 に、アンケート結果を図 5.6 に示す。

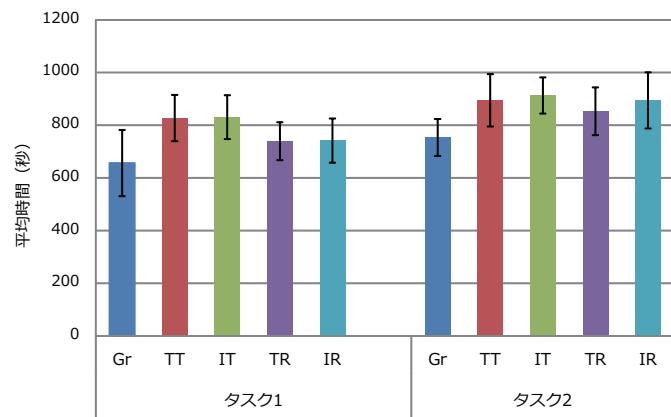


図 5.4: 各操作にかかった平均時間とその標準偏差。Gr, TT, IT, TR, IR はそれぞれ gripping, thumb tapping, index finger tapping, thumb rubbing, index finger rubbing を表す。

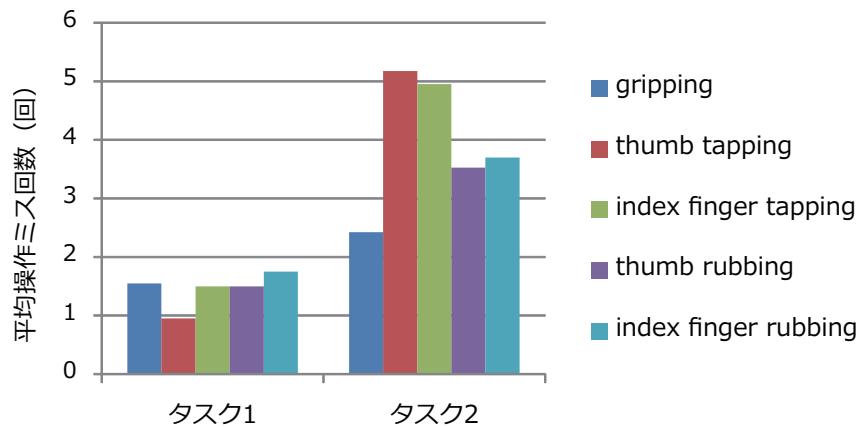


図 5.5: 平均入力ミス回数

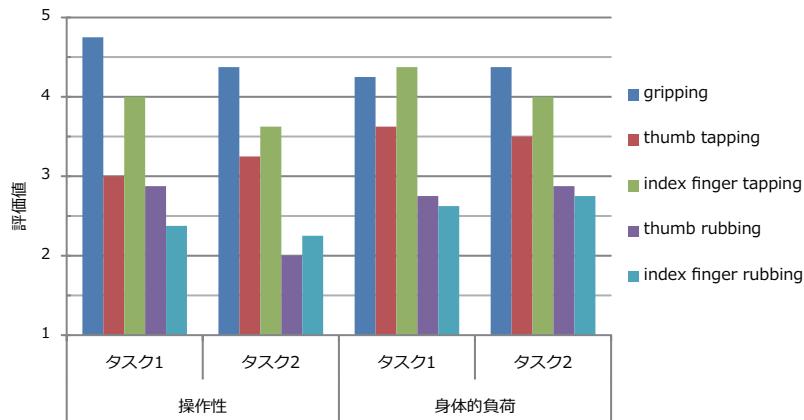


図 5.6: アンケート結果

まず、動作の行いやすさについて検証する。認識ミスや被験者の操作ミスにより、被験者の意図通りに入力操作を行えない試行があった。このような試行は結果から除外した。

ここで、実験結果を検証する上で必要となる人間の認知処理と試作システムの処理時間について説明する。人間の認知処理は、知覚系から刺激を認識し、認知系でその信号の意味的な処理を行い、それを運動系へと伝えるという手順を踏む。今回の実験の場合、聴覚で音を認識し、脳内でその音の意味処理と運動系への反応を決定し、それを基に実際に指を動かすという処理を行っており、この一連の処理には 300~400 ms ほど時間がかかるとされる [12]。また、今回の試作したシステムでは、PS Stylus からセンサ値を読み取る際の時間分解能は 200 ms である。システムが gripping を認識する際には 200 ms 程度の遅延、tapping と rubbing を識別する際には連続した 2 回を使用するため、400 ms 程度の遅延が生じる。つまり、gripping はトリガー音が鳴ってから 500~600 ms 程度、tapping と rubbing は 700~800 ms 程度で反応できれば良い結果であるといえる。

タスク 1 の平均時間の結果を見ると、gripping では 600 ms、tapping では 800 ms を若干超えているが、それほど大きな遅延は見られない。タスク 1 のアンケート結果を見ると、操作性については gripping と tapping は平均点の 3 を超えている。よって、gripping と tapping はそれぞれの操作を単独で行いやすい動作であると期待できる。タスク 2 の平均時間の結果を見ると、全ての操作において期待する操作時間よりも 100~200 ms 程度遅延が発生していることがわかる。タスク 2 では、別の作業へ集中している状態から Finger Action を行うため、被験者は作業の切り換える必要がある。よって、この遅延は作業の切り換えによるものであると推測される。タスク 2 のアンケート結果を見ると、操作性については gripping と tapping は平均点の 3 を超えている。よって、gripping と tapping は単独で行う場合よりも 100~200 ms 程度時間がかかるが、操作としては行いやすいといえる。

入力ミスの回数について検証すると、図 5.5 に示すように、Finger Action の全操作においてタスク 1 とタスク 2 の平均入力ミス回数には大きな差があることがわかる。そこで、タスク 1 とタスク 2 の平均入力ミス回数について有意水準 1% で t 検定を行ったところ、tapping に有

意差が認められた ( thumb tapping:  $p < 0.01$ , index finger tapping:  $p < 0.01$  ). 入力ミスの中でセンシングミスの回数は両者に大きな違いはないと考えられるため, tapping は別の作業中に行うと入力操作のミスが多くなることがいえる。

次に, 身体的負担について検証する。図5.6のアンケート結果を見ると, タスク1, 2ともに gripping と tapping が 3.5 を超えていることがわかる。よって gripping と tapping は, 単独で操作だけを行う場合も別の作業を行いながら操作する場合も, 身体的負担を感じることなく操作できることが期待できる。

結果について整理する。gripping は単独で行う場合と, 別の作業中に行う場合の両方で実用性が高い入力操作であるといえる。tapping は gripping と同様に単独でも別の作業中に行う場合でも実用性が高いといえる。しかしながら, tapping は別の作業中に行う場合に多くの操作ミスが発生してしまう。これは tapping が指を一度ペンから離す動作であるためであると思われる。gripping と tapping について入力操作とアプリケーションの機能とのマッピングを考えると, gripping は別の作業中に行う必要があるような機能に, tapping は単独で行うような機能に割り当てると良い。

rubbing は現時点では実用的であるとはいえない。rubbing が行いにくい原因について, 複数の被験者はセンサの配置の悪さを挙げている。今回の実験で用いた PS Stylus には感圧センサを再配置できる機構は存在せず, 被験者の手の大きさ等を考慮に入れたセンサの配置が行えなかった。よって, 手の大きさを考慮に入れてセンサを配置することにより, rubbing の実用性の改善が期待できる。

## 5.7 Finger Action の応用

ペン入力インターフェースが最も役に立つ利用シーンの一つは文字や絵を描く作業の支援である。そこで, Finger Action のアプリケーションとしてペイントツールを開発した。

### 5.7.1 ペンメタファの導入

人間が普段行っているペンの使い方にできるだけ近いインターフェースを実現するため, ペンメタファを採用した。たとえば, 筆ペンを用いて書字を行う場合, ペンを強く握ることでインクの量を減らし, 弱く握ることで増やすことができる。また, 筆に付着した大量のインクや絵の具は筆を軽く叩くことで散布することができる。本ペイントツールにはこのような筆ペンメタファや筆メタファを導入した。

### 5.7.2 開発したペイントツール

開発したペイントツールの外観を図5.7に示す。メニュー パネル等の GUI は一切存在せず, 全てのメニュー操作は PS Stylus により行える。Finger Action の各インタラクション手法とペイントツールの機能との対応関係を表5.1に示す。インタラクション手法とツールの機能と



図 5.7: 開発したペイントツール

表 5.1: ペイントツールにおける操作割り当て

インタラクション手法	割り当てた操作
gripping	インクの量の調節
index finger tapping	インクの散布
index finger rubbing	インク色の変更

の割り当ては、それらの対応関係が出来る限りイメージしやすいように考慮した。また、5.6節の実験で得た、各手法の適性も考慮に入れた。

それぞれの機能について説明する。gripping には筆ペンメタファが応用されている。実際の筆ペンは強く握ることで多量のインクがにじみ出てくる。本ツールでは、PS Stylus を強く握りながら描くと太い線になり、弱く握りながら描くと細い線になる。PS Stylus を握る強さに応じて描かれる線の太さは線形に変化する。gripping は描画作業中に行いやすいことも、このように機能を割り当てた一つの要因である。

index finger tapping には筆メタファが応用されている。インクや絵の具を大量に付けた筆は、筆を軽く叩くとインクや絵の具が紙面に飛び散る。本ツールでは、この筆メタファを利用してあり、PS Stylus を軽く叩くことにより、インクをキャンバスに散布させることができる。PS Stylus を叩く強さに応じて、飛び散るインクの量やペン先から飛び散ったインクまで距離が変化する。tapping はそれ単独で行うときに操作ミスが少ないことも、このような機能割り当ての参考にした。

rubbing は実際のペンで一般的に行われている動作ではないため、index finger rubbing と機能のマッピングはペンメタファを利用してイメージしやすいマッピングにすることが難しい。そこで、利用頻度が高いことが予想されるインク色の変更機能を index finger rubbing に割り当てた。index finger rubbing を行うごとにインクの色相が一定の割合で変化する。

一般的なペイントツールでは、本ツールで実装しているような簡単な機能であっても必ず GUI を用いたメニュー操作を必要とする。しかしながら、本ツールでは Finger Action を利用することでメニュー操作なしに三つの機能を使うことができる。一般的に簡単であるとはいえないペン型デバイスと GUI を用いたメニュー操作が必要ではないため、ペン入力インターフェースを用いたコンピュータ操作の操作性向上が期待できる。

# 第6章 ペンを握る動作を利用したインタラクション手法

第5章では、ペンを握る指の動作を利用したインタラクション手法として、gripping, thumb tapping, index finger tapping, thumb rubbing, index finger rubbing の五つのインタラクション手法を創出した [82]。これらの操作性について評価した結果、gripping は特に評価が高く、インタラクション手法としての高い潜在能力を確認することができた。gripping とは、ペングリップを握る指の動作のうち、ペンを握る動作を利用したインタラクション手法である。本章では、gripping に着目し、ペン入力インターフェース向けのインタラクション手法としての可能性を探究する [83, 122, 123, 124]。まず、gripping の特徴について整理し、gripping 専用のデバイスとして改良を加えた Pressure-Sensitive Stylus について紹介する。そして、gripping が扱える入力を離散値入力と連続値入力にわけ、それぞれを実現するために必要となる調査を、三つの実験を基に行った。最後に、gripping の離散値入力と連続値入力を用いたアプリケーションについて紹介する。

## 6.1 gripping の潜在能力

第5章では、併行動作としてペンを握る指の動作を利用したインタラクション手法、Finger Action について紹介した。Finger Action は gripping, thumb tapping, index finger tapping, thumb rubbing, index finger rubbing の五つのインタラクション手法から構成され、ペン先の状態、つまりペン先がディスプレイに接しているか否かに関わらず操作できるという特徴を持つ。五つのインタラクション手法の操作性について検証を行った結果、gripping がペン先の状態に依存せずに高い操作性を持ち、インタラクション手法として高い潜在能力を持つことが確認できた [82]。

本章では、ペン入力インターフェース向けのインタラクション手法としての gripping の能力について詳しく調査する。

## 6.2 ペンを握る動作を利用したインタラクション手法

ペンを握る動作を利用したインタラクション手法、gripping について、これを行うための動作、およびインタラクションの観点から整理する。

### 6.2.1 ペンを握る動作の特徴

人間はペンを使うときに、図 6.1 のようにペングリップを親指、人差し指、中指の 3 本の指で握る [106] が、握る強さはほとんど意識していない。一方で、人間の指先には高密度の受容体が存在し、身体の中でも特に鋭い感覚を持つ [105] ため、ペンを握る動作は細かな制御が可能な動作であることが期待できる。

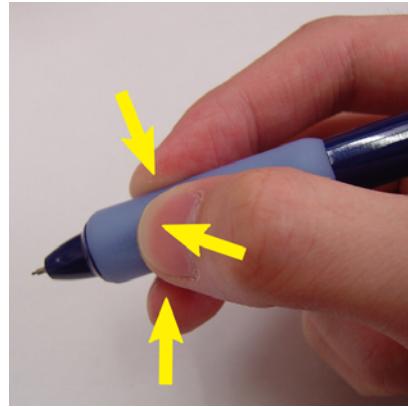


図 6.1: ペンを握る指の様子

また、5.3 節で述べたように、ペンを握る動作はペンを使用する際に行う動作である。よって、新しい動作を学習する必要がなく、普段行っている何気ない動作を少し意識するだけで行えるという利点がある。

ペンを握る動作はその動作単独で遂行できる一方、ペンを握る動作はペンを使用する際に行う動作であることから、文字や絵を書きながらあっても遂行できる。つまり、ペン先がディスプレイに接しているか否かというペン先の状態に関わらず遂行できるという特徴を持つ。

### 6.2.2 gripping

gripping はペンを握る動作を利用した、ペン入力インターフェース向けのインタラクション手法である。ユーザはペンを握る力を加減することで gripping を行う。

gripping はペンを握る動作を用いるため、ペンの持ち方や使い方等、ペンの使用により得た経験を活用できる。このことがさまざまな利点を生み出す。ペンを握る強さは細かく調整できるため、gripping では細かな入力の制御が可能になることが期待できる。また、gripping はペンを握る動作という普段ペンを使用する際の動作を利用するため、操作方法の学習にかかるコストが低いと思われる。さらに、gripping はペンがタッチディスプレイから離れた状態でも、接した状態でも操作を行えるため、さまざまな利用シーンにおけるインタラクション手法として適用できる。

gripping はペンを握る力というアナログな情報を利用するため、入力値として連続値をとる。よって、連続値の入力インターフェースとして利用できる。ここで連続値とは、コンピュータ

での処理の都合上，デジタル化するために実装上は離散値であるが，人間がそれを連続値として認知できるだけの十分な細かさの離散値のことを指す．また，ペン入力インターフェースにおいて，筆圧の連続値を離散値としても扱うことは成功しており[65]，grippingも連続値だけではなく，離散値の入力手法としても利用できる可能性がある．

本研究では，grippingを離散値入力，および連続値入力が可能なインタラクション手法として検討を行う．それぞれ入力手法としての利点が存在する一方，それらの実現にはいくつかの検討すべき課題が残されている．

### 6.2.3 異散値入力の利点と課題

grippingによる離散値入力により，直接GUIを操作することなく，キーボードでのショートカットキーのような操作が行えるようになるため，操作性の向上が期待できる．

筆圧を用いて離散値入力を行う手法[65]も存在する．筆圧はペンがディスプレイに接した状態でしか操作が行えないという欠点がある一方，grippingはペンがディスプレイから離れた状態でも行えるため，筆圧に比べて優位性がある．

grippingによる離散値入力を実現するためには，離散値入力を行うための人間の能力を調査する必要がある．具体的には，何段階の離散値を使い分けることが可能か，入力値を確定するためのトリガー操作にはどのような操作が適切かはわからない．そこで，これらを解明することを課題とする．

### 6.2.4 連続値入力の利点と課題

ペン入力インターフェースは座標入力とともに筆圧のような連続値を入力することができる．筆圧による連続値入力はAdobe Photoshop等の商用アプリケーションでも利用されており，絵や文字を書く等の作業を行うにあたって，連続値入力は有用で必要不可欠な入力操作となっている．

grippingも筆圧と同様に連続値入力が行える．grippingと筆圧を組み合わせることで二つの連続値を同時に入力できるようになり，既存の入力インターフェースでは行えなかったような操作が可能になることが予想される．たとえば，二つのパラメータを同時に変更する等の操作が可能になる．

しかしながら，人間がペンを握る動作と筆圧を同時に制御できるか否かはわからない．そこで，これを解明することを課題とする．

### 6.2.5 類似するインタラクション手法との比較

grippingのように，ペン入力インターフェースの入力操作を増やす手段としては筆圧やバレルボタンがある．これらの手法とgrippingの特徴を表6.1に示す．

grippingと筆圧は連続値と離散値の両方を入力することができる一方，バレルボタンは2値の離散値しか入力できない．grippingはペンとディスプレイが接している状態と接していない

表 6.1: gripping と筆圧 , バレルボタンの比較

	入力値	操作場所
gripping	連続値 / 離散値	空中 / 平面上
筆圧	連続値 / 離散値	平面上
バレルボタン	2 値の離散値	平面上 (実装)

い状態のどちらでも入力を行うことができる一方 , 筆圧とバレルボタンはペンとディスプレイが接している状態でしか入力操作が行えない . バレルボタンは実装次第でペンとディスプレイが接していない状態でも操作可能であるが , 筆圧はペン先とディスプレイ面にかかる力であるため , 操作時に必ずディスプレイと接触している必要がある .

筆圧やバレルボタンと比較したときの gripping の利点は , ペン先の状態 , つまりペンがディスプレイに接しているか否かに関わらず , 連続値と離散値の両方を入力できることである .

## 6.3 gripping の実現

gripping を実現するためにはペンを握る力を加減する動作を取得する必要がある . 5.5 節で説明した Finger Action を取得するためのデバイス , PS Stylus によってペンを握る力を加減する動作は取得できるが , 時間分解能が低く , 必要以上のセンサを備えている . そこで , gripping 用のデバイスとして , PS Stylus の改良を試みた .

### 6.3.1 Pressure-Sensitive Stylus の改良

改良した PS Stylus を図 6.2 に示す . PS Stylus は全長 170 mm , グリップ部の太さは 20 mm , 重量は 45 g , ペン先から 90 mm の位置に重心がある . 後述する三つの実験での握る力の計測にはこのデバイスを用いた .

ペン握る際に利用する 3 本の指にかかる力を検出するために , インターリンク製の感圧センサ FSR402 を三つ使用した . FSR402 は直径約 18 mm の円形の感圧領域を持ち , その感圧領域に圧力を加えることで電気抵抗が変化する . ペングリップにマジックテープのループ部を巻き付け , 感圧センサの裏にマジックテープのフック部を貼り付けることで , 図 6.3 のようにペングリップにセンサを固定した . これはセンサ位置を可変にし , ユーザごとに異なる把持位置を微調整可能にするためである . 将来的にはペングリップを全て覆うことが可能な感圧センサの利用を想定している . 三つそれぞれの感圧センサにより検出したアナログ値は AVR マイコン , Arduino Duemilanove 328 によって AD 変換され , PC へデジタル信号として送られる . センサと Arduino を接続する導線として LAN ケーブルを利用した . Arduino と PC は USB ケーブルで接続した .

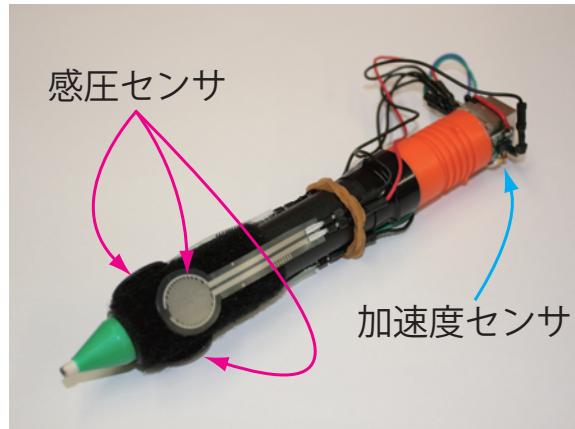


図 6.2: 改良した Pressure-Sensitive Stylus



図 6.3: Pressure-Sensitive Stylus のグリップ部

### 6.3.2 ワイヤレス Pressure-Sensitive Stylus

6.3.1 節で紹介した PS Stylus は PC とケーブルで繋がっており、操作の邪魔になることも考えられる。そこで、ケーブルなしで PC と接続できる、ワイヤレス版の PS Stylus を開発した。

使用したモジュールを図 6.4 に示す。図 6.4 の左から、PC と無線通信するための Bluetooth モジュールである BlueSMiRF WRL-00582、感圧センサから値を読むための AVR マイコンである Arduino Pro Mini 328 5 V 16 MHz、およびバッテリのリチウムイオンポリマー電池である。Bluetooth モジュールは Arduino が AD 変換した後のセンサ値を、Serial Port Profile(SPP) プロファイルを用いて PC に送信する。

これらを小さなボックスの中に入れ、それをペンの上部に付加した。ボックスを取り付けたワイヤレス PS Stylus の外観を図 6.5 に示す。ワイヤレス PS Stylus は全長 175 mm、グリップ部の太さは 20 mm、重量は 95 g、ペン先から 120 mm の位置に重心がある。

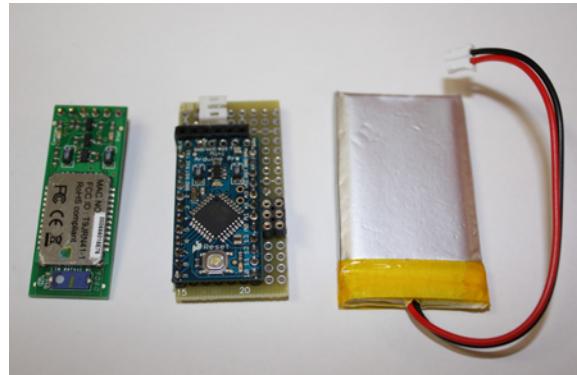


図 6.4: ワイヤレス Pressure-Sensitive Stylus に利用したモジュール

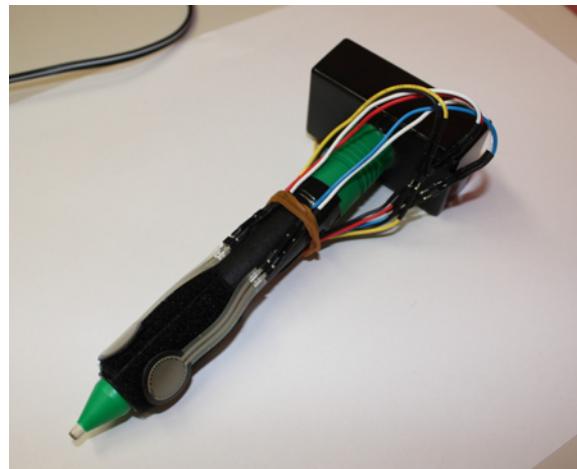


図 6.5: ワイヤレス Pressure-Sensitive Stylus

### 6.3.3 デバイスの設計

人間は指先に強い力を長時間加え続けると、その検知能力は徐々に低下し、力の制御がうまくできなくなる。これは、触圧の検知を行う指先の機械受容器が一定時間の順応後、信号の発火を止めることに起因する [34]。よって、ペンを強く握らなければならない設計は避けるべきである。また、ペン自体にも重量があるため、ペンの保持にも最低限の力が必要となる。そこで、PS Stylus は 30 ~ 500 g 重の力を検出するように設計した。この力は三つのセンサから検出した圧力の平均値を用いて算出している。

検出した圧力は 1024 の分解能で処理される。センサの特性上、センサの出力値と実際にかかる力の関係は線形ではなく対数特性を持つため、ソフトウェアで特性が線形になるように補正した。これにより、センサの出力値と実際の圧力は概ね比例するようになったが、完全な線形特性には補正できていないため、かかる力が弱いときに出力値がやや大きくなる傾向

がある。また、PS Stylus の時間分解能は 20 ms であるため、操作に対する遅延をほとんど感じずに操作することができる。これらは PS Stylus、ワイヤレス PS Stylus ともに共通である。

## 6.4 実験 1: 離散値入力に関する調査

### 6.4.1 目的

6.2.3 節で述べたように、gripping による離散値入力を実現するための課題が存在する。本実験では、gripping により何段階の離散値入力を使い分けることが可能か、また、入力値を確定するためにはどのようなトリガー操作が適切かについて調査する。握る力のような人間の力感覚を用いたインタラクション手法では、フィードバックが重要になると考えられる。そこで、フィードバックの重要性についても同時に調査する [122]。

ここで、後者のトリガー操作の必要性について詳細を説明する。ターゲット選択のような操作を行う場合、選択を確定するためのトリガーが必要であるが、マウスを使った操作では一般的にはクリックがトリガー操作になっている。よって、gripping のトリガー操作としてクリックに相当するタップを使用することも考えられるが、ペンがディスプレイから離れた状態でも操作が完結するという gripping の大きな特徴が失われてしまう。そこで、ペンがディスプレイから離れた状態で行うことができる以下の四つの操作をトリガー操作として検討する。

- *Keeping*

握る力を一定時間（本実験では 1000 ms）維持する操作

- *Quick Release*

ペングリップに加えている力を素早く緩める操作

- *Finger Release*

ペンから人差し指のみを離す操作

- *Swinging*

ペンを振る操作

図 6.6 に、これらのトリガー操作における握る力の変化をモデル化した図を示す。図 6.6 の横軸は時間、縦軸は握る力、オレンジ色の領域は目標入力値、青線は握る力の変化を示す。図 6.6(c) の赤線は人差し指のみの力の変化、図 6.6(d) の緑線は加速度の変化を示す。

*Swinging* は第 4 章で提案したインタラクション手法の 1 つであるが、*Swinging* は空中で行え、かつ手首を回転させるだけで行える簡単な動作で行えるため、今回これを採用した。

### 6.4.2 被験者と実験環境

被験者は 22 ~ 26 歳の男性 6 名、女性 2 名の合計 8 名のボランティアで、7 名は右利き、1 名は左利きである。ペンの持ち方は被験者ごとに多少異なっていたが、全員が 3 本の指を用

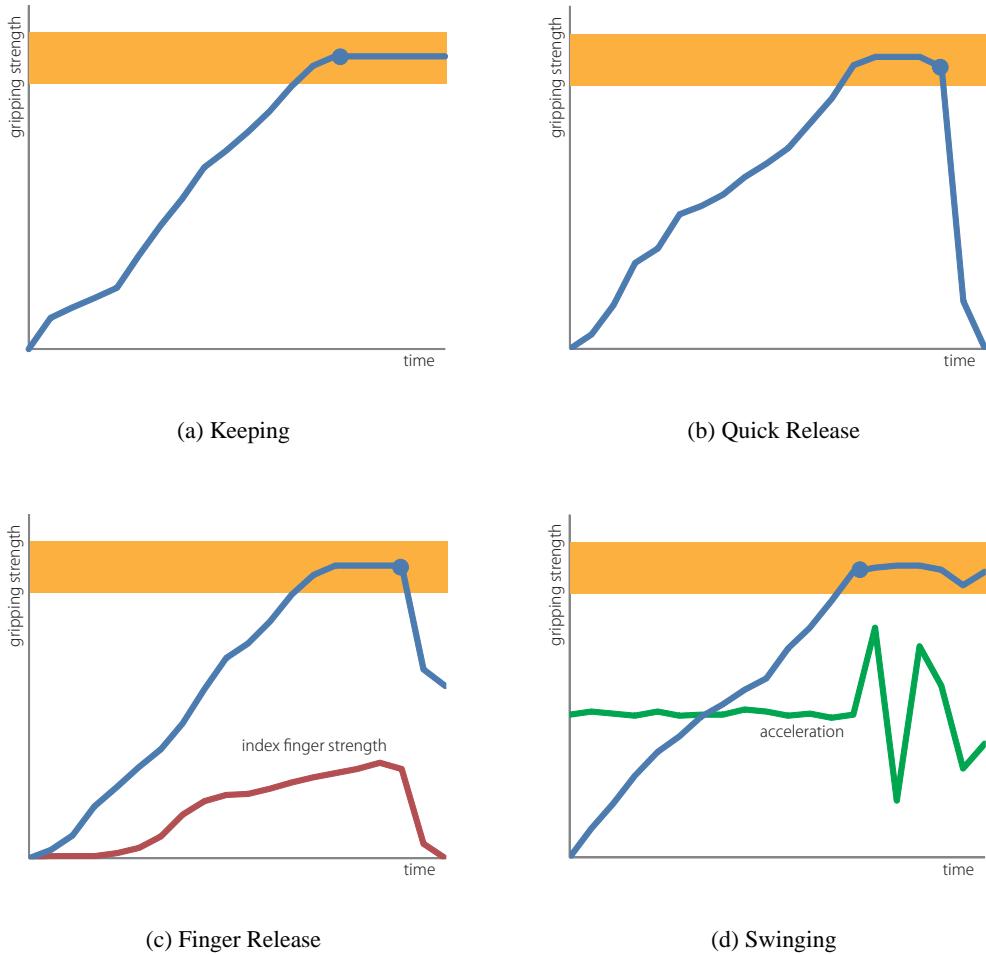


図 6.6: 比較する四つのトリガー操作の力変化のモデル

いていた。被験者が普段通りにペンを握ることができるように、各被験者の握り方に応じて PS Stylus の三つの感圧センサ位置を微調整した。被験者には普段通りのペンの持ち方をするように指示をしたため、極端にペンを立てたり寝かせたりして持つ被験者はいなかった。

*Swinging* を検出するために、ペンの上部に加速度センサモジュールを配置した。このセンサモジュールは図 6.2 のように小型であり、重量が 2 g であるため、PS Stylus の使い勝手に与える影響は少ない。この加速度センサでは、3 軸の加速度を  $\pm 2$  G の範囲で検出できる。実験には PS Stylus 以外に  $1280 \times 1024$  ピクセルの解像度を持つ 20 インチの LCD を用いた。握る力は PS Stylus の三つのセンサの平均値であるが、*Finger Release* を検出するために入差し指に相当するセンサの値を個別に利用した。実験用ソフトウェアは Intel Core 2 Quad 2.83 GHz、メモリ 4 GB、Windows Vista を搭載した PC で実行した。

### 6.4.3 タスク

握る力を制御してターゲットを選択するというタスクを行った。被験者には図 6.7 のようなボックスとカーソルが提示される。図 6.7(a) の最下部に表示されている円形がカーソルである。ボックスのサイズは  $600 \times 800$  ピクセルで、カーソルは握る力に応じてボックス内を図 6.7(b) のように垂直方向に移動する。1024 段階の圧力レベルは 800 ピクセルに均一にマッピングされており、圧力レベルが 0 のときにカーソルは図 6.7(a) のようにボックスの最下部に提示され、1023 のときに最上部に表示される。ボックス内にはターゲットが一つ提示される。被験者はそのターゲット内にカーソルを入れることでターゲットを選択し、その状態でトリガー操作を行うことでターゲット選択を確定する。トリガー操作は *Keeping*, *Quick Release*, *Finger Release*, *Swinging* の 4 種類であり、それぞれ、力を維持し始めたときの力、力を緩める直前の力、人差し指を離す直前の力、ペンを振る直前の力を測定し、トリガーとして利用した。

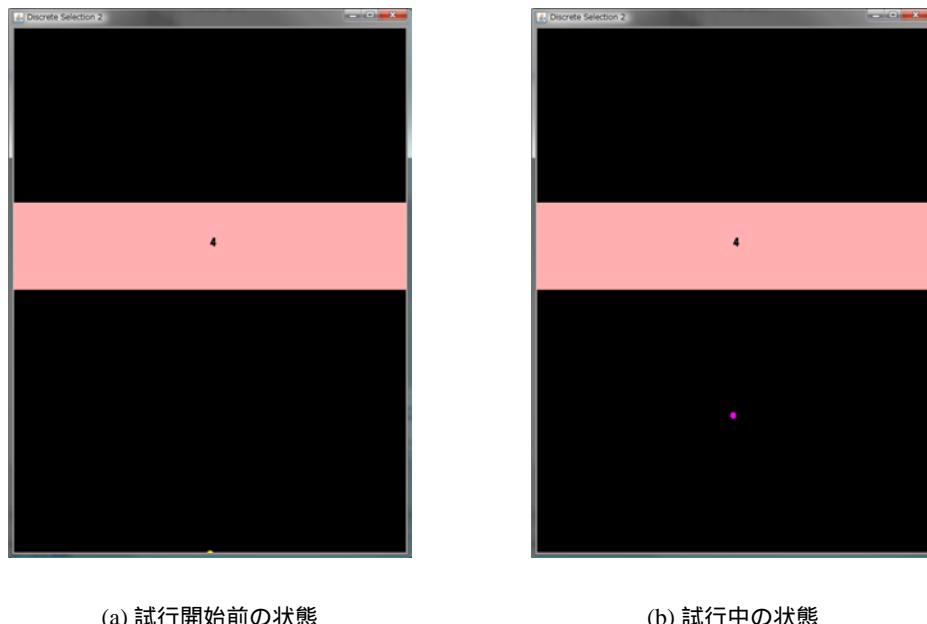


図 6.7: 実験 1 で被験者に提示されるボックスとカーソル

視覚的フィードバックが gripping の操作性に与える影響を調査するために、2 種類のフィードバックを用意した。完全フィードバック（以下、FF）では、ターゲットとカーソルの動きが提示される。カーソルがターゲット内に入るとターゲットの色が変わるという視覚的フィードバックも備える。一方、部分フィードバック（以下、PF）では、初期状態ではターゲットとカーソルが提示されているが、試行が始まるとカーソルは非表示になる。カーソルがターゲット内に入ってもターゲットの色は変化しない。つまり、被験者は自身の記憶と力感覚のみでカーソルをターゲット内に入れることになる。この PF タスクでは、Marking Menus [42] の

ような、コンピュータ熟練者を対象とした eyes-free インタラクションをシミュレートすることを狙う。FF と PF の比較により、gripping におけるフィードバックの重要性の検証も行う。離散値の段階数  $n$  は 2~12 の 11 段階を用意した。各段階のターゲットサイズは  $800/n$  である。 $n$  が大きくなればなるほど微妙な力加減が必要になり、被験者はより慎重な操作が要求される。各被験者はトリガー操作ごとに、11 段階の  $n$  の試行をそれぞれ 6 回ずつ行った。ターゲット選択に失敗した場合、成功するまでその試行を繰り返した。ターゲット選択後、圧力レベルが 0 になるまでペンを握る力を緩めると次のターゲットがセットされ、再び力を加えることで次の計測が開始する。4 種類のトリガー操作を行う順番は被験者ごとに変え、カウンターバランスをとった。まず FF で全試行を行い、その後 PF で行った。PF は gripping 熟練者を対象としているため、初めに FF を行ってもらうことで gripping の経験を少しでも積んでもらう狙いがある。まとめると、この実験では、11 段階 × 6 試行 × 被験者 8 人 × 4 トリガー操作 × 2 フィードバック、合計 4224 回の正解ターゲット選択が行われた。

#### 6.4.4 パフォーマンスの測定

一般にコンピュータへの入力操作は素早く、容易に、正確に行えることが求められる。そこで本実験では、ターゲット選択のエラー率 (ER)、ターゲットをクロスした回数 (NC)、ターゲットの選択時間 (ST) の三つの観点から評価を行う。ER は 1 回のターゲット選択で発生するエラー率、NC はターゲットにカーソルが入った後にカーソルがターゲットの境界をクロスした数 (たとえば  $NC = 2$  は、カーソルがターゲット内に入った後に一旦ターゲットを出て再度入ったことを示す)、ST は被験者が力を入れ始めてからターゲット選択が完了するまでの時間である。ER は正確さ、NC は容易さ、ST は素早さを示す指標として利用する。

#### 6.4.5 結果

NC と ST は正解ターゲット選択の試行から得たデータである。また、 $\pm 2\sigma$  を超えた 24 個の計測値を外れ値としてデータセットから取り除いた。

##### エラー率の分析

エラー率 (ER) を図 6.8 に示す。図 6.8(a) に示す FF のグラフを見ると、*Keeping* が他の手法と比較して明らかにエラー率が低いことがわかる。また、*Keeping* は  $n$  が増加してもエラー率がほぼ 0 で推移する一方で、他の 3 手法は  $n$  が増加するごとにエラー率が増加している。4 つの手法について分散分析を行った結果、有意差があることがわかった ( $p < 0.001$ )。さらに、4 手法のペア 6 組に対して t 検定を行った結果、*Keeping* とその他 3 手法全てのペアに有意差があることがわかった ( $p < 0.001$ )。

一方、図 6.8(b) に示す PF のグラフを見ると、*Keeping* が他の手法と比較してややエラー率が低いことがわかる。また、FF とは異なり、全手法とも  $n$  が増加するごとにエラー率が増加している。FF と同様に分散分析を行った結果、有意差は確認できなかった ( $p = 0.050$ )。

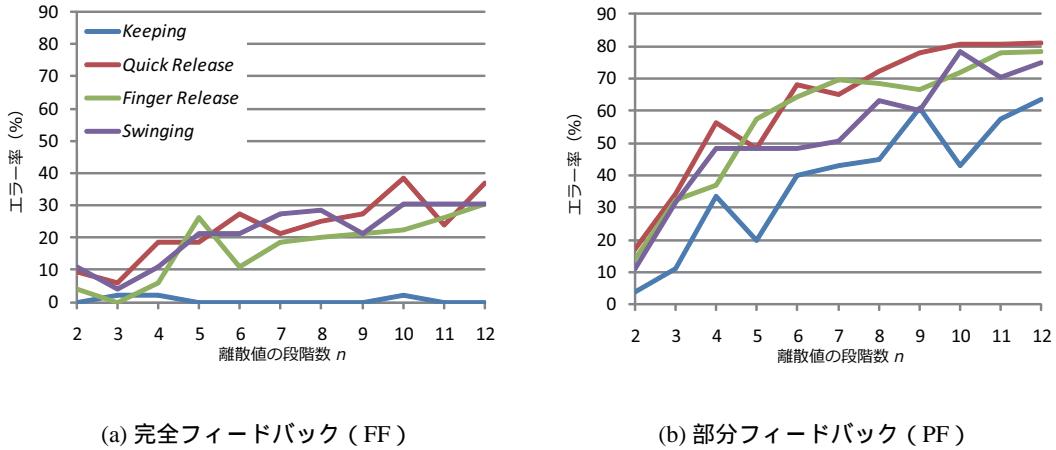


図 6.8: エラー率のグラフ

### クロス数の分析

クロス数 (NC) を図 6.9 に示す。図 6.9(a) に示す FF のグラフを見ると、Quick Release のクロス数が他の手法よりもやや少ないことがわかる。また、4 手法とも  $n$  が増加するにつれてクロス数が指数関数的に増加する傾向が読み取れる。特に  $n$  が 6 を超えてから増加傾向が強い。四つの手法について分散分析を行った結果、有意差は確認できなかった ( $p = 0.313$ )。

また、図 6.9(b) に示す PF のグラフを見ると、 $n$  が 10 以上のときは PF の方が FF よりもクロス数が少ない傾向を読み取れる。四つの手法について分散分析を行った結果、FF と同様に有意差は確認できなかった ( $p = 0.167$ )。

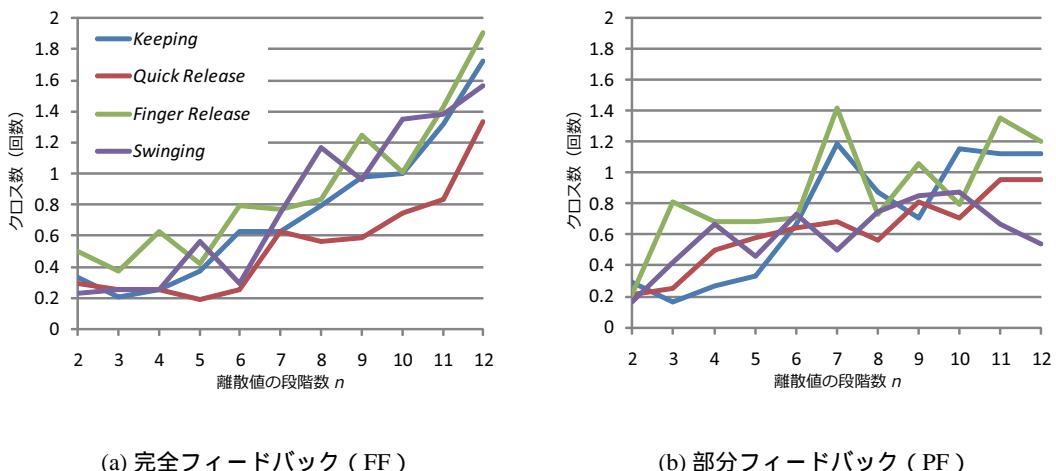


図 6.9: クロス数のグラフ

### 選択時間の分析

選択時間 (ST) を図 6.10 に示す。図 6.10(a) に示す FF のグラフを見ると, *Keeping* が他の手法と比較して 1000 ms 程度多くの時間を選択に費やしていることがわかる。また, 各手法とも  $n$  が増加するにつれて緩やかに選択時間も増加する傾向がある。四つの手法について分散分析を行った結果, 有意差があることがわかった ( $p < 0.001$ )。さらに, 4 手法のペア 6 組に対して t 検定を行った結果, *Keeping* とその他の 3 手法全てのペア, および *Quick Release* と *Finger Release* のペアに有意差があることがわかった ( $p < 0.001$ )。

図 6.10(b) に示す PF のグラフを見ると, FF と同様に, 他の手法と比較して *Keeping* での選択に多くの時間を費やしていることがわかる。一方, *Swinging* はほとんどの場合に最も早く,  $n$  の値によらず安定した選択時間を示している。四つの手法について分散分析を行った結果, 有意差があることがわかった ( $p < 0.001$ )。さらに, 4 手法のペア 6 組に対して t 検定を行った結果, *Quick Release* と *Finger Release* のペアを除いた五つのペアに有意差があることがわかった ( $p < 0.001$ )。

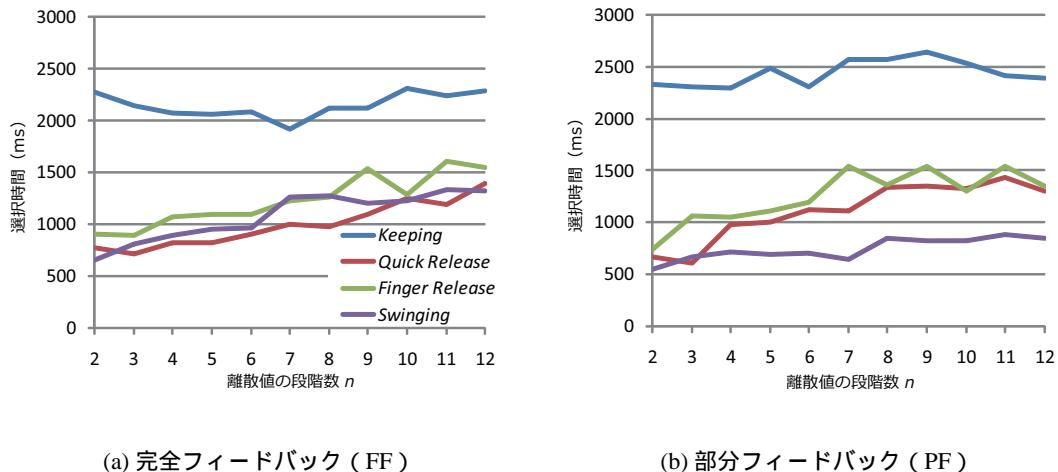


図 6.10: 選択時間のグラフ

### 6.4.6 考察

#### 離散値の段階数 $n$ の決定

FF のエラー率と選択時間は  $n$  の増加につれて緩やかな増加を示しているのに対して, クロス数は  $n$  が 6 を超えると急に増加傾向が強まる。つまり,  $n$  が 6 を超えると操作が困難になる傾向が強いことがわかる。よって, gripping による離散値入力は 6 段階以内で行うことが適切であるといえる。

### トリガー操作の選定

FFのエラー率を比較すると, *Keeping* は  $n$  が増加してもほぼ 0 である一方, その他の手法は  $n$  の増加に伴いエラー率も上昇した. つまり, *Keeping* は  $n$  が増加しても正確に操作できることがわかる. よって, *Keeping* が適切なトリガー操作として有望であると考えた. 選択時間では, *Keeping* は他の手法よりも選択に時間がかかり, その差はおよそ 1000 ms であることがわかった. *Keeping* は握る力を 1000 ms 間維持することで選択を確定する操作であるため, この差は在って然るべき差であるといえる. しかしながら, 1 回の操作時間が 2000 ms とやや長く, フラストレーションが溜まることも考えられる. これについては今後の課題として, 力を維持する時間等の調整による操作時間の短縮を目指す.

### 視覚的フィードバック

PFは熟練者向けのタスクとして設計したが, 今回の被験者には熟練者はいなかった. 図 6.8 に示した FF と PF のエラー率を比較すると, PF は FF の 2~3 倍程度のエラーが発生していることがわかる. 熟練者であれば PF のエラー率が低くなるか否かは確認できないが, 初心者が gripping を行う際にはフィードバックは非常に重要であるといえる. ユーザに対して適切な視覚的フィードバックを与えることで, gripping は有用な離散値入力のための操作になると考えられる.

## 6.5 実験 2: ペンを握る力と筆圧の関係に関する調査

### 6.5.1 目的

6.2.4 節で述べたように, 人間がペンを握る動作と筆圧を同時に制御できるか否かわからぬ. 空中でも強くペンを握ることができることから, 小さい筆圧に対して握る力を大きくすることは容易であると考えられる. 一方, 握る力が小さいままで大きな筆圧を加えることは物理的に困難であると想像できる. 実験 2 ではまず, 握る力と筆圧を同時に制御できる境界を明らかにするために, 最小のペンを握る力で加えることができる筆圧について調査する [83].

### 6.5.2 被験者と実験環境

被験者は 22~26 歳の男性 6 名のボランティア, 5 名は右利き, 1 名は左利きであった. 全被験者とも 3 本の指を用いて正しくペンを握っており, 実験 1 と同様に感圧センサの位置は被験者ごとに微調整した. ペンの握り方の指示も実験 1 と同様に行い, 極端にペンを立てたり寝かせたりして持つ被験者はいなかった.

筆圧を測定するために, PS Stylus に付加しているものと同じ感圧センサをもう一つ利用した. 感圧センサを机の上に置き, 感圧センサをペン先で押さえることで筆圧を測定した. 市販のペンタブレット等を利用して筆圧を測定することも考えたが, センシング特性を揃える

ために gripping と筆圧の測定に同じセンサを利用した。ペンを握る力と筆圧は 20 ms の時間分解能で計測した。

被験者はイスに座った状態でペンを握り、ペン先で机の上に置かれたセンサに圧力を加えた。

### 6.5.3 タスクと測定

被験者はまずペンを軽く握り、ペン先を感圧センサに接触させる。被験者は最小限の力でペンを握り、できる限り大きい筆圧を感圧センサに加えていく。さらに、筆圧が最大になるまで徐々に加える力を強くする。筆圧を強くしようとするとペンを握る力も自然と強くなるため、ペンを握る力と筆圧との関係が明らかになる。各被験者はこの試行を 5 回繰り返した。

### 6.5.4 結果

全計測データをプロットした散布図を図 6.11 に示す。X 軸と Y 軸はそれぞれペンを握る力と筆圧を、A-F はそれぞれの被験者の計測値を表す。

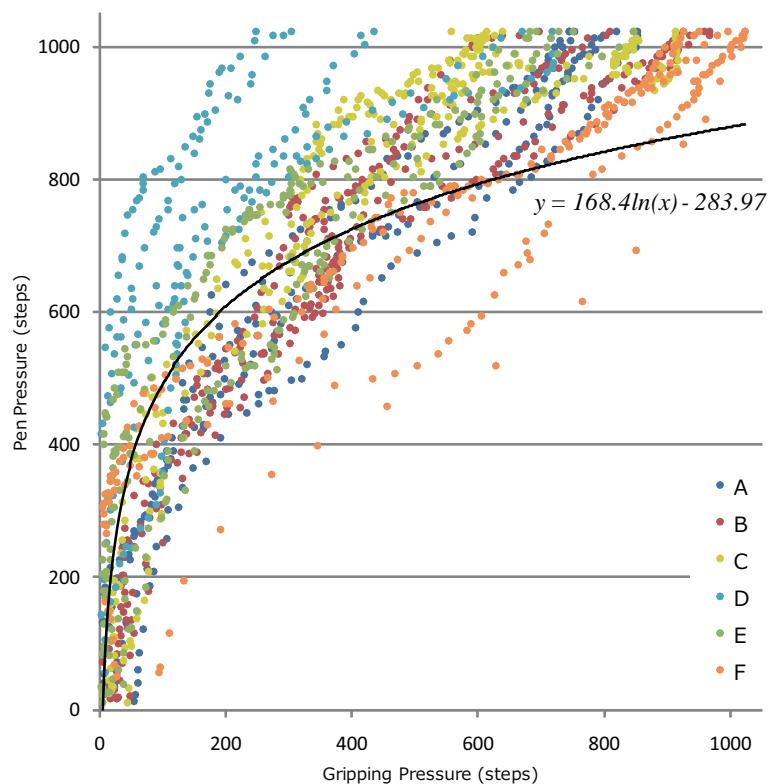


図 6.11: 計測したペンを握る力と筆圧をプロットした散布図

全計測値を分析した結果、ペンを握る力と筆圧は対数に回帰していることがわかった( $R^2 = 0.754$ )。また、各被験者の計測値を分析した結果、被験者 F が最もペンを強く握る傾向があることもわかった。

### 6.5.5 考察

図 6.11 を見ると、ペンを握る力が非常に弱い状態では強い筆圧を加えることができないが、それ以外の状態では二つの力を同時に制御できる可能性があることがわかる。ペンを握る力と筆圧の 2 次元の力空間として捉えると、ある一定の空間内であれば二つの力を同時に制御できる可能性がある。

被験者 F が最もペンを強く握る傾向があった。被験者 F のペンを握る力を  $x$  とおくと、そのときに加えられる最大の筆圧  $y$  は  $y = 168.4 \log x - 284$  で表現できる。よって、ペンを握る力と筆圧の関係の目安としてこの式を採用する。この曲線と X 軸で囲まれた部分が二つの力を同時に制御できる力空間である。二つの関係を単純に  $y = x$  と定義するよりも、特に握る力が弱いときにより広い力空間が利用できる。この結果は、たとえば gripping と筆圧を両方同時に利用するアプリケーションを設計する際に、ある握る力に対してユーザが取りうる筆圧の範囲を設定するために利用できる。

## 6.6 実験 3: 適切な力空間に関する調査

### 6.6.1 目的

6.5 節で述べた実験 2 の結果、人間が同時に制御可能であろうペンを握る力と筆圧の力空間が明らかになった。そこで実験 3 では、力空間の中で人間が快適に制御できる領域を調査する [83]。

### 6.6.2 被験者と実験環境

被験者は 22~26 歳の男性 7 名、女性 2 名の合計 9 名のボランティアで、8 名は右利き、1 名は左利きであった。全被験者とも 3 本の指を用いて正しくペンを握っており、実験 1, 2 と同様に感圧センサの位置は被験者ごとに微調整した。ペンの握り方の指示も実験 1, 2 と同様に行い、極端にペンを立てたり寝かせたりして持つ被験者はいなかった。

筆圧の測定には実験 2 と同様に感圧センサを一つ利用した。被験者はイスに座った状態でペンを握り、ペン先で机の上に置かれたセンサに圧力を加えた。実験には PS Stylus 以外に 1280×1024 ピクセルの解像度を持つ 20 インチの LCD を用いた。

### 6.6.3 タスク

握る力と筆圧を同時に制御し、力空間内の指定された領域をターゲットとして選択するタスクを行った。

被験者には、図 6.12 に示す、被験者が加えている力とターゲットとなる力を示した Pressure Map が提示される。Pressure Map は握る力を X 軸に、筆圧を Y 軸にとった力空間を表すマップである。被験者が加えている力を青色のカーソルで提示する。Pressure Map は左下を原点とし、握る力と筆圧の両方が 0 のとき、図 6.12(a) のようにカーソルは原点（左下）に配置される。被験者はカーソルの近傍に表示されているラベル表示により座標軸と力とのマッピングを常に確認することができる。Pressure Map は  $800 \times 800$  ピクセルで、握る力と筆圧ともに 1024 段階のうち、800 段階 (0–799) を 800 ピクセルにマッピングした。図 6.11 より、800 段階以上の筆圧はほとんど利用できないことがわかるため、このように設計した。Pressure Map は縦横に 5 分割されており、合計 25 の矩形が存在する。これらの矩形が、被験者が選択するターゲットとなる。Pressure Map 上に描かれている赤色の曲線は、実験 2 で求めた、人間が同時に制御可能なペンを握る力と筆圧の境界を示す。曲線の上側領域にあたる力を加えることは困難なため、面積の半分以上が曲線の上側にある三つの矩形はターゲット候補から外した。そして、左上から順番に 1–22 の番号を割り振った。

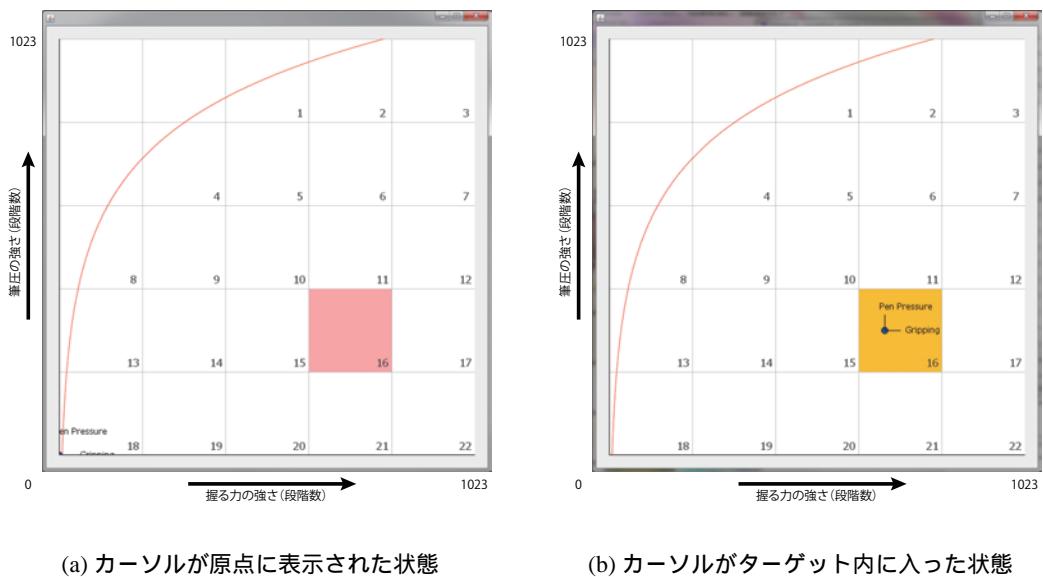


図 6.12: 被験者に示された Pressure Map

タスクは 2 種類あり、それらの違いは試行開始時のカーソルの位置である。タスク 1 では、カーソルが Pressure Map の原点にある状態から試行を始め、タスク 2 ではカーソルが右上、すなわち握る力と筆圧が最大の状態から試行を始める。gripping と筆圧を両方同時に利用するアプリケーションを実際に利用するシーンでは、入力開始時の二つの力の大きさは一定で

はない。そこで、入力開始時の力の大きさに依存して快適に入力ができる領域が変化するか否かを調べるために2種類のタスクを用意した。

実験を開始し、まず被験者がカーソルを開始位置に合わせると、ターゲットが提示される。被験者は握る力と筆圧を調節してターゲットにカーソルを合わせる。カーソルがターゲット内に入ると、図6.12(b)のようにターゲットの色が橙色に変化するため、被験者は視覚的フィードバックを基に力を制御できる。カーソル内で*Keeping*を行うことで、そのターゲットを選択することができる。これをターゲットの数、すなわち22回行う。ターゲットの提示順はLatin squareを用いてバランスを取った。各被験者はこれを3回繰り返した。まとめると、この実験では、被験者9人×22のターゲット×3試行×2タスクの合計1188回のターゲット選択が行われた。

#### 6.6.4 パフォーマンスの測定

ターゲット選択時間(ST)とカーソルの移動距離(CM)の二つの観点から評価を行った。STは素早さ、CMは容易さを示す指標として利用する。開始時のカーソル位置とターゲット間の直線距離はターゲットごとに異なるため、STとCMをそのまま利用できない。そこで、開始時のカーソル位置とターゲット間の距離Dをターゲットごとに定め、STとCMそれぞれをDで割ることで距離の影響を解消した。タスク1の場合、距離には原点からターゲット矩形の左下の座標までのユークリッド距離を採用した。タスク2の場合、距離には右上の座標(800, 800)からターゲット矩形の右上の座標までのユークリッド距離を採用した。そして、一つのターゲット矩形の辺の長さを1としたときの各ターゲットのユークリッド距離に1を加えたものを距離Dとする。これは原点を含むターゲットの距離を1とするためである。たとえば、タスク1でのターゲット14の距離は $D = 1 + \sqrt{2}$ 、ターゲット20の距離は $D = 3$ である。

#### 6.6.5 結果

$\pm 2\sigma$ を超える61の計測値を外れ値としてデータセットから取り除いた。

##### ターゲット選択時間の分析

ターゲット選択時間(ST)の結果を図6.13に示す。まず、タスク1とタスク2の22のデータセットについて相関を調べた。その結果、ターゲット選択時間には相関は見られなかった( $r = -0.008$ )。また、図6.13(a)より、タスク1では1, 4, 8の三つのターゲット選択に多くの時間を費やしていることがわかった。さらに、図6.13(b)より、タスク2では1と2の二つのターゲット選択に多くの時間を費やしていることがわかった。

選択に時間がかかった1, 2, 4, 8の四つのデータセットとそれ以外の18のデータセットにわけ、それについて相関を調べた。その結果、1, 2, 4, 8のデータセットには強い負の相関が見られた( $r = -0.996$ )。もう一方には、やや弱い負の相関が見られた( $r = -0.623$ )。

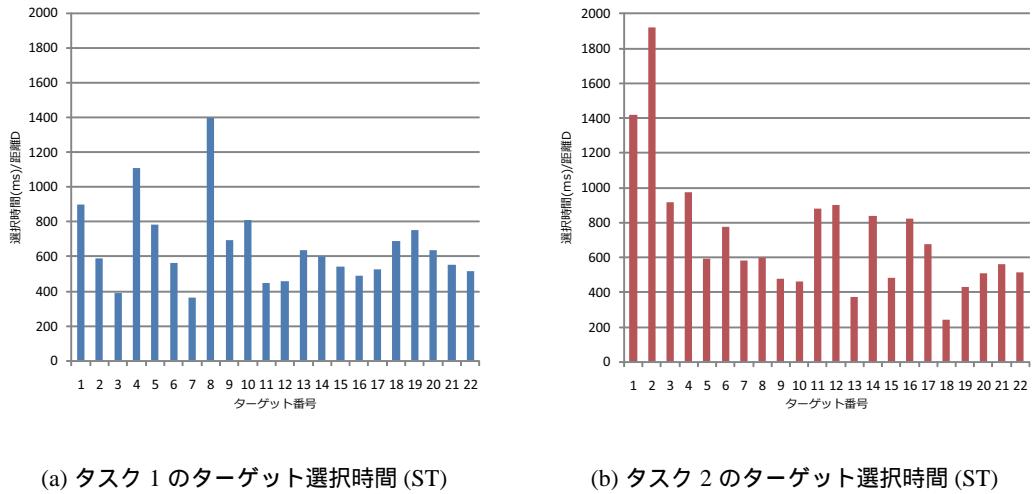


図 6.13: ターゲット選択時間 (ST) の結果

#### カーソル移動距離の分析

カーソル移動距離 (CM) の結果を図 6.14 に示す。まず、タスク 1 とタスク 2 の相関を調べた。その結果、カーソル移動距離には相関は見られなかった ( $r = 0.585$ )。図 6.14(a) より、タスク 1 では 1 と 8 のターゲット選択にカーソル移動が多いことがわかった。また、図 6.14(b) より、タスク 2 では 1, 2, 4 のターゲット選択にカーソル移動が多いことがわかった。

さらに、選択に時間がかかった 1, 2, 4, 8 の四つのデータセットとそれ以外の 18 のデータセットにわけ、それぞれについて相関を調べた。その結果、1, 2, 4, 8 のデータセットには負の弱い相関が見られた ( $r = -0.625$ )。もう一方には、相関は見られなかった ( $r = 0.323$ )。

#### 6.6.6 考察

ターゲット選択時間 (ST) とカーソル移動距離 (CM) の結果を解析した結果、両方において 1, 2, 4, 8 の四つのターゲット選択が難しいことがわかった。四つのターゲットの位置を図 6.15 の Pressure Map に示す。図 6.15 を見ると、選択が困難なターゲットは全て実験 2 で求めた曲線に沿ったものであることがわかる。つまり、この矩形領域に相当する力を加える操作は時間がかかり、簡単には行えないことがいえる。また、四つのターゲットに関してはタスク 1 とタスク 2 の間に負の相関が見られた。これは、入力開始時の力の大きさにより、選択が困難な場所が異なることを示す。しかしながら、どちらの場合にもその場所は曲線に沿っているため、やはりこれらの領域は選択が難しいといえる。よって、これらの領域をインタラクションに利用することは望ましくない。一方、それ以外の領域にはタスク 1 とタスク 2 の間に相関は見られなかったが、ある程度素早く容易にターゲットを選択できている。よって、これらの領域はインタラクションに利用可能であると思われる。

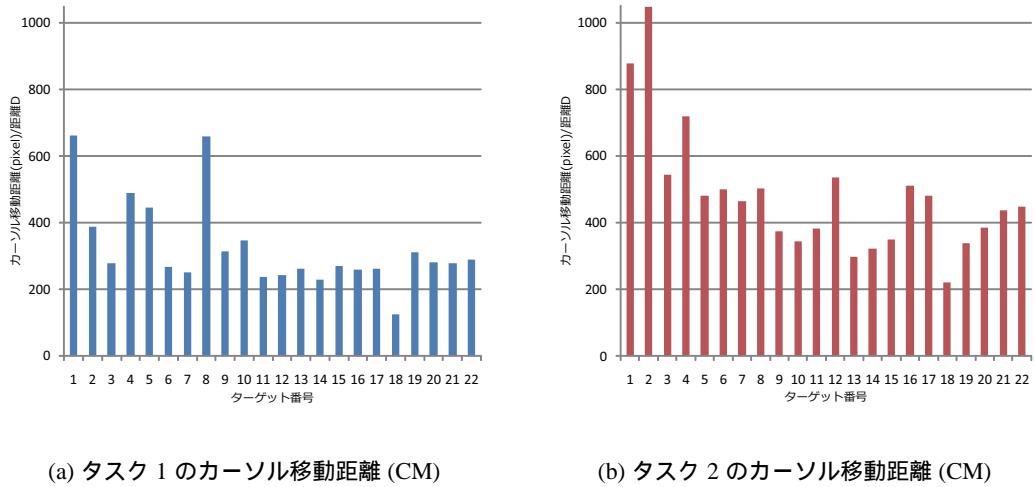


図 6.14: カーソル移動距離 (CM) の結果

実験の結果、25の領域に分割された力空間のうち、物理的に入力が困難な3領域、および選択が困難であると判明した4領域を除いた18の領域では、握る力と筆圧を組み合わせた操作が容易に行えることがわかった。つまり、握る力と筆圧の二つの力を自由に制御できる力空間は十分に広いことがいえる。よって、grippingと筆圧を組み合わせて二つの連続値を同時に入力するインターフェースの実現が可能であることが示された。

grippingと筆圧を組み合わせた操作が可能であることを示したが、握る力の大きさによって加えることができる筆圧の範囲が異なるため、筆圧をある範囲の連続値の入力として利用するためには補正が必要である。そのため、実験結果に基づいて、握る力が  $x$  のときに入力できるであろう最大の筆圧を  $y = 0.667x + 213$  という式で表現することにした。これはターゲット2と8の右下の座標を線形に結ぶ直線（図6.15中の緑の直線）の式である。実際には、この式が示す  $y$  よりも大きい筆圧を加えることも可能な場合もあるが、個人差を考慮し、かつアプリケーションにおける実装を簡単にするためにこの式を採用した。6.8.3節で紹介するgrippingと筆圧を組み合わせたを利用したアプリケーションでは、この式を利用した設計を行っている。

## 6.7 議論

実験2、3の結果、人間は握る力と筆圧を同時に使うことが可能であることがわかった。しかしながら、これらの実験では被験者全員が20代であったため、これらの結果が高齢者や子供を含めたあらゆる人間に対して適用できるとは限らない。高齢者や子供を対象とした実験はgrippingに関する今後の課題の一つであるが、少なくとも20代の男女で良好な結果を得られたことには意義がある。

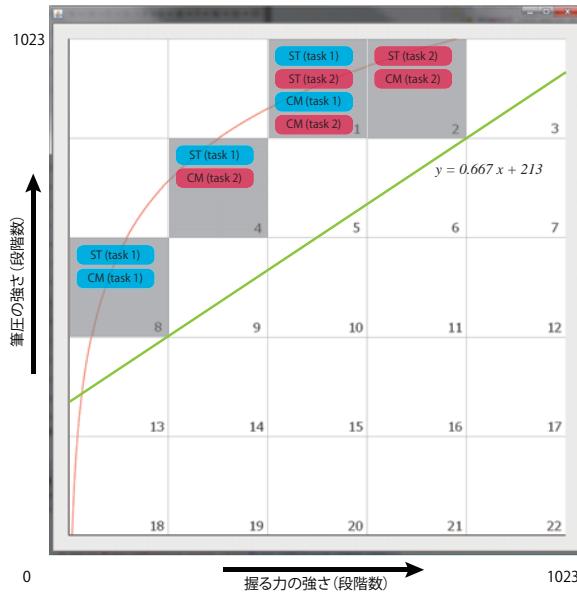


図 6.15: 選択が困難な四つのターゲットを示した Pressure Map

また、今回の実験では gripping を長時間使用することによる順応の影響について検証を行っていない。gripping だけでなく、力を加えることによるインタラクションを考える場合、順応を考慮に入れた設計は必要不可欠であるため、これは今後検討すべき課題である。

## 6.8 gripping の応用

三つの実験結果を基に、いくつかのアプリケーションを設計した。離散値入力を用いたアプリケーションおよび連続値入力を用いたアプリケーション、gripping と筆圧を組み合わせたアプリケーションにわけて紹介する。

### 6.8.1 離散値入力の応用

6.4 節で述べた実験 1 の結果、gripping は最大で 6 段階の離散値入力として利用できることがわかった。gripping による離散値入力はペン先がディスプレイから離れた状態で操作できるため、タップやストローク操作とは独立した入力操作を提供できる。これは筆圧やバ렐ボタンでは実現できない、gripping でのみ実現可能な操作である。

#### gripping ランチャ

gripping ランチャは六つの離散値に対して、アプリケーション固有の機能を割り当て、キーボードのショートカットキーのような操作を実現するアプリケーションである。ペン型デバ

イスでメニューバーや小さなメニューアイコン等のメニューインターフェースを操作することは容易ではないが、gripping ランチャを利用することで、利用頻度の高いアプリケーションの起動や、アプリケーション内で利用頻度の高い機能の呼び出し等を容易に行える。

また、小画面デバイスの場合、画面領域は貴重であり、メニューアイコンの表示等に領域を浪費することは望ましくない。gripping によりランチャを表示するようなインターフェースを設計することで、表示領域を必要としないメニューインターフェースを構築できる。

具体的な実装例として、ペイントツールでの機能呼び出しのショートカットとして実装した gripping ランチャを図 6.16 に示す。メニュー項目は六つあり、縦方向に以下の順番で並んでいる。

- ランチャ表示 / 非表示
- カラーパレット
- 画像貼り付け
- 消しゴム
- 鉛筆
- ペン

ペン先がディスプレイに接していない状態でペンを強く握り、離散値の最大値を入力すると、gripping ランチャが表示される。ランチャが表示されると、入力した離散値に応じて選択項目がハイライトされる。図 6.16 では下から 3 番目の項目がハイライトされている。項目をハイライトした状態を一定時間維持することで、項目の選択を確定することができる。

### gripping ソフトウェアキーボード

ペン入力インターフェースにとって文字入力は大きな課題の一つである。物理的なキーボードが存在しない環境では、文字入力のためにソフトウェアキーボードが利用される。ソフトウェアキーボードでは、キーボードのかな / 英字、大文字 / 小文字等のモード切り替えを行うために、物理的なキーボードと同様にシフトキーや Caps Lock キー、半角全角キー等を操作する必要がある。gripping の離散値入力にそれらを割り当てることで、モード切り替えのためのボタン操作を行うことなく、かなと英字が混在するような文章を入力することが可能となる。

図 6.17 に実装した gripping ソフトウェアキーボードを示す。図 6.17(a) は小文字アルファベットのキーセット、図 6.17(b) はかな文字のキーセットである。このアプリケーションでは離散入力を用いて 4 つのキーセットの切り替えを行う。キーボードの右側にあるパネルには四つのキーセットとカーソルが表示されている。図 6.17(b) は切り替えるための gripping を行っている途中のスクリーンショットである。切り替えたいキーセットにカーソルを合わせ



図 6.16: ペイントツールに実装した gripping ランチャ

て *Keeping* を行うことでキーセットの切り替えが行える。gripping の熟練者はキーセットの表示を見ずにキーセットを素早く切り替えることができる。

また、ストロークを用いた文字入力手法 [60, 113] も提案されているが、それらにも切り替え操作が伴う。そのような切り替え操作に対しても gripping は応用可能である。

### 6.8.2 連続値入力の応用

6.2.2 節で述べたように、gripping では連続値入力を行うことができる。以下に、二つのアプリケーションを示す。一つ目のアプリケーションは連続値入力を意識的に行うもの、二つ目は無意識的な連続値入力を利用したものである。

#### ペイントツール

6.8.1 節で紹介した gripping ランチャを実装したペイントツールを開発した。gripping ランチャにある六つのメニュー項目のうち、以下の四つの機能が連続値入力に対応する。

- カラーパレット

カラーパレットは HSV 色空間で表現されており、図 6.18(a) に示すような円形のパレットである。色相が角度、彩度が中心からの距離で表現されている。明度は gripping で入力する連続値に割り当てられており、強く握ることで図 6.18(b) のようにカラーパレットの明度が低くなる。これにより、HSV 色空間で表現できるあらゆる色を選択できる。



(a) 小文字アルファベットのキーセット



(b) かな文字のキーセット

図 6.17: gripping ソフトウェアキー ボード

- 画像貼り付け

画像を貼り付ける機能である。gripping で入力する連続値により、貼り付ける画像のサイズを変更できる。強く握ると大きな画像になり、弱く握ると小さな画像になる。

- 消しゴム

描かれた文字や絵を消去する機能である。gripping で入力する連続値により、消しゴムのサイズを変更できる。強く握るとサイズが大きくなり、弱く握ると小さくなる。

- ペン

筆ペンのような感覚で描ける機能である。gripping で入力する連続値により、筆先のサイズを変更できる。強く握ると筆先が太くなり、弱く握ると細くなる。

### 行動推定

行動推定は無意識的なペンを握る動作を連続値入力として捉えたアプリケーションである。システムは常にペングリップにかかる圧力を 1024 段階で検出している。よって、ユーザが意識的に加える力だけでなく、無意識的に加えている握る力も検出できる。このような無意識的な力も含めてペンを握る力の時間変化を解析することで、ユーザの行動を推定可能ではないかと考えた。

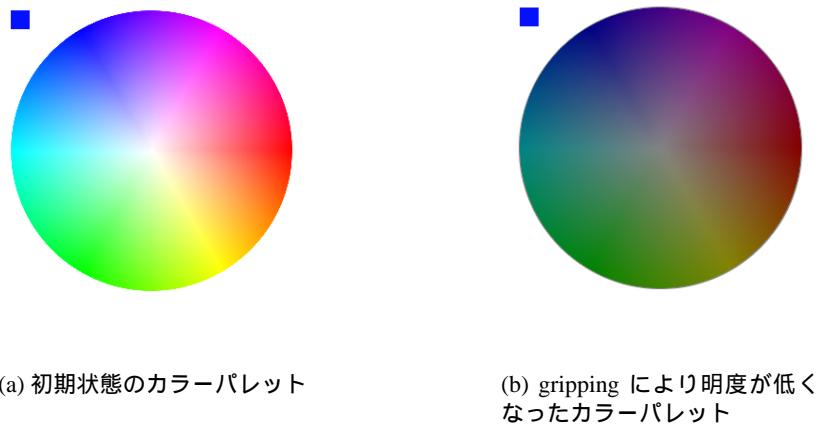


図 6.18: gripping で明度を調節可能なカラーパレット

握る力の変化に関する簡単な実験を行った結果、描画作業を始める前にペンを握っている状態、描画を始めるためにペンをディスプレイに近づけた状態、描画作業を行っている状態の三つの状態が推定可能であることがわかった。

これをキャンバスの表示内容に切り替えに応用した。ペンを休めている状態、つまりペンを握っているだけの状態では図 6.19(a) に示すように自動的にプレビュー表示になり、描画を始めた状態、つまり描画を始めるためにペンをディスプレイに近づけた状態では図 6.19(b) のように自動的に描画用のグリッドが表示される。

このように、ユーザは握る力を意識することなく操作を行っているだけであるが、システムがユーザの状態を自動的に検出し、状態に適したサポートを提供することが可能になる。

### 6.8.3 gripping と筆圧を組み合わせたインタラクション手法の応用

実験 2, 3 の結果、gripping と筆圧を同時に利用した入力操作が可能であることがわかった。つまり、操作を中断せずに二つのパラメータを同時に変更しながら作業を行うことができる。このような操作が行えるようになることも gripping の大きな特徴である。ここでは、gripping と筆圧を組み合わせたアプリケーションを二つ示す。

#### ペイントツール

いくつかのペイントツールでは筆圧をサポートしており、ストローク操作を行いながら筆圧を加えることで線幅を変更できる。本ツールも同様に筆圧で線幅を変更でき、筆圧を強くするにつれて線幅が太くなる。さらに、gripping により線の彩度を変更できる。強くペンを握ると彩度が低くなり、弱く握ると彩度が高くなる。gripping と筆圧を用いると、図 6.20 のよ

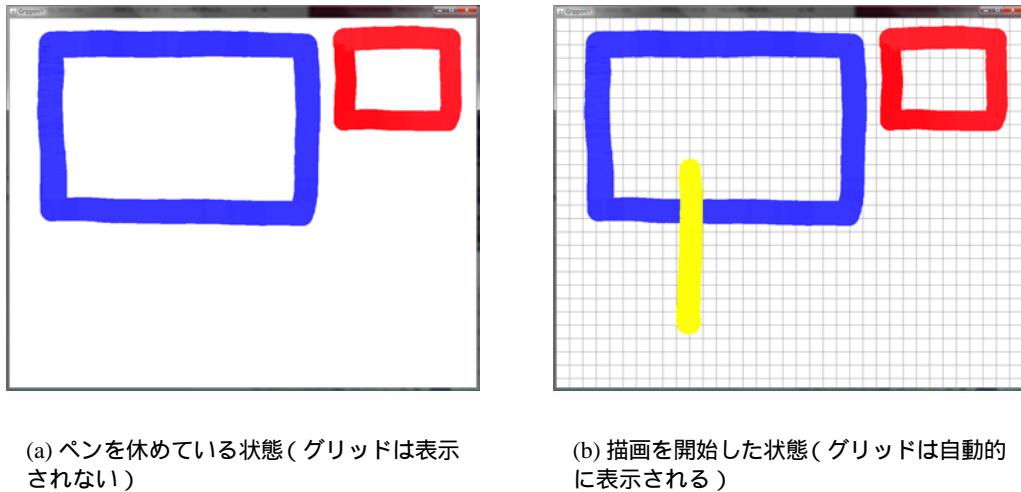


図 6.19: 行動推定アプリケーション

うな線が描ける。図 6.20 の上の線は筆圧のみ、真ん中の線は握る強さのみ、下の線は両方を変更しながら描いたものである。それぞれの割り当ては変更することができ、グラデーションの制御や透明度の制御にも割り当てることができる。このようなストローク操作を行いながら二つのパラメータをペンのみで同時に変更する操作は、gripping を開発したことにより実現可能になった操作である [123]。

本ツールの試用に際して、「両方の力を同時に制御することは難しいと予想していたが、予想に反して簡単であった」等のコメントが得られた。アプリケーションレベルでの実用性に関する厳密な評価は今後行う必要があるが、この試用から gripping と筆圧の組み合わせ操作のアプリケーションレベルでの高い実用性が期待できることがわかった。

### 地図ビューア

地図操作はスクロールや、ズーム、書き込み等、さまざまな作業を伴う操作の一つである。そこで、gripping の離散値入力と連続値入力、および筆圧を利用してペン型デバイスのみで地図操作を快適に行える地図ビューアを開発した。本ツールを操作する様子を図 6.21 に示す。

入力操作と呼び出される機能は表 6.2 のように対応付けを行った。ピンモードでは地図上にピンを立て、場所の記録等が行える。メモモードでは地図上に手書きのメモを残すことができる。移動モードでは地図をスクロールできる。これらの操作は離散値を入力後にディスプレイをタップ、もしくはストロークすることで行える。ペン先がディスプレイに接していない状態で連続値入力を行うと、地図のズームアウト操作が行える。また、筆圧を入力することで地図のズームイン操作が行える。

このように、gripping と筆圧を活用することで、多くのモードがあるアプリケーションでも GUI 操作を行うことなく、ペン型デバイスだけでさまざまな操作が可能になる。

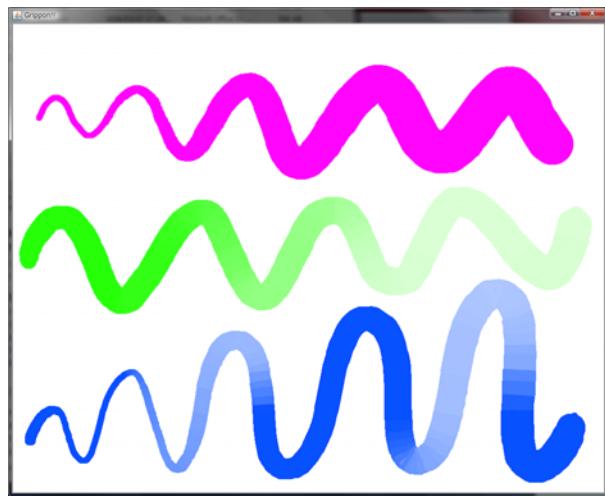


図 6.20: gripping と筆圧の組み合わせ操作が可能なペイントツール。上の線は筆圧のみ、真ん中の線は握る強さのみ、下の線は両方を変更しながら描いた。

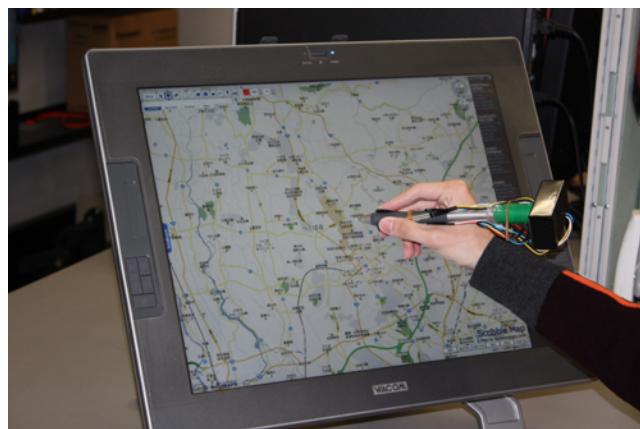


図 6.21: 地図ビューアを操作する様子

表 6.2: 地図ビューアにおける操作割り当て

インタラクション手法	割り当てた操作
離散値 (1)	ピンモード
離散値 (2)	メモモード
離散値 (3)	移動モード
連続値入力	ズームアウト
筆圧	ズームイン

# 第7章 タッチする指を識別するインタラクション手法

本章では、指入力インターフェースに併行動作を適用して創出した、タッチする指を識別するインタラクション手法について述べる。本インタラクション手法では、併行動作として指を使い分ける動作を利用する。従来の指入力インターフェースでは、タッチする指自体に意味はなく、タッチされた座標が入力情報として意味を持っていた。一方、タッチする指を使い分ける動作を指入力インターフェースに適用することで、タッチする指自体に意味を持たせることができになる。たとえば、既存の指入力インターフェースでは人差し指でのタッチと中指でのタッチはどちらも同じ1点のタッチとして扱われるが、それらを別の入力として扱えるようになる。このようなインタラクションスタイルを Finger-Specific Interaction として提唱する。本章では、Finger-Specific Interaction のコンセプトと有効性、およびプロトタイプシステムとして開発した画像処理ベースのシステム、指入力インターフェースの実際の利用シーンを想定したアプリケーションについて紹介する。

## 7.1 指入力インターフェース

2.4.1節で述べたように、タッチインターフェースはペンで操作するタイプと指で操作するタイプの二つにわけることができる。指入力インターフェースとは、指で操作するタッチインターフェースである。

指入力インターフェースは、公共の場所における端末から、テーブルトップや壁面ディスプレイ、モバイルデバイスに至るまで、さまざまなデバイスのインターフェースとして採用されている[125]。指入力インターフェースを用いたインタラクションの大きな特徴の一つとして、指を用いてディスプレイ上のオブジェクトに直接触れて操作が行える点が挙げられる。このような操作スタイルは、従来のようなマウスベースの間接操作よりもユーザに自然な操作感を与えることができる。

大画面ディスプレイを持つ指入力インターフェースの場合、画面全体を見渡すことができ、さらに全てが操作可能な領域であるため、複数人による操作が容易に行えるという特徴も持つ。

### 7.1.1 指入力インターフェースの利点

他のインターフェースと比べて直感的な操作感覚や自然なインタラクションスタイルを与えることができる点に加えて、指入力インターフェースの大きな利点の一つとして、指のみで操

作が行えることが挙げられる。指入力インターフェースの操作には特別な装置やデバイスを必要とせず、指だけで座標入力操作が行える。ディスプレイに触れるだけで操作が行えるという直感さと容易さを兼ね備えたインターフェースである。

### 7.1.2 指入力インターフェースの課題

指のみで操作が行えることは容易に座標入力が行えるという利点である一方、このことは、マウスの右クリック／左クリックのように、座標入力に別の状態を加えたような入力が行えないという制約にもなっている。この制約は、指入力インターフェースがマウス等のデバイスと比較して扱える入力情報量が少ないとこの原因の一つである。このことは、まだ表出化していない、指入力インターフェースに内在する問題点の一つであると筆者は考えている。

一般的に、指入力インターフェースはマウス等のデバイスと比較して誤入力が発生しやすいという問題も持つ。誤入力が発生する原因にはさまざまな議論が存在する。その一つが、指によるオクルージョンの発生、つまり操作する指でディスプレイが隠れてしまうことである。このオクルージョンにより、タッチしたい場所を正確にタッチすることができないため、誤入力が発生してしまう。また、操作に対する物理的フィードバックがないことも原因の一つとして挙げられる。マウス操作ではボタンが物理的に押下されるためにクリック感を得ることができると、指入力インターフェースの場合にはそれがない。

これらのオクルージョン問題やフィードバック問題は、指入力インターフェースの操作において、ディスプレイ上のオブジェクトを正確にタッチすることが要求されるために問題となっているとも考えることができる。たとえば、指がディスプレイを隠してしまうため、正確なタッチが困難になり、オクルージョン問題が発生している。また、タッチすること自体を触覚フィードバックとして人間は認知しているが、その触覚フィードバックからでは、ディスプレイ上のどこをタッチしたかをフィードバックとして得ることができない。よって、正確なタッチが困難となり、フィードバック問題が発生している。

誤入力の問題は、特にモバイルデバイスにおいて問題になりやすい。モバイルデバイスはディスプレイが小型であるため、操作用UIのために大きなディスプレイ領域を使用することができず、ボタン等のGUIが小さくなりがちである。このようなモバイル環境ではオクルージョン問題やフィードバック問題はその操作性に大きな影響を与えている。

## 7.2 目指す指入力インターフェース

指入力インターフェースの持つ直感的な操作感覚や自然なインタラクション等の良さを活かしつつ、上述した問題点を解決するインターフェースを開発する。具体的には、以下のような指入力インターフェースを目指す。

従来の指入力インターフェースの使いやすさを維持する

指入力インターフェースは、指のみで操作が行えることや、指を用いてディスプレイ上の

オブジェクトに直接触れて操作が行えることが使いやすさの要因である。よって、これらの特徴を維持するような指入力インターフェースを目指す。

#### 指入力インターフェースのみでさまざまな入力が行える

指入力インターフェースは指をタッチすること以外には入力情報がない。他のデバイスを補助的に利用することで、入力情報を増加させるというアプローチも考えられるが、従来の指入力インターフェースの使いやすさを失ってしまいかねない。よって、指でタッチする動作に内在する情報のみを利用する。

#### 要求されるタッチの正確さを軽減することができる

オクルージョンとフィードバックのなさが問題となる原因として、正確なタッチが要求されることを取り上げた。指入力インターフェースの操作に要求されるタッチの正確さを軽減できる指入力インターフェースを目指すことにより、オクルージョン問題とフィードバック問題を間接的に解決することを目指す。

### 7.3 Finger-Specific Interaction の提唱

#### 7.3.1 コンセプト

指入力インターフェース向けのインタラクション手法のコンセプトとして、Finger-Specific Interaction（以下、FSI）を提唱する。FSIは、タッチする指を識別し、指の違いを入力として扱えるようにするというコンセプトである。FSIにより、今までほとんど使われてこなかった指の違いを指入力インターフェースのインタラクションで活用可能になる。たとえば、既存の指入力インターフェースでは人差し指でのタッチと中指でのタッチはどちらも同じ1点のタッチとして扱われるが、FSIではそれらを別の入力として扱う。つまり、FSIではそれぞれの指が意味を持つようになる。既存の指入力インターフェースでは、ディスプレイ上に表示されたオブジェクトにタッチすることで初めて意味をなす入力となっていたが、FSIではタッチする行為自体が入力としての意味を持つ。FSIはマルチタッチ操作時における指の識別も可能であるため、複数の指や複数人で操作が行えるという指入力インターフェースの特徴を活用することができる。

#### 7.3.2 有効性

タッチする指の違いをインタラクションに利用可能にすることで、以下の3点が実現可能になる。

##### 入力プリミティブ数の増加

まず、入力プリミティブと入力ボキャブラリについて説明する。入力プリミティブとは、入力情報の最小単位のことを指す。たとえば、マウスの右クリックや左クリック、キーボード

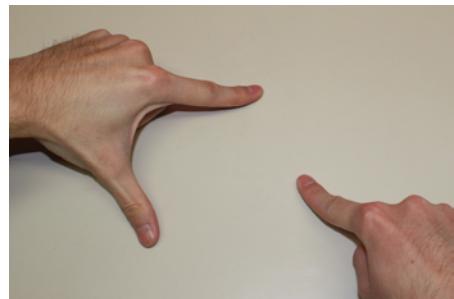
の各キーはそれぞれが入力プリミティブである。入力ボキャブラリとは、入力プリミティブの組み合わせた入力情報である。たとえば、入力操作にマウスの右クリックとキーボードのControlキーとShiftキーを用いる場合、その組み合わせは7通り存在し、それらが入力ボキャブラリである。

既存の指入力インターフェースのインタラクション手法では、タッチされた指の位置と、その時間変化が入力情報として利用されてきた。既存の指入力インターフェースの場合、指の区別が行えないため、どの指で入力してもそれらは同じ一つの入力として扱われてしまう。つまり、入力プリミティブ数は1である。一方、FSIでは指を識別するため、各指をそれぞれ異なる入力として扱うことができる。よって、操作に用いる指の数が入力プリミティブ数となる。

マルチタッチ環境では複数の指を組み合わせることもでき、 $n$ 本の指の組み合わせの数  $2^n - 1$  が入力ボキャブラリ数となる。図7.1に示すマルチタッチ操作を例として挙げる。3点のマルチタッチ操作として、図7.1(a)のような片手で3点をタッチする操作と、図7.1(b)のような片手で2点、もう一方の手で1点をタッチする操作を考える。既存の指入力インターフェースの場合、両者は同一の入力として扱われる一方で、FSIでは異なる入力として扱われる。



(a) 片手での3点タッチ操作



(b) 両手での3点タッチ操作

図7.1: 3点タッチ操作のイメージ。Finger-Specific Interactionではこれら二つの操作が異なる入力として扱われる。

タッチする指の本数によりその振る舞いを変えることができる指入力インターフェース[3]やApple Magic TrackPadのようなデバイスも存在する。これらは個々の指では一つの振る舞いしかできないため、操作に2本以上の指を用いる場合でも入力プリミティブ数は1であると考えることができる。操作に用いる指の数が $n$ 本の場合、入力ボキャブラリ数は $n$ である。

FSIはスマートフォンやPDA等の小型ディスプレイを持つモバイルデバイスと特に相性が良い。モバイルデバイスにおいて入力ボキャブラリ数を増やすために、しばしばマルチタッチ操作が採用されているが、小型ディスプレイでのマルチタッチ操作は必ずしも快適な操作であるとはいえない。FSIを採用することで、シングルタッチのみでマルチタッチ以上のさまざまな操作が可能になる。FSIをモバイルデバイスに適用することは入力プリミティブの観点から有用であることがわかり、ユーザビリティの向上が期待できる。

### eyes-free インタラクションの実現

従来の指入力インターフェースではディスプレイ面に表示されたオブジェクトを指で操作するため、その操作には物理的フィードバックが欠けている。そのため、操作する際には必ず操作対象を視認する必要があり、eyes-free インタラクション、つまりディスプレイから目を離して操作を行うことは困難である。FSI は人間の運動出力、すなわち指でタッチする行為自体がそのままコンピュータへの入力となる。たとえば、メニュー項目を選択する場合、既存の指入力インターフェースではまずメニューアイコンを視認し、そのアイコンを正確にタッチする必要がある。一方で、FSI では指をタッチするだけでよく、指をタッチする行為自体が実行するコマンドの選択操作となる。よって、ディスプレイ上の GUI を視認する必要のない操作、つまり eyes-free インタラクションが可能になる。このことから、FSI は人間の運動出力とコマンド選択をダイレクトに繋ぐインターフェース技術と考えることもできる。

指入力インターフェースを備える携帯音楽プレイヤー等のモバイルデバイスの操作には、必ずディスプレイを視認しなければならない。しかしながら、FSI を採用することでポケットやバッグに入れたままディスプレイを見ることなく操作が行えるようになる。このように、eyes-free インタラクションの観点からも FSI とモバイルデバイスの相性の良さがうかがえる。

### 人間の識別

操作する指を区別することによって、DiamondTouch [15] のように操作する人間を識別することも可能になる。これは、指入力インターフェースを用いて CSCW のような複数人での協調作業を行う場面では有効である。FSI では指という細かい粒度で人間を識別することができるため、他人に干渉されない操作等、人の識別だけでも実現できる操作だけでなく、ユーザの指ごとに操作に対する優先度や権限を付与すること等が可能になる。

## 7.4 Finger-Specific Interaction における検討課題

FSI には 7.3.2 節で述べたような利点がある一方で、FSI を利用するためには記憶と運動特性に関して検討すべき課題が存在する。

### 7.4.1 対応付けを記憶する困難さ

FSI では、 $n$  本の指で操作する場合、その組み合わせ数は  $2^n - 1$  になる。この組み合わせの多さは大きな利点である一方、数が膨大になるためユーザが指の組み合わせとアプリケーションごとの機能との対応付けを記憶することが困難であるという課題もある。

この解決策の一つとして、指の持つ意味の活用が考えられる。国や文化によって違いはあるが、それぞれの指にはなんらかの文化的意味が付与されていることが多い。日本では指は複数の意味を持つ。たとえば、親指から順に、父、母、兄、姉、幼児を意味し、薬指は薬を塗るときに使われる。人差し指と中指の組み合わせはハサミや勝利を意味する。また、日常生活

活シーンでは、ものを摘むときには片手の親指と人差し指を用い、モノを引き延ばすときには両手の指を使う。このような指の持つ意味を活用して対応付けを行うことが、課題を解決するための手がかりになるとを考えている。

別の解決方法として、ユーザ自身が積極的にマッピングを行う方法が考えられる。たとえば、モードやコマンドを表す絵や文字の書かれた指輪やつけ爪を用意しておき、ユーザがそれらの中から使いたいものを取捨選択して指に付ける。このような仕組みにすることで、指を見るだけで指のコマンドの対応付けが把握することが可能になる。また、自発的に対応付けを行っているという点で、あらかじめ決められた対応付けと比較して記憶しやすいと思われる。指輪やつけ爪を指に付加するという行為自体は、実際に机上でペンや消しゴムを手に取って使うこととほぼ同等の行為と見なすことができ、実世界指向のインタラクションスタイルを採用していると捉えることができる。この観点から、この方法はユーザに強い負担を強いることはないと考えられる。

#### 7.4.2 負の運動特性

指の組み合わせによっては指の運動特性として行いにくい組み合わせが存在する。たとえば、片手5本の指をタッチする動作を行う場合、薬指以外の4本の指を同時にタッチする動作や、中指と小指のみを同時にタッチする動作は多少困難である[44]。実際のアプリケーションにおいて、これらの負の運動特性をどのように扱うかということもFSIの課題である。

### 7.5 Finger-Specific Interaction の実現

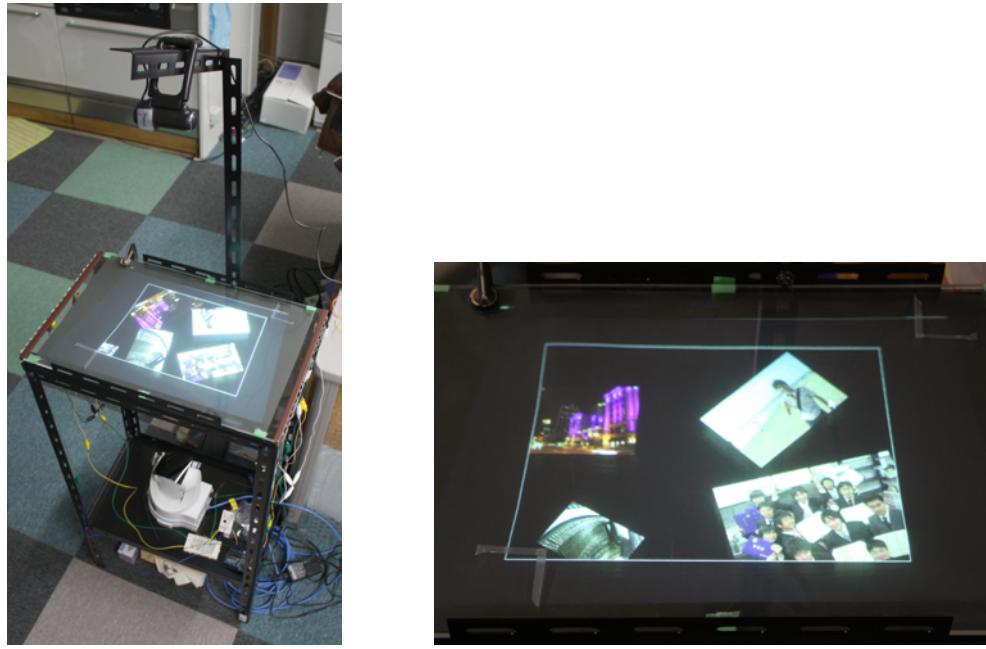
FSIを実現するためにプロトタイプシステムとして開発したテーブルトップインタフェースについて述べる。FSIを実現するためにはどの座標をどの指でタッチしたかを検出する必要がある。本システムでは、タッチの検出と指の検出を個別に行い、それらを統合するというアプローチを採用した。ここでは、本システムの概要、およびタッチ検出技術と指検出技術についてハードウェアとソフトウェアの観点からその実装について詳細を説明する。

#### 7.5.1 プロトタイプシステム

FSIを実現するためのプロトタイプシステムとして、図7.2に示す450×300 mmサイズのテーブルトップインタフェースを開発した。デバイスのフレームにはL型アングル材を使用しており、地面から900 mmの位置にディスプレイが設置してある。

このプロトタイプシステムを構成するデバイスや素材を以下に示す。

- 可視光カメラ<sup>‡</sup>
- 偏光フィルム(2枚)<sup>‡</sup>
- アクリル板<sup>†</sup>



(a) デバイス全体の外観 . スクリーンの上部に指検出用のカメラが取り付けてある .

(b) ディスプレイ部の外観 .

図 7.2: 開発したテーブルトップインタフェース

- 赤外線 LED <sup>†</sup>
- トレーシングペーパ
- 赤外線カメラ <sup>†</sup>
- プロジェクタ
- カラーマーカ <sup>‡</sup>

具体的なデバイス構成を図 7.3 に示す . まず , ディスプレイの表示はプロジェクタでリアプロジェクションすることで実現している . プロジェクタの光はアクリル板の下に貼り付けられたディヒューザで拡散され , その映像が投影される . ディヒューザにはトレーシングペーパを利用した . トレーシングペーパは非常に薄いため , 投影面と反対面 , つまりユーザから見える面にも映像が鮮明に映し出させる .

プロジェクタには EPSON 製の ELP-710 を利用した . トレーシングペーパにはコクヨ製の A3・薄口・40 g/m<sup>2</sup> を用いた .

FSI を実現するためには , どの座標をどの指でタッチしたかを検出する必要がある . このプロトタイプシステムでは , タッチ位置の検出と指の識別を個別に行い , それらのディスプレイ

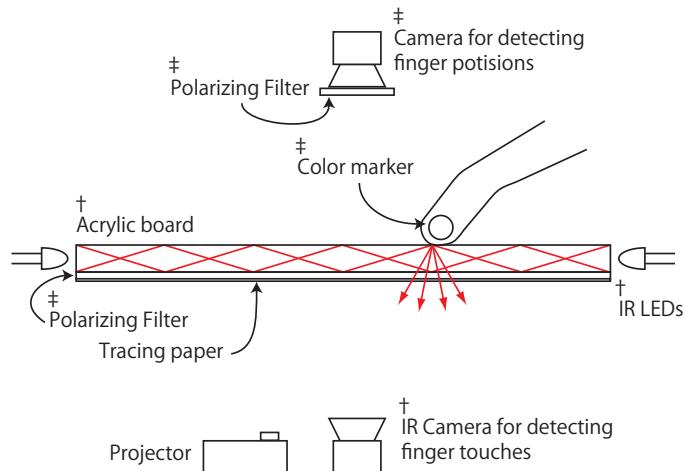


図 7.3: ハードウェア構成 . † の付いたデバイスはタッチ位置を , ‡ の付いたデバイスは指を検出するために用いる .

面における相対位置を統合することでタッチ位置と指との対応付けを行うというアプローチを採用した . デバイス構成のうち , † の付いたものはタッチ位置の検出用のデバイスや素材 , ‡ の付いたものは指検出用のデバイスや素材である .

### 7.5.2 タッチを検出する技術

#### ハードウェア実装

指入力インターフェースの研究において , タッチを検出する手法はいくつか提案されている . その中でも , 安価 , かつ容易にタッチを検出できる手法として , FTIR (Frustrated Total Internal Reflection) [23] という手法がある . FTIR は赤外光がアクリル板の中で全反射する現象を応用した技術である . 図 7.3 のように , アクリル板の横から赤外線 LED を用いて赤外光を照射する . このとき , 赤外光はアクリル板の中を全反射するため , アクリル板の中に赤外光が閉じ込められる . この状態でアクリル板に指を触ると , 指で触れた面の反射率が変わり , その部分から赤外光が漏れ出す . その漏れた赤外光を赤外線カメラで検出することにより , タッチ位置を検出することができる .

本システムでは , テーブル下部に配置した赤外線カメラにより , タッチ位置を検出する . タッチ面には  $450 \times 300 \text{ mm}$  で , 厚さ  $5 \text{ mm}$  のアクリル板を使用した . アクリル板のエッジは赤外光を照射しやすいように紙やすりとコンパウンドを用いて磨いた . アクリル板のエッジに LED を固定するために , 図 7.4 に示すプラスチックモールのコ型カプセを用いた .

赤外線カメラとして , アクシスコミュニケーションズ製の AXIS 214 PTZ ネットワークカメラを利用した . 一般的なカメラは撮影した映像に近赤外線が映り込むことを防ぐために , 赤外線カットフィルタが取り付けてある . AXIS 214 カメラは通常の可視光カメラであるが , 設定により赤外線カットフィルタを取り外すことができる . 赤外線カットフィルタを

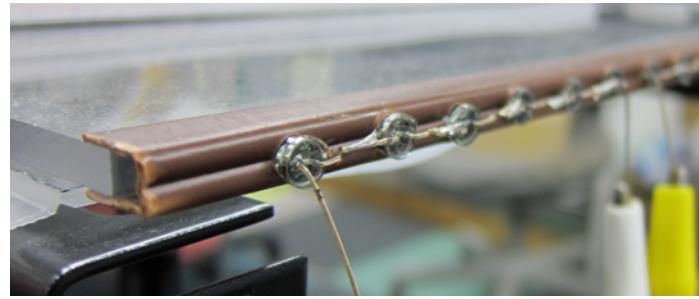


図 7.4: コ型カブセを用いてアクリル板に固定した赤外線 LED

取り外し、カメラのレンズ部に可視光カットフィルタを取り付けることで赤外線カメラとして利用した。可視光カットフィルタには富士フィルム製の IR 86 光吸収・赤外線透過フィルタを用いた。また、赤外光を照射する赤外線 LED にはピーク波長が 940 nm のものを使用した。

### ソフトウェア実装

赤外線カメラで撮影した画像を処理することで、タッチした場所の検出を行う。赤外線カメラから取得した生画像を図 7.5(a) に示す。中央付近の白く光っている二カ所が指をタッチしている場所である。まず、この画像に対して一定の閾値を用いて図 7.5(b) の二値化画像にする。赤外線カメラの映像は環境光により変化することがあるため、二値化に用いる閾値は柔軟に変更できるようにした。次に、二値化画像に対してラベリングを行う。ラベリングとは、連続した色の画素集合に対して同じラベルを付与する処理であり、画素の“塊”を見つけることができる。発見した塊のうち、サイズの小さいものは指のタッチではなくノイズであると判断し、除去する。そして、残った塊から楕円を生成し、この楕円領域を、指をタッチした領域とする。図 7.5(c) の緑の領域が生成された楕円領域である。最後に、それぞれの楕円領域の重心を算出し、その重心値を、指をタッチした座標とする。

これらの処理を行うソフトウェアは Java の標準ライブラリ、およびラベリング用ライブラリである BlobDetection<sup>1</sup>を用いて開発した。

### 7.5.3 指を検出する技術

#### ハードウェア実装

本システムでは、指の検出を行うために画像処理ベースの手法を開発した。カメラ画像を用いて指先の認識を行う。

図 7.2(a) のように、操作の邪魔にならないテーブルの上部 420 mm の位置に可視光カメラを配置した。このカメラでテーブル上を撮影し、その画像から指を検出する。高精度の指検出を行うために、指先に単色のカラーマーカーを貼り付けた。カラーマーカーを用いた指の認識

<sup>1</sup><http://www.v3ga.net/processing/BlobDetection/>

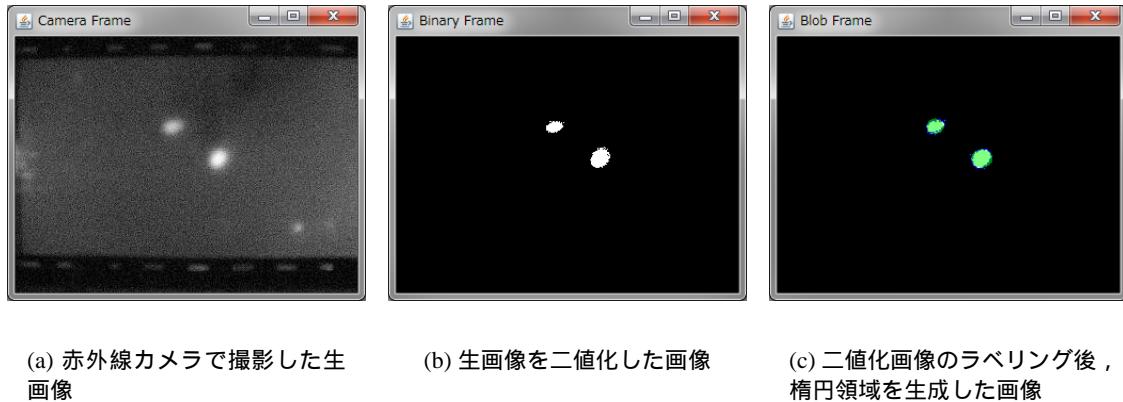


図 7.5: タッチを検出するための画像処理手順

はいくつかの研究 [52, 103] でも採用されており、高精度で指を検出できることが確認されている。本システムで使用したマーカは直径 8 mm のシールであり、非常に安価で手に入れることができる。インタラクションに用いる指ごとに異なる色のマーカを貼り付け、その色情報をテーブル上部のカメラで読み取ることで指の位置を検出する。

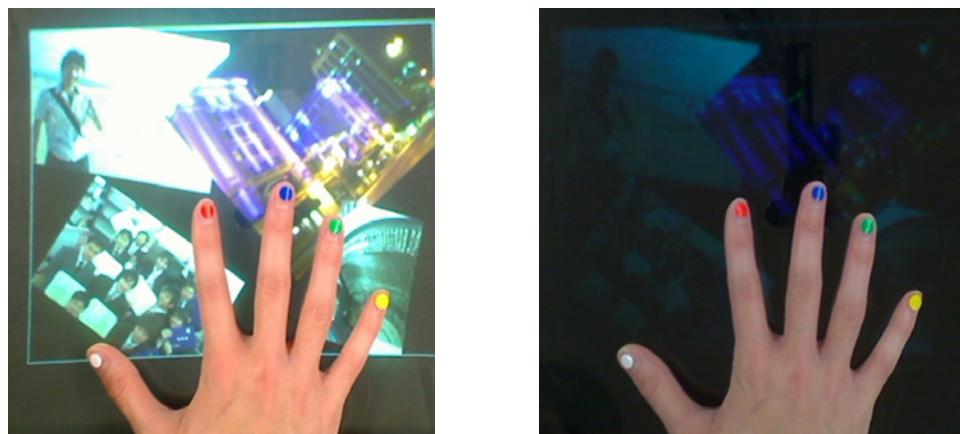
ディスプレイに何も投影していない場合はこの方法で指を頑健に検出することが可能であるが、ディスプレイに映像を投影すると、カメラから撮影される画像は図 7.6(a) のようになり、映像の色とマーカの色を誤検出してしまうことがある。この問題は文献 [39] でも利用されている偏光フィルムを 2 枚用いることで解決した。ディスプレイ面のアクリル板とトレー シングペーパの間に偏光フィルムを取り付け、その偏光フィルムと偏光軸が直交する方向に回転させたもう 1 枚の偏光フィルムをテーブル上部のカメラのレンズに貼り付ける。これにより、投影された映像がカットされた、図 7.6(b) のような手のみが映る画像を取得可能になる。

可視光カメラには、ロジクール製の Webcam Pro 9000 を用いた。偏光フィルムには、ケニス製のワイド版偏光フィルムを利用した。

### ソフトウェア実装

ディスプレイ上部に設置したカメラで撮影した画像を処理することで、指の検出を行う。カメラから取得した生画像を図 7.7(a) に示す。人差し指に赤色のカラーマーカ、中指に黄色のカラーマーカを貼り付けている。生画像からカラーマーカと類似する色を持つ画素のみを抽出する。色を抽出した画像を図 7.7(b) に示す。カラーマーカの色はあらかじめシステムに与えられている。色は HSV 色空間で扱っており、あらかじめ与えられたカラーマーカの色情報と各画素の色が、色相、彩度、輝度の全てにおいて一定の範囲内にある場合、同一の色として扱う。そして、類似する色を持つ画素の重心を算出し、その重心値を指の位置とする。

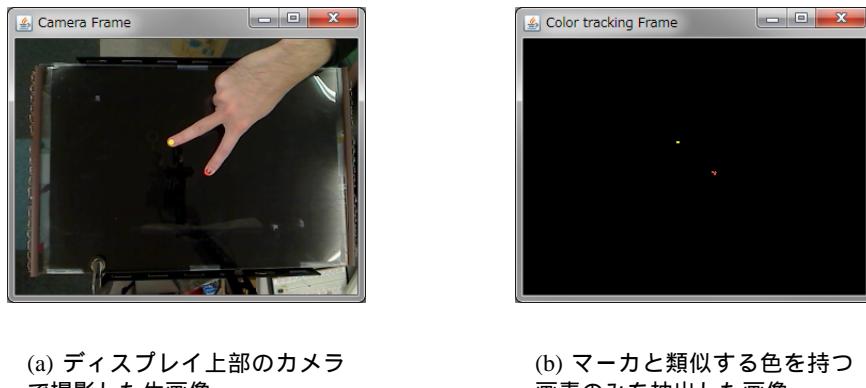
これらの処理を行うソフトウェアは Java の標準ライブラリ、および映像や音声等のメディアを扱うためのフレームワークである Java Media Framework (JMF) を用いて開発した。



(a) 偏光フィルムなしの画像。プロジェクタの映像と色マーカの区別がつきにくい。

(b) 偏光フィルム設置後の画像。偏光フィルムによりプロジェクタの映像が遮断され、色マーカのみを撮影できる。

図 7.6: テーブル上部のカメラで撮影した映像



(a) ディスプレイ上部のカメラで撮影した生画像

(b) マーカと類似する色を持つ画素のみを抽出した画像

図 7.7: 指を検出するための画像処理手順

## 7.6 アプリケーション

FSI のアプリケーション例をいくつか紹介する。これらは指入力インターフェースを実際の利用する際のさまざまなシナリオを想定している。また、7.4 節で述べた、指入力インターフェースの抱える問題点の解決に寄与するものや、指の持つ負の運動特性をうまく活用するものも開発した。

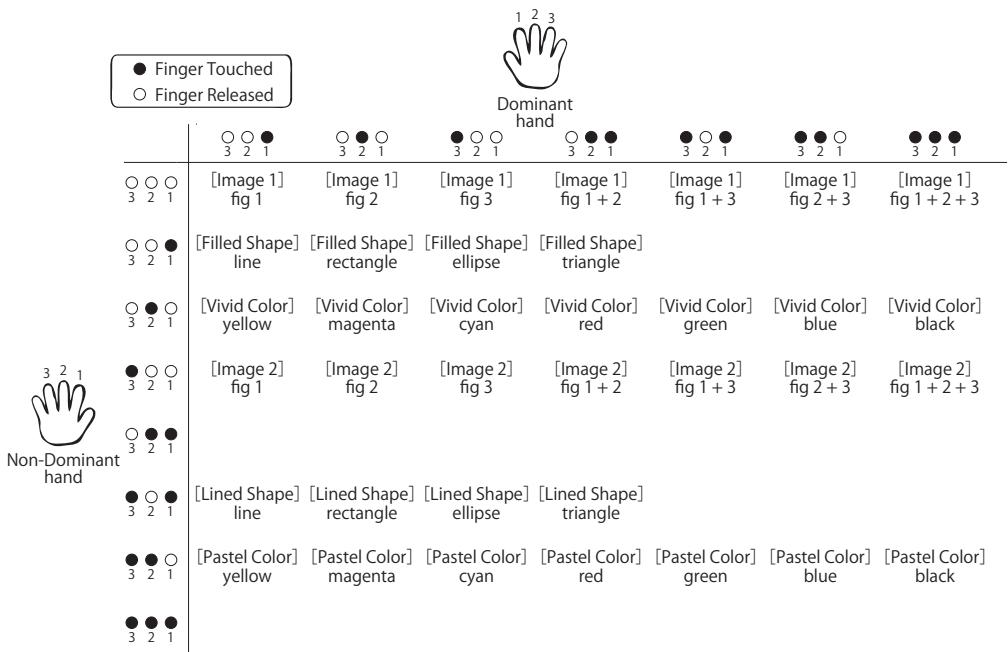
### 7.6.1 大画面向けペイントツール

このペイントツールは大画面指入力インターフェースでの利用を想定している。一般的に、ペイントツールにはペン、図形描画、画像貼り付け等の多くのモードがある。さらに、描画する図形の大きさや色等の多くのパラメータを入力する必要がある。よって、従来のペイントツールではモード切り替えやパラメータ入力のためのGUI操作が必須であった。本ツールではFSIを適用することにより、GUI操作を必要としない操作体系を実現した。

本ツールの操作には、両手の3本の指（人差し指、中指、薬指）を利用する。合計6本の指の組み合わせは63通りあり、これらにペイントツールの機能を割り当てる。操作に両手を用いるようなタスクでは、利き手では細かい動作や高い精度が要求される操作を行い、非利き手では粗い動作や高い精度が要求されない操作を行うというAsymmetric-dependent Taskが提案されており[21]、それが良いパフォーマンスを示すことも実証されている[27]。本ツールでもこれを応用し、利き手の指に操作の主となる操作を、非利き手には補助的な操作としてモード切り替えを割り当てる。

#### 操作の割り当て

具体的な操作の割り当てを図7.8に示す。図7.8の黒丸は該当する指をタッチした状態、白丸はタッチしていない状態を表す。操作スタイルとして、「非利き手によるモード選択」+「利き手による操作の実行」という両手を組み合わせた操作を基本とする。以下に、非利き手の観点から操作体系について説明する。



● Finger Touched  
○ Finger Released

Dominant hand

	○ ○ ○ 3 2 1	○ ○ ● 3 2 1	○ ● ○ 3 2 1	● ○ ○ 3 2 1	● ○ ● 3 2 1	● ○ ○ 3 2 1	● ● ○ 3 2 1	
○ ○ ○ 3 2 1	[Image 1] fig 1	[Image 1] fig 2	[Image 1] fig 3	[Image 1] fig 1+2	[Image 1] fig 1+3	[Image 1] fig 2+3	[Image 1] fig 1+2+3	
○ ○ ● 3 2 1	[Filled Shape] line	[Filled Shape] rectangle	[Filled Shape] ellipse	[Filled Shape] triangle				
○ ● ○ 3 2 1	[Vivid Color] yellow	[Vivid Color] magenta	[Vivid Color] cyan	[Vivid Color] red	[Vivid Color] green	[Vivid Color] blue	[Vivid Color] black	
Non-Dominant hand	○ ○ ○ 3 2 1	[Image 2] fig 1	[Image 2] fig 2	[Image 2] fig 3	[Image 2] fig 1+2	[Image 2] fig 1+3	[Image 2] fig 2+3	[Image 2] fig 1+2+3
○ ○ ● 3 2 1	[Lined Shape] line	[Lined Shape] rectangle	[Lined Shape] ellipse	[Lined Shape] triangle				
○ ● ○ 3 2 1	[Pastel Color] yellow	[Pastel Color] magenta	[Pastel Color] cyan	[Pastel Color] red	[Pastel Color] green	[Pastel Color] blue	[Pastel Color] black	
● ● ○ 3 2 1								

図7.8: ペイントツールにおける指の組み合わせと機能との対応付け一覧

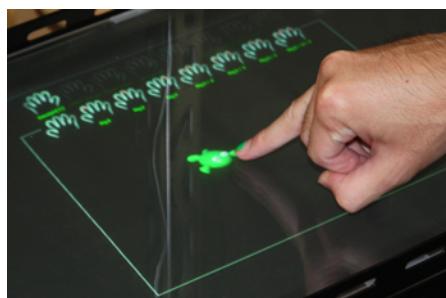
- 非利き手を使わない(0本の指を使うと考える), 利き手の指のみでの操作  
非利き手を使わない状態は0本の指を使うと考え, 画像貼り付けモードを割り当てた. 各指にはあらかじめ画像が割り当てられており, タッチする指に応じて図7.9のよう画像を貼り付ける操作となる. 複数の指をタッチしてもタッチ位置と指の対応は崩れず検出できるため, 図7.9(d)のように指と画像の対応関係を保持したまま画像を貼り付けることができる.



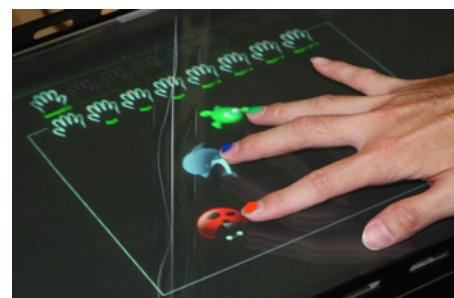
(a) 利き手の人差し指をタッチする操作



(b) 利き手の中指をタッチする操作



(c) 利き手の薬指をタッチする操作



(d) 利き手の人差し指, 中指, 薬指を同時にタッチする動作

図7.9: 利き手のみでの操作する様子

- 非利き手の人差し指を用いた操作

非利き手の人差し指には, 図形描画(塗りつぶし)モード割り当てた. 操作時の様子を図7.10に示す. 非利き手の人差し指をタッチしながら利き手の人差し指をタッチすると, それらの指の間に直線を描画でき, 利き手を中指にすると塗りつぶし矩形, 利き手を薬指にすると塗りつぶし橢円を描画できる. また, 非利き手の人差し指をタッチしながら利き手の人差し指と中指をタッチすると塗りつぶし三角形を描画できる.

描画する図形のサイズも描画時に同時に指定できるため, GUIによるパラメータ指定する操作なしに図形を描画できる. つまり, モードとパラメータの同時入力が可能である.

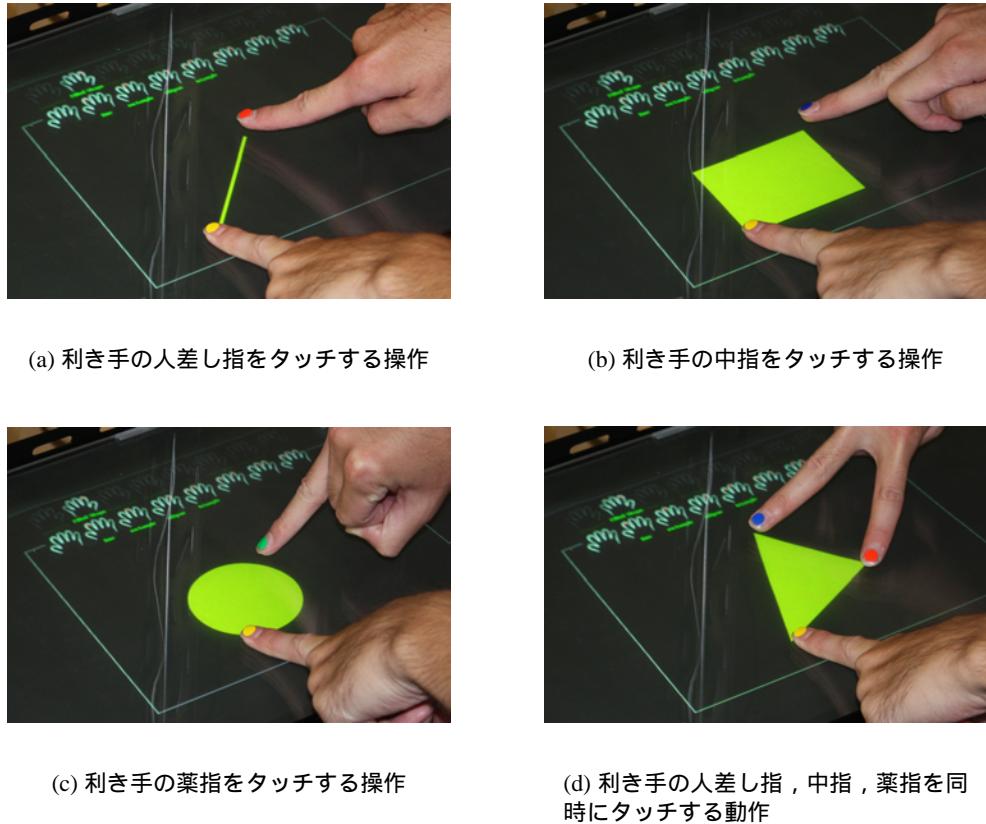


図 7.10: 非利き手の人差し指を用いて操作する様子

- 非利き手の中指を用いた操作

非利き手の中指には、色（ビビッドカラー）選択モードを割り当てた。各指にはあらかじめビビッドカラーの色が割り当てられており、指をタッチすることでその色を選択することができる。デフォルトの色の割り当ては図 7.11 に示すように、減法混色をベースにした割り当てとした。人差し指には黄、中指にはマゼンダ、薬指にはシアンが割り当ててある。人差し指と中指の組み合わせには黄色とマゼンダの混色、つまり赤が割り当てられている。同様に、人差し指と薬指には緑、中指と薬指には青、全ての指の組み合わせには黒が割り当てられている。

- 非利き手の薬指を用いた操作

非利き手の薬指には、キーボードのシフトキーのような振る舞いをするモード、つまり同時にタッチした指の補助や拡張を行うモードを割り当てた。非利き手の別の指と組み合わせることで意味をなす操作となる。

非利き手の薬指のみをタッチした場合は 0 本の指との組み合わせと考える。よって、利き手の指のみでの操作の拡張となる。具体的には図 7.12 のように、利き手の指のみで

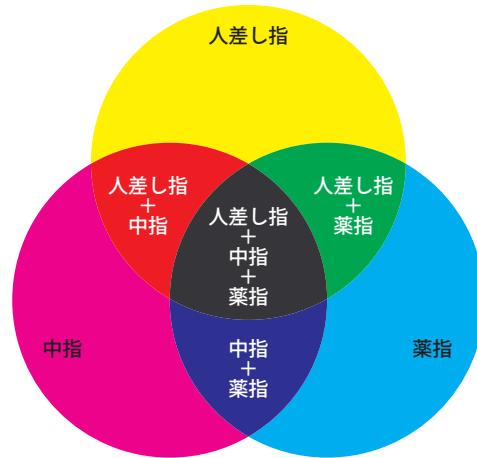
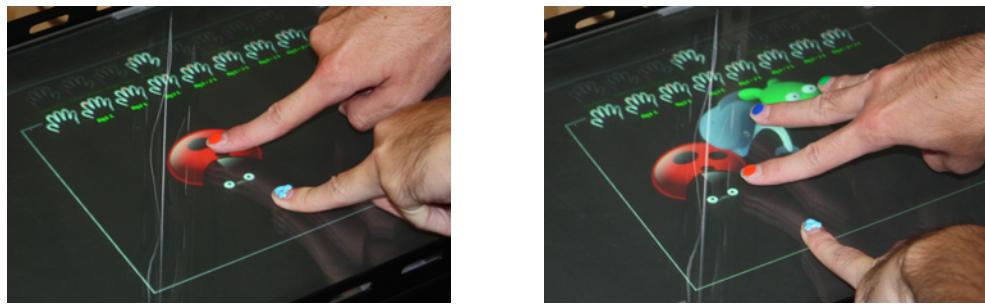


図 7.11: 非利き手の中指をタッチしたときに利き手に割り当てられている色



(a) 利き手の人差し指をタッチする操作

(b) 利き手の人差し指，中指，薬指を同時にタッチする動作

図 7.12: 非利き手の薬指を用いて操作する様子

操作した場合に貼り付けられる画像のサイズを大きくすることができる。指と画像との対応関係は変化せず、画像のサイズのみが変化する。

- 非利き手の人差し指と薬指を用いた操作  
非利き手の人差し指を用いた操作の補助機能として、図形描画（輪郭）モード割り当てた。指と機能との対応関係は非利き手の人差し指を用いた操作と同じであるが、描画される図形は塗りつぶしではなく、図 7.13 のように輪郭のみが描画される。
- 非利き手の中指と薬指を用いた操作  
非利き手の中指を用いた操作の補助機能として、色（パステルカラー）選択モードを割り当てた。指と色との基本的な対応関係は非利き手の中指を用いた操作と同じであるが、図 7.14 のようにそれぞれパステルカラーが割り当てられている。

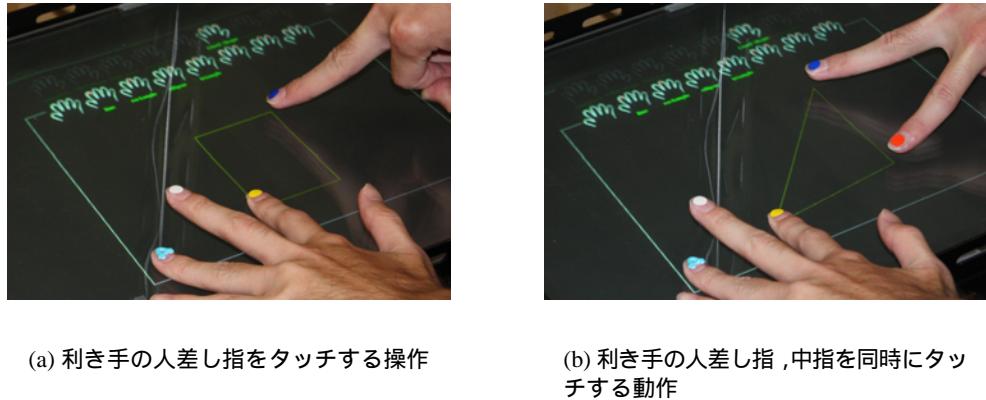


図 7.13: 非利き手の人差し指と薬指を用いて操作する様子

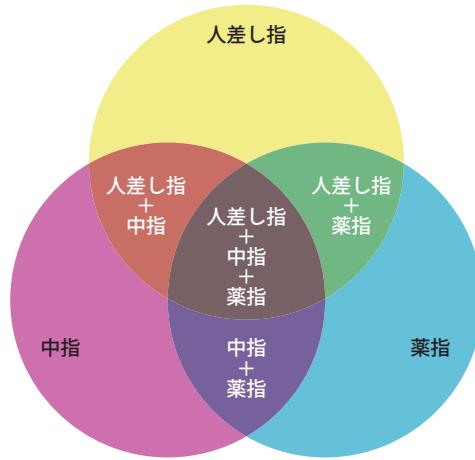


図 7.14: 非利き手の中指と薬指をタッチしたときに利き手に割り当てられている色

### 操作割り当ての表示機能

上述したように、本ツールでは操作に両手で合計 6 本の指を操作に使用し、その組み合わせは 63 通りある。本ツールでは 63 の組み合わせ全てに対して機能を割り当てていないが、指と機能の対応関係をすぐに記憶することは容易ではない。この課題を解決するために、ディスプレイ上部に対応関係を提示するパネルを表示した。パネルの外観を図 7.15 に示す。このパネルは操作中に常時表示されており、タッチしている指に応じて表示内容が動的に変化する。対応関係が記憶できていない場合、ユーザはこのパネルを見ながら操作することで、指と機能の対応を確認することができる。

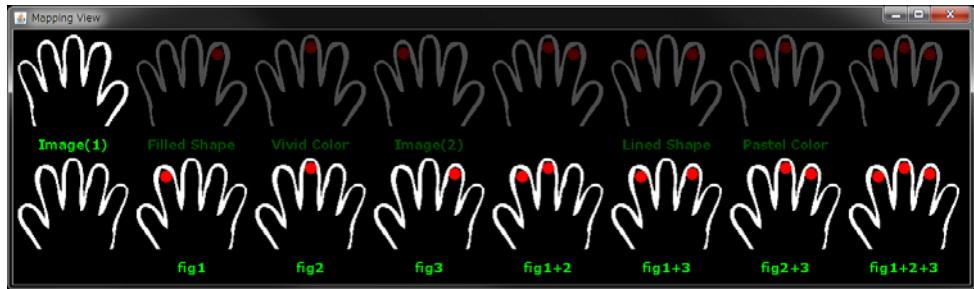


図 7.15: 指と機能との対応関係を提示するパネル

### 7.6.2 ソフトウェアキーボード

キーボードを用いて文字を入力する場合、かな／英字、大文字／小文字等のモード切り替えを行わなければならず、それらのモードを切り替えるためには全角半角キーや Shift キー、CapsLock キーを押下する必要がある。指入力インターフェースを採用するモバイルデバイスのほとんどにはソフトウェアキーボードが搭載されているが、物理キーボードと同様にモード切り替えが必要である。

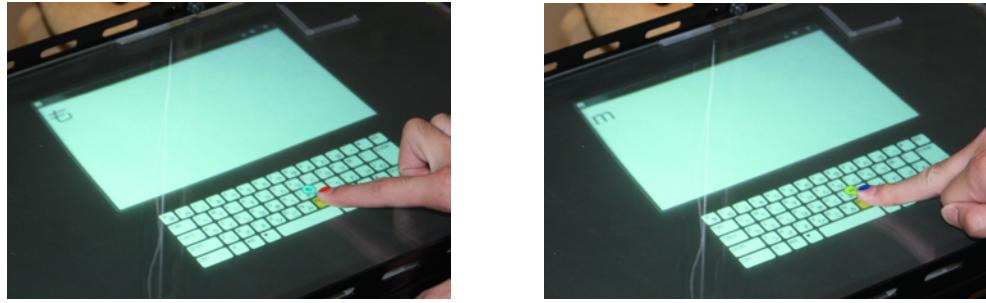
ソフトウェアキーボードに FSI を適用し、各指にキーセットを割り当てることで、これらのモード切り替えなしに文字入力が行えるようになる。具体的には、人差し指にかな、中指に小文字アルファベット、薬指に大文字アルファベットのキーセットを割り当てた。この割り当ては、予想されるキーの使用頻度と、指の動かしやすさを考慮に入れて行った。

本ソフトウェアキーボードは大画面ディスプレイ向けとモバイル向けの 2 種類を開発した。大画面ディスプレイ向けソフトウェアキーボードを図 7.16 に、モバイル向けを図 7.17 に示す。かなと英字が混在する文章を入力するにはモード切り替えの手間がかかるが、このキーボードを使用すると、片手で容易にそのような文章を入力できるようになる。特に、モバイルデバイスの場合はキーボード自体のサイズが小さく、両手による操作が困難であるため、モード切り替え操作には煩わしさを伴うことが多い。よって、FSI を適用したソフトウェアキーボードは特にモバイルデバイスで有用であると考えられる。

### 7.6.3 モバイル向け音楽プレイヤ

この音楽プレイヤは小型ディスプレイを持つモバイルデバイスを想定している。インターフェースとしてタッチパネルのみを持つモバイルデバイスは増加しており、それらを操作には物理的フィードバックがないため、必ずディスプレイを見ながら操作する必要がある。よって、デバイスをポケットやバッグに入れたまま操作を行えないという問題点がある。また、ディスプレイが小さいため、マルチタッチ操作は必ずしも快適に行えない。

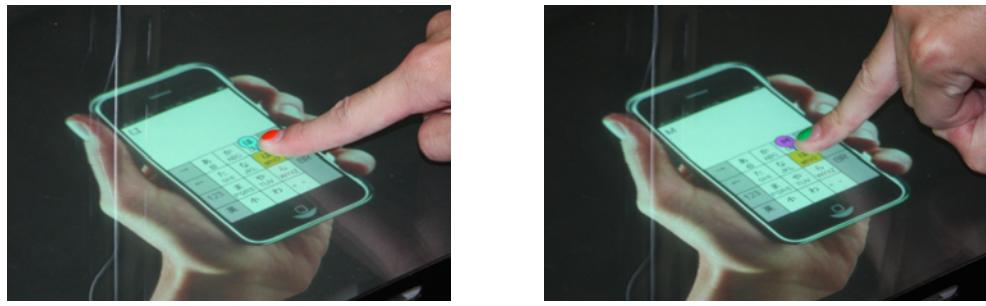
開発した音楽プレイヤを図 7.18 に示す。ディスプレイサイズが小さい指入力インターフェースを持つモバイルデバイスを模したデザインを採用している。このアプリケーションを用いて FSI を採用したモバイルデバイスのシミュレーションが行える。



(a) 人差し指にはかなキーセットが対応

(b) 中指には小文字アルファベットキーセットが対応

図 7.16: 大画面向けソフトウェアキー ボード



(a) 人差し指にはかなキーセットが対応

(b) 薬指には大文字アルファベットキーセットが対応

図 7.17: モバイルデバイス向けソフトウェアキー ボード

モバイルデバイスは片手で持ち、もう一方の手で操作することが想定されるため、操作には片手のみを利用するように設計とした。人差し指、中指、薬指、親指にはそれぞれ再生、一時停止、楽曲選択、音量調節機能を割り当てた。前者二つはトグル操作であるため、タッチするだけで機能が呼び出される。後者二つは“量”を入力する必要があるため、指をスライドさせる距離で量を入力する。

#### 7.6.4 CSCW 向けフォトビューア

グループワーク等の複数人で行う作業を支援することを想定したアプリケーションとしてフォトビューアを開発した。ペイントツールは大画面ディスプレイを持つ指入力インターフェースを用いた単独での作業を想定したアプリケーションである一方、本アプリケーションは複

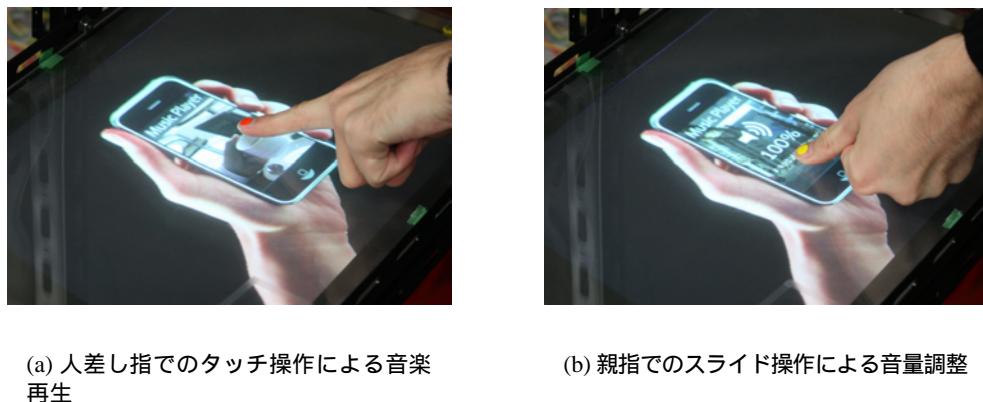


図 7.18: モバイル向け音楽プレイヤ

数人での作業を想定する。7.3.2 節で示した、タッチする指の違いから操作する人の違いも区別できることを活用する。

1人で操作する様子を図 7.19(a) に示す。1人が1本の指で操作すると写真の移動、2本の指で操作すると写真をスケーリングや回転が行える。一方、複数人で操作すると、同じ2点タッチ操作でも図 7.19(b) のように写真のコピー操作となる。コピーの種類は2種類存在し、2人がそれぞれ人差し指を写真の上にのせると写真をディープコピー、中指をのせるとシャローコピーする。

これらは共同編集と独自編集が混在しがちな CSCW では有効である。また、指ごとに操作の優先度を設定しておくことが可能で、ユーザは優先的に操作したいものは優先度の高い指で操作することにより、操作権を失わずに操作を継続することも可能になる。

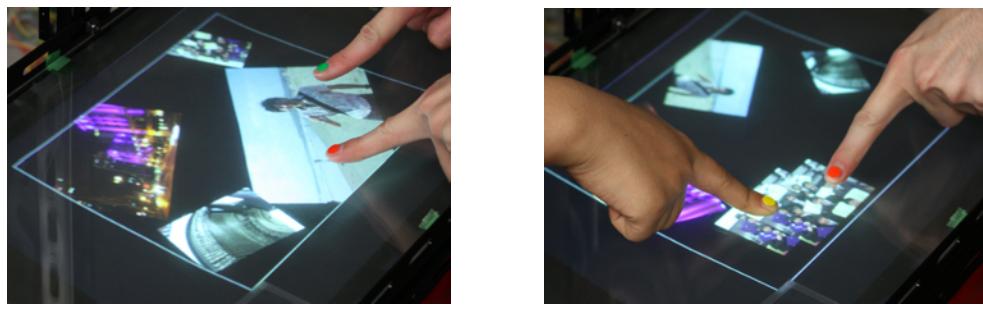


図 7.19: CSCW 向けフォトビューア

### 7.6.5 ショルダーサーフィンを防止する認証

指入力インターフェースにおける問題点の一つとして、認証に関する問題がある。指入力インターフェースはその性格上、操作の様子を操作者以外が閲覧することが容易であり、それが望まれている部分もある。一方で、パスフレーズを入力するような認証操作はあまり見られたくない行為であるが、認証行為を隠すような振る舞いをすることは、周囲の人間を信用していないというシグナルとして捉えられかねなく、場面によっては行いにくい行為である[37]。

一般的な認証方法として、PIN (Personal Identification Number) 認証がある。これは数桁の数字により個人認証を行う方法である。この PIN 認証は指入力インターフェースにおいても頻繁に利用されている認証方法であるが、指入力インターフェースにおける PIN 認証の使用はショルダーサーフィン攻撃に特に脆弱である。

このような指入力インターフェースにおける認証問題を解決するために、FSI を用いた PIN 認証アプリケーション、FSI-PIN 認証を開発した。通常の PIN 認証では数桁の数字を入力するが、FSI-PIN 認証では各数字に対して入力に使用する指情報を付加する。つまり、認証を成功させるためには、図 7.20 のように正しい指で正しい数字を入力する必要がある。

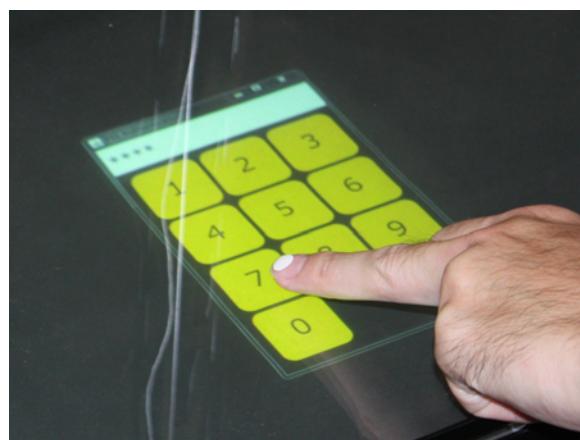


図 7.20: FSI-PIN 認証を操作する様子

これはショルダーサーフィンの防止に対して大きく二つの効果をもたらす。まず、ショルダーサーフィンを試みる攻撃者は、数字だけでなく、数字をタッチする指も記憶する必要がある。FSI-PIN 認証では両手 10 本全ての指を用いた操作を想定しているため、 $n$  術のパスフレーズの場合、PIN 認証と比較して FSI-PIN 認証でのパスフレーズの組み合わせは  $10^n$  倍になる。よって、攻撃者はパスフレーズを記憶することが困難になる。また、FSI-PIN 認証では両手を用いた操作であるため、必然的に入力パネルを両手で覆うような操作スタイルになる。よって、ユーザが意図的に認証行為を隠す振る舞いをすることなく、操作するだけで自然と操作の隠蔽が行われる。これは手元を隠す行為に関する問題をうまく解決しているといえる。

FSI-PIN 認証は、7.4.1 節で述べたような、指の組み合わせが膨大で記憶しづらいという FSI の欠点を逆に有効活用したアプリケーションであるといえる。

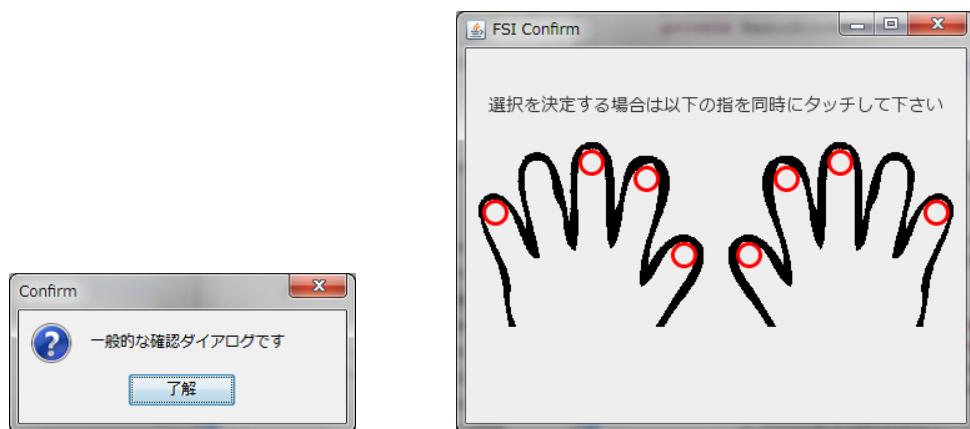
### 7.6.6 運動特性を活用した確認ダイアログ

一般的な確認ダイアログは図 7.21(a) のように、ボタンを押下するだけで確認作業が終わってしまう。よって、確認ダイアログはその内容をよく確認されることなく閉じられてしまうことがある。

これを防ぐために、確認ダイアログに FSI を適用した。7.4.2 節で述べたように、薬指以外の 4 本の指をタッチする動作や、中指と小指のみをタッチする動作は人間にとって多少困難であるという指の運動特性がある。開発した確認ダイアログは、この負の運動特性を有効活用したアプリケーションである。

本確認ダイアログは図 7.21(b) のように、タッチする指をユーザに提示する。確認ダイアログを閉じるために、ユーザは提示された指のみをタッチする必要がある。ユーザにタッチさせる指は薬指以外の 4 本や、中指と小指のみ等、人間が行いにくい指の組み合わせになっている。提示される指の組み合わせは常に同じではなく、ダイアログごとにランダムに提示されるため、動作に慣れることを防ぐことができる。この確認ダイアログにより、ユーザがすぐにダイアログを閉じることがなくなり、ダイアログの内容を確認するための時間がとれるようになることが期待できる。

本確認ダイアログは、FSI の構築にあたって障壁となり得る指の負の運動特性を、逆に有効活用したアプリケーションであるといえる。



(a) 一般的な確認ダイアログ

(b) Finger-Specific Interaction を用いた確認ダイアログ

図 7.21: 確認ダイアログ

## 7.7 実装に関する議論

7.5.1 節で説明したプロトタイプシステムでは、指を識別するために単色のカラーマーカーを用いた画像処理ベースの方法を採用している。この方法でもある程度高精度な指検出が可能であるが、同時に検出できる指の数には限界がある。高解像度のカメラを用いることで、現在の単純なカラーマーカーだけでなく、色のパターンやマーカーの形状も利用できるようになる。さらに、マーカーを指輪やつけ爪等にし、ファッショングループで楽しめるものにすることで、積極的な利用を促進できる可能性もある。また、手指認識アルゴリズム [57, 102, 118] を導入することで、マーカレスの実装が可能になることも期待できる。画像処理ベースで実装する場合はこのような改善案が考えられる。

一方、この画像処理ベースの手法では、モバイルデバイス向けに実用的な実装を行うことが難しい。別アプローチとして、人間の体に微弱な電流を流し、タッチする指から伝わる微弱な電流を検出するという電気的メカニズムを利用した方法が考えられる。その他にも、指に小型の加速度センサを付加し、指の動作から指のタッチを検出する方法や、表面筋電位を利用する方法等が考えられる。

これらの実装案を実現することは FSI を発展させるために必要不可欠であり、指入力インターフェースのインタラクションスタイルを一変させる可能性を秘めた重要な課題である。

## 第8章 結論

本章では、本研究により得られた成果を簡潔に述べると共に、本研究のヒューマン・コンピュータ・インターラクション分野、およびタッチインターフェースに対する貢献について整理する。また、それらに対する今後の展望と課題について述べる。

### 8.1 研究成果の概要

本研究では、コンピュータ操作における人間の目標とシステムの物理状態を隔てている淵を最小化するために、人間が経験的に体得してきた知的能力や身体的能力をコンピュータとのインターラクションに活用可能にすることを目指した。

そのためのアプローチとして、可併行性動作（併行動作）という動作を定義し、これをインターラクションに利用することを提唱した。併行動作とは、既存の操作を構成する動作と同時に、かつ同一の肢体で遂行可能な動作である。併行動作は、既存のインターフェースへと適用することで、既存操作との互換性や連続性を保持したまま、入力チャネルを増加することができるという特徴がある。

併行動作に基づくインターラクション手法の実現可能性を示すために、併行動作の具体的な適用例として、ペン入力インターフェースと指入力インターフェースへの適用を行った。それぞれのインターフェースにおいて、併行動作として適切な動作を考察し、その動作を利用したインターラクション手法を提案した。さらに、提案したインターラクション手法を実現するためのハードウェアおよびソフトウェアを開発し、アプリケーション例を示した。また、被験者実験を行い、提案したインターラクション手法の有用性と共に、提案手法の問題点や改善点の検証を行った。

ペン入力インターフェースと指入力インターフェースの実システム開発においては、2.5節で述べた三つの技術的課題を解決した。

#### ハードウェアレベルの課題の解決

課題として、併行動作を検出するためのハードウェア実装、および既存インターフェースの操作を妨げないハードウェア設計を挙げた。これに対して、空中での手の動作を検出するために加速度センサを利用した Context Sensitive Stylus (CS Stylus)、指の動作を検出するために感圧センサを利用した Pressure-Sensitive Stylus (PS Stylus)、および指を識別するために複数台のカメラやアクリル板、赤外線 LED、偏光フィルム等を用いたテーブルトップインターフェースの設計と開発を行い、実際にそれらが利用可能であることを示すことで課題を解決した。

### 動作検出用ソフトウェアレベルの課題の解決

課題として、開発したハードウェアから検出できるセンサ値を用いて、併行動作を検出するための動作検出用ソフトウェアの開発を挙げた。これに対して、CS Stylus から空中での手の動作を検出するために DP マッチング、PS Stylus から指の動作を検出するためにセンサ値の時間変化を用いた。また、テーブルトップインターフェースにおけるタッチする指の識別では、タッチ位置の検出と指の検出を個別に行い、その相対位置を用いる手法を開発した。そして、これらが実際に利用可能であることを示すことで課題を解決した。

### アプリケーションレベルの課題の解決

併行動作により新たに創出したインタラクション手法の扱い方、創出したインタラクション手法の導入が適したアプリケーションの開発、およびアプリケーションの機能との対応付けを課題として挙げた。これに対して、ペン入力インターフェースでは、ペンメタファの導入やペイントツールや地図ビューアの開発、およびペン入力インターフェースが苦手とするモード切り替えやパラメータ選択操作への応用を示すことで課題を解決した。指入力インターフェースでは、指の運動特性を考慮したアプリケーション設計、指入力インターフェースが苦手とする認証アプリケーションやディスプレイを視認せずに操作可能なアプリケーションの開発、および両手を使ったタスクや指の運動特性を考慮した機能割り当てを行うことで課題を解決した。

ペン入力インターフェースと指入力インターフェースに対して併行動作を応用した結果、それぞれのインターフェースにおいて既存操作との互換性や連続性を保持したまま、入力操作の数を増加できることを確認した。これら一連の応用結果より、併行動作に基づくインタラクション手法の実現可能性、およびその有効性を示すことができた。

## 8.2 貢献

研究全体の貢献として、ヒューマン・コンピュータ・インタラクション分野への貢献について筆者の主張を述べる。また、本研究はペン入力インターフェースと指入力インターフェースの発展にも寄与した。それぞれのインターフェースに対する貢献について述べる。

### 8.2.1 ヒューマン・コンピュータ・インタラクション分野への貢献

本研究のヒューマン・コンピュータ・インタラクション分野における貢献は以下の 3 点に整理することができる。

#### インタラクションに利用可能な人間の能力の増幅

本研究の最大の貢献は、併行動作に基づくインタラクション手法を提唱したことで、人間が経験的に体得してきた知的能力や身体的能力をコンピュータとのインタラクション

に活用可能にし、人間がコンピュータとのインタラクションに利用できる能力を増幅させたことである。これにより、人間がコンピュータとのインタラクションに利用できる情報伝達チャネルが増加し、コンピュータに人間の操作意図を伝えやすくなつた。その結果として、人間の目標とシステムの物理状態の間に存在する「実行の淵」をさらに小さくすることができた。

#### 併行動作の提唱

数ある人間の動作の中から、インタラクションに利用価値のある動作として併行動作を定義し、その利用を提唱した。併行動作とは、既存の操作を構成する動作と同時に、かつ同一の肢体で遂行可能な動作であり、共通動作との互換性や基本動作群との連続性を持つ。この併行動作をインタラクションに応用することで、既存操作との互換性や連続性を保持したまま、入力チャネルを増加することができるようになった。

#### 併行動作のタッチインターフェースへの適用

併行動作に基づくインタラクション手法の実現可能性を示すために、併行動作の具体的な適用例として、タッチインターフェースへの適用を試みた。ペン入力インターフェースでは、併行動作として空中での手の動作とペンを握る指の動作、ペンを握る力を加減する動作を適用し、指入力インターフェースでは、併行動作として指を使い分ける動作を適用した。これらタッチインターフェースへの適用を行い、それらの有効性を示したことで、併行動作に基づくインタラクション手法の実現可能性と有効性を示した。

#### 8.2.2 ペン入力インターフェースへの貢献

ペン入力インターフェースの入力動作として、空中での手の動作とペンを握る指の動作、ペンを握る力を加減する動作はこれまで十分に検討されておらず、インタラクション設計の余地が残されていた。これらの動作をインタラクションに利用可能にし、インタラクション設計の余地を有効活用したことがペン入力インターフェースに対する大きな貢献である。

また、4.1.2 節で述べたペン入力インターフェースに内在する課題の解決にも貢献した。まず、空中での手の動作をインタラクションに利用することで、ペン先がディスプレイに接していない状態でもインタラクションが可能になった。これにより、ペンがディスプレイに近づいた際にペン先によってユーザの視線が遮られてしまう問題の解決に貢献した。また、入力操作数を増加させたことにより、ペンのみを用いてモード切り替えやパラメータ選択等が可能になり、メニュー等の GUI 操作の必要性を減少させることができた。これにより、細かいポインティング操作により主作業への意識や集中力が奪われてしまう問題や、大画面ディスプレイにおける GUI 操作の困難さの解決に貢献した。

#### 8.2.3 指入力インターフェースへの貢献

指入力インターフェースの入力動作として、今まで無視されていた指を使い分ける動作をインタラクションに利用可能にしたことが指入力インターフェースに対する大きな貢献である。

指入力インターフェースはモバイルデバイスを中心にさまざまなデバイスに搭載されており、マルチタッチ入力が主流になりつつある。モバイルデバイスの利用シーンを考えると、デバイスを非利き手で持ち、利き手の指で操作するという利用スタイルが多く、またモバイルデバイスに搭載されているディスプレイサイズも小さい。このような特徴を考えると、全てのデバイスにおいて、マルチタッチ入力は必ずしも適当なインタラクション手法ではない。本研究では、指入力インターフェースの新たなインタラクション手法として、タッチする指を識別するインタラクション手法を提案した。指入力インターフェースのインタラクション手法として新たな方向性と可能性を示したことで、マルチタッチ入力の利用が主流になりつつある指入力インターフェースの設計指針に風穴を開けたといえる。このことも指入力インターフェースに対する本研究の貢献である。

また、7.1.2節で述べた指入力インターフェースに内在する課題の解決にも貢献した。指入力インターフェースは指だけで操作が行えるという利点を持つ一方で、タッチした指の座標値しか入力情報を持たないため、コマンド選択やパラメータ入力等を行うためにはGUI操作が必須となっていた。指を識別可能にすることで、指の組み合わせの利用も含めて入力情報量は格段に増加し、GUI操作の必要ない操作体系を構築可能になった。

さらに、オクルージョン問題やフィードバック問題の解決にも貢献した。これらの問題は、指入力インターフェースの操作においてディスプレイ上のオブジェクトを正確にタッチすることが要求されるために問題となっているとも考えることができる。既存の指入力インターフェースではディスプレイに表示されたオブジェクトをタッチすることで初めて意味をなす入力となっていたが、指の識別が可能になったことにより指でタッチする行為 자체が意味を持つ入力となった。つまり、指入力インターフェースの操作に必ずしも正確なタッチが必要なくなった。よって、ディスプレイが隠れてしまうオクルージョン問題や、どのオブジェクトを操作したかがわからないフィードバック問題を間接的に解決できたといえる。これに伴い、ディスプレイを視認することなく操作を行うeyes-freeインタラクションも実現できた。

### 8.3 今後の課題と展望

最後に、本研究を進める中で明らかとなった課題や、今後の展望について述べる。研究全体としての今後の課題と展望、およびペン入力インターフェースと指入力インターフェースそれぞれにおける今後の課題と展望について述べる。

#### 8.3.1 全体としての課題と展望

本研究では、人間が経験的に体得してきた知的能力や身体的能力をコンピュータとのインタラクションに活用可能にすることを目指し、併行動作という身体動作を利用した。また、3.1節で述べた関連研究でもいくつかの知的能力や身体的能力を利用している。しかしながら、本研究や関連研究で活用した知的能力や身体的能力ごく一部であり、活用されずに残っている知的能力や身体的能力はまだ多く存在する。このようなコンピュータとのインタラクション

に使われていない人間の能力を見つけ、それを利用可能にすることが今後の研究として期待される。

また、本研究で提案した併行動作は、人間が無意識的にも行える動作であり、それをコンピュータとのインタラクションに利用可能にした。これを別の観点から見ると、併行動作は今まで無意識的に行っており、インタラクションとは無関係の動作であったが、それをインタラクションに応用したことで、それらの動作を意識的に行わなければならなくなつとも捉えることができる。つまり、適切なインタラクション設計を行わないと、無意識的な動作により誤入力や誤操作が多発する恐れがあるといえる。本論文で示した適用例では適切な設計を行つたためにこれは問題とならなかつたが、新たに併行動作に基づくインタラクション手法を構築する場合にはこの点に注意を払う必要がある。今後は、このような問題の発生を防ぐために、個々のアプリケーションのインタラクション設計に適用可能なフレームワーク構築が望まれる。

### 8.3.2 ペン入力インターフェースに対する課題と展望

本研究では、併行動作として、空中での手の動作、およびペンを握る指の動作に着目し、それらをペン入力インターフェースへと適用したが、人間がペンを扱う様子を観察すると、利用できる可能性のある動作が多く残っている。たとえば、指の関節の動作や手首の動作等は利用できる可能性があると思われる。このような残存する併行動作を見つけ出し、それをペン入力インターフェースに応用することは今後の課題の一つである。

また、これらの動作をユーザの能動的な動作としてインタラクションに応用するだけではなく、6.8.2節で述べた行動推定アプリケーションのように、無意識的な動作としてこれらの動作を活用し、ユーザの意図を読み取り適切な支援を自動的に行うような応用も求められる。

ペン入力インターフェースは文字や絵を描く等の創造活動を支援することを得意としており、それらを主とするペイントツール等のためのインターフェースとしては優れている。一方で、ペン入力インターフェースはそれ以外の有効なアプリケーションはほとんど存在しない。ペイントツール以外の、ペン入力インターフェースの利点や特徴を活かした真に有効なアプリケーションを開発することも課題として挙げられる。

### 8.3.3 指入力インターフェースに対する課題と展望

本研究で指入力インターフェース向けのインタラクション手法のコンセプトとして提案した Finger-Specific Interaction (FSI) は、指だけで入力できる情報量を格段に増やす一方で、ユーザはアプリケーションごとの指と機能との対応関係を記憶する必要がある。7.4.1節で述べたように、本研究では指の持つ文化的な意味や指輪、つけ爪を用いた仕組みを提案したが、更なる検討の余地は残されている。今後の課題として、記憶を助けるための更なる知的能力の活用方法や仕組みの考案、さらに記憶を容易にするアプリケーション設計のためのフレームワーク構築等が必要になる。

本研究では、指の動かしやすさや機能の使用頻度等を考慮し、アプリケーションごとに指と機能との適切な割り当てを決定したが、この指と機能との割り当てを最適化するためには多くの労力を要する。そこで、使いやすさの観点から指に対する機能の割り当てを定めたガイドラインを策定する必要がある。

また、7.6節で多くの応用例を示したように、大画面ディスプレイ向けやモバイルデバイス向け、CSCW向け等、指入力インターフェースはさまざまなデバイスや場面において利用されているため、全ての指入力インターフェースに適合する統一的ガイドラインの作成は困難であると考えられる。一方で、全く異なる複数のガイドラインに沿って開発されたアプリケーションはユーザビリティの低下を招く恐れがある。そこで、指と機能との割り当てに関するガイドラインは、全体として一定の統一感を持たせつつ、デバイスごとや利用場面ごとに切り分けて策定する必要がある。

本研究において開発したテーブルトップインターフェースは、7.5.1節で示したようにプロトタイプシステムであり、そのデバイス構成はモバイル環境での実装には適していない。7.7節に示したような実装や、従来とは全く異なるメカニズムを持つ指入力インターフェースの実現が必要である。

指入力インターフェースにおける全般的な課題としては、オクルージョンやフィードバックの問題がある。8.2.3節で述べたように、本研究によりこれらの問題を間接的に解決した。また、オクルージョン問題を解決するために表示位置を変更[90]やオクルージョンを気にせずに操作が行えるGUIの提案[55]、さらにフィードバック問題を解決するために指入力インターフェースに触覚フィードバックの導入を試みる研究[63, 117]等が行われ、これらの問題の解決を目指しているが、まだ検討は不十分であると思われる。今後の指入力インターフェースの発展のためにはこれらの問題を解決が望まれる。

また、指入力インターフェースは指だけで操作できることや、複数人での作業が行いやすいことから、モバイルデバイスやCSCW向けのテーブルトップインターフェースとして利用されているが、一方で、ペン入力インターフェースと同様に真に有効なアプリケーションはほとんど存在していない。指入力インターフェースの特徴や利点を最大限に活用したアプリケーションや、指入力インターフェースでのみ実現可能な独創的アプリケーションを開発することも今後期待される。

## 謝辞

本論文をまとめるにあたり、指導教員である筑波大学大学院システム情報工学研究科コンピュータサイエンス専攻の三末和男准教授、および副指導教員である田中二郎教授には終始多大な助言を頂きました。本論文に関する指導に加えて、2006年の博士前期課程入学以来、現在に至るまでの約5年間に渡り、研究の進め方や発表資料の作成に関するノウハウ、発表の仕方等、多岐に渡るご指導を頂きました。また、ティーチングアシスタントや研究補助員の職に就く機会を与えて頂いたことにより、教育や学生指導に関する知識と経験を得ることができました。

筑波大学大学院システム情報工学研究科コンピュータサイエンス専攻の高橋伸准教授、および志築文太郎講師にはゼミでの貴重なご助言だけでなく、個人的にも研究の相談に乗って頂くと共に、激励の言葉を頂きました。

本論文の審査にあたり、田中二郎教授および三末和男准教授と共に審査委員を務めて頂いた筑波大学大学院システム情報工学研究科コンピュータサイエンス専攻の北川博之教授、および三谷純准教授、知能機能システム専攻の葛岡英明教授、大学院図書館情報メディア研究科図書館情報メディア専攻の井上智雄准教授には的確なご指摘を頂き、本論文をまとめるに至りました。

博士後期課程においては、グローバルCOEプログラム“サイバニクス：人・機械・情報系の融合複合”に研究補助員として教育・研究プログラムに参加させて頂きました。本プログラムにおいて経済的支援を頂いたと共に、世界水準の研究に携わる機会や、異分野の研究に関する知識や複眼的思考を学ぶ機会を与えて頂きました。これに関して、本プログラムの拠点リーダである筑波大学大学院システム情報工学研究科知能機能システム専攻の山海嘉之教授をはじめ、多くの先生方、および本プログラムにおける活動の補助をして頂いた事務局スタッフの方々には多大なご支援を頂きました。

筑波大学大学院システム情報工学研究科コンピュータサイエンス専攻の事務室の方々には研究支援に関する事務手続きでの支援に加えて、さまざまな面においてお世話になりました。

所属研究室であるインタラクティブプログラミング研究室の学生、およびOB・OGの方々には、ゼミ等を通じてご意見を頂き、生活面でもさまざまなご支援を頂きました。特に、OGのSimona Vasilacheさんには英文の校正に多くの時間を割いて頂きました。

最後に、研究を続けることを快諾し、研究生活を支援してくれた家族、および特に精神面を支え、研究を応援してくれた友人等があり、ここまで研究を進めることができました。

以上の方々に、心より感謝を申し上げます。ありがとうございました。

## 参考文献

- [1] Johnny Accot and Shumin Zhai. More than dotting the i's - Foundations for crossing-based interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'02)*, pp. 73–80, 2002.
- [2] Georg Apitz and François Guimbretière. CrossY: A Crossing-Based Drawing Application. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology (UIST'04)*, pp. 3–12, 2004.
- [3] Gilles Bailly, Eric Lecolinet, and Yves Guiard. Finger-Count & Radial-Stroke Shortcuts: Two Techniques for Augmenting Linear Menus on Multi-Touch Surfaces. In *Proceedings of the 28th International Conference on Human Factors in Computing Systems (CHI'10)*, pp. 591–594, 2010.
- [4] Hrvoje Benko, Andrew D. Wilson, and Patrick Baudisch. Precise Selection Techniques for Multi-Touch Screens. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'06)*, pp. 1263–1272, 2006.
- [5] Xiaojun Bi, Tomer Moscovich, Gonzalo Ramos, Ravin Balakrishnan, and Ken Hinckley. An Exploration of Pen Rolling for Pen-based Interaction. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology (UIST'08)*, pp. 191–200, 2008.
- [6] Florian Block, Michael Haller, Hans Gellersen, Carl Gutwin, and Mark Billinghurst. VoodooSketch — Extending Interactive Surfaces with Adaptable Interface Palettes. In *Proceedings of the 2nd International Conference on Tangible and Embedded Interaction (TEI'08)*, pp. 55–58, 2008.
- [7] Richard A. Bolt. “Put-that-there”: Voice and gesture at the graphics interface. In *Proceedings of the 7th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'80)*, pp. 262–270, 1980.
- [8] Scott Brave, Hiroshi Ishii, and Andrew Dahley. Tangible Interfaces for Remote Collaboration and Communication. In *Proceedings of the ACM 1998 Conference on Computer Supported Cooperative Work (CSCW'98)*, pp. 169–178, 1998.
- [9] Grigore C. Burdea. *Force and touch feedback for virtual reality*. John Wiley & Sons, Inc., 1996.

- [10] Jack Callahan, Don Hopkins, Mark Weiser, and Ben Shneiderman. An Empirical Comparison of Pie vs. Linear Menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'88)*, pp. 95–100, 1988.
- [11] Xiang Cao, Andrew D. Wilson, Ravin Balakrishnan, Ken Hinckley, and Scott E. Hudson. ShapeTouch: Leveraging Contact Shape on Interactive Surfaces. In *Proceedings of 3rd IEEE International Workshop on Horizontal Interactive Human Computer Systems (TABLETOP'08)*, pp. 139–146, 2008.
- [12] Stuart K. Card, Thomas P. Moran, and Allen Newell. *The Psychology of Human-Computer Interaction*. Lawrence Erlbaum Associates, Inc. Publishers, 1984.
- [13] Jared Cechanowicz, Pourang Irani, and Sriram Subramanian. Augmenting the Mouse with Pressure Sensitive Input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'07)*, pp. 1385–1394, 2007.
- [14] Andrew Dahley, Craig Wisneski, and Hiroshi Ishii. Water Lamp and Pinwheels: Ambient Projection of Digital Information into Architectural Space. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'98)*, pp. 269–270, 1998.
- [15] Paul Dietz and Darren Leigh. DiamondTouch: a Multi-User Touch Technologys. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology (UIST'01)*, pp. 219–226, 2001.
- [16] Rebecca Fiebrink, Dan Morris, and Meredith Ringel Morris. Dynamic Mapping of Physical Controls for Tabletop Groupware. In *Proceedings of the 27th International Conference on Human Factors in Computing Systems (CHI'09)*, pp. 471–480, 2009.
- [17] George W. Fitzmaurice, Hiroshi Ishii, and William A. S. Buxton. Bricks: Laying the Foundations for Graspable User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'95)*, pp. 442–449, 1995.
- [18] Dustin Freeman, Hrvoje Benko, Meredith Ringel Morris, and Daniel Wigdor. ShadowGuides: Visualizations for In-Situ Learning of Multi-Touch and Whole-Hand Gestures. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces (ITS'09)*, pp. 165–172, 2009.
- [19] Masaaki Fukumoto and Toshiaki Sugimura. Active Click: Tactile Feedback for Touch Panels. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'01)*, pp. 121–122, 2001.
- [20] Jörg Geißler. Shuffle, throw or take it! Working Efficiently with an Interactive Wall. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'98)*, pp. 265–266, 1998.

- [21] Yves Guiard. Asymmetric Division of Labor in Human Skilled Bimanual Action: The Kinematic Chain as a Model. *Journal of Motor Behavior*, Vol. 19, pp. 486–517, 1987.
- [22] François Guimbretiére and Terry Winograd. FlowMenu: Combining Command, Text, and Data Entry. In *Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology (UIST'00)*, pp. 213–216, 2000.
- [23] Jefferson Y. Han. Low-Cost Multi-Touch Sensing through Frustrated Total Internal Reflection. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology (UIST'05)*, pp. 115–118, 2005.
- [24] Beverly L. Harrison, Kenneth P. Fishkin, Anuj Gujar, Carlos Mochon, and Roy Want. Squeeze Me, Hold Me, Tilt Me! An Exploration of Manipulative User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'98)*, pp. 17–24, 1998.
- [25] Otmar Hilliges, Shahram Izadi, Andrew D. Wilson, Steve Hodges, Armando Garcia-Mendoza, and Andreas Butz. Interactions in the Air: Adding Further Depth to iIteractive Tabletops. In *Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology (UIST'09)*, pp. 139–148, 2009.
- [26] Ken Hinckley, Patrick Baudisch, Gonzalo Ramos, and François Guimbretiére. Design and Analysis of Delimiters for Selection-Action Pen Gesture Phrases in Scriboli. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'05)*, pp. 451–460, 2005.
- [27] Ken Hinckley, Randy Pausch, Dennis Proffitt, James Patten, and Neal Kassell. Cooperative Bimanual Action. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'97)*, pp. 27–34, 1997.
- [28] Matthew Hirsch, Douglas Lanman, Henry Holtzman, and Ramesh Raskar. BiDi Screen: A Thin, Depth-Sensing LCD for 3D Interaction using Lights Fields. *ACM Transations on Graphics*, Vol. 28, No. 5, pp. 159:1–159:9, 2009.
- [29] Don Hopkins. The Design and Implementation of Pie Menus. *Dr. Dobb's Journal*, Vol. 16, No. 12, pp. 16–26, 1991.
- [30] Hiroshi Ishii and Brygg Ullmer. Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'97)*, pp. 234–241, 1997.
- [31] Sergi Jordà, Günter Geiger, Marcos Alonso, and Martin Kaltenbrunner. The reacTable: Exploring the Synergy between Live Music Performance and Tabletop Tangible Interfaces.

- In *Proceedings of the 1st International Conference on Tangible and Embedded Interaction (TEI'07)*, pp. 139–146, 2007.
- [32] 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)*, p. 17:1, 2008.
- [33] Sho Kamuro, Kouta Minamizawa, Naoki Kawakami, and Susumu Tachi. Pen de Touch. In *ACM SIGGRAPH 2009 Emerging Technologies (SIGGRAPH'09)*, p. 17:1, 2009.
- [34] Eric R. Kandel, James H. Schwartz, and Thomas M. Jessell. *Principles of Neural Science*. McGraw-Hill, 2000.
- [35] Shaun K. Kane, Jeffrey P. Bigham, and Jacob O. Wobbrock. Slide Rule: Making Mobile Touch Screens Accessible to Blind People Using Multi-Touch Interaction Techniques. In *Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '08)*, pp. 73–80, 2008.
- [36] Thomas Kienzl, Ulf Marsche, Nadja Kapeller, and Adam Gokcezade. tangible workbench “TW” — with changeable markers. In *ACM SIGGRAPH 2008 New Tech Demos (SIGGRAPH'08)*, p. 37:1, 2008.
- [37] David Kim, Paul Dunphy, Pam Briggs, Jonathan Hook, John Nicholson, James Nicholson, and Patrick Olivier. Multi-Touch Authentication on Tabletops. In *Proceedings of the 28th International Conference on Human Factors in Computing Systems (CHI'10)*, pp. 1093–1102, 2010.
- [38] Seoktae Kim, Hyunjung Kim, Boram Lee, Tek-Jin Nam, and Woohun Lee. Inflatable Mouse: Volume-adjustable Mouse with Air-pressure-sensitive Input and Haptic Feedback. In *Proceedings of the 26th Annual SIGCHI Conference on Human Factors in Computing Systems (CHI'08)*, pp. 211–214, 2008.
- [39] Hideki Koike, Wataru Nishikawa, and Kentaro Fukuchi. Transparent 2-D Markers on an LCD Tabletop System. In *Proceedings of the 27th International Conference on Human Factors in Computing Systems (CHI'09)*, pp. 163–172, 2009.
- [40] Russell Kruger, Sheelagh Carpendale, Stacey D. Scott, and Anthony Tang. Fluid Integration of Rotation and Translation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'05)*, pp. 601–610, 2005.
- [41] Gordon P. Kurtenbach, Abigail J. Sellen, and William A.S. Buxton. An Empirical Evaluation of Some Articulatory and Cognitive Aspects of Marking Menus. *Journal of Human Computer Interaction*, Vol. 8, No. 1, pp. 1–23, 1993.

- [42] Gordon Kurtenbach and William Buxton. The Limits Of Expert Performance Using Hierarchic Marking Menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'93)*, pp. 482–487, 1993.
- [43] Ki-Uk Kyung and Jun-Young Lee. wUbi-Pen : Windows Graphical User Interface Interacting with Haptic Feedback Stylus. In *ACM SIGGRAPH 2008 New Tech Demos (SIGGRAPH'08)*, pp. 42:1–42:4, 2008.
- [44] G. Julian Lepinski, Tovi Grossman, and George Fitzmaurice. The Design and Evaluation of Multitouch Marking Menus. In *Proceedings of the 28th International Conference on Human Factors in Computing Systems (CHI'10)*, pp. 2233–2242, 2010.
- [45] Erh li Early Shen, Sung sheng Daniel Tsai, Hao hua Chu, Yung jen Jane Hsu, and Chi wen Euro Chen. Double-side Multi-touch Input for Mobile Devices. In *Proceedings of the 27th International Conference on Human Factors in Computing Systems (CHI'09)*, pp. 4339–4344, 2009.
- [46] m. c. schraefel, Grham Smith, and Patrick Baudisch. Curve Dial: Eyes-Free Parameter Entry for GUIs. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'05)*, pp. 1146–1147, 2005.
- [47] Shahzad Malik and Joe Laszlo. Visual Touchpad: A Two-handed Gestural Input Device. In *Proceedings of the 6th international conference on Multimodal interfaces (ICMI'04)*, pp. 289–296, 2004.
- [48] Shahzad Malik, Abhishek Ranjan, and Ravin Balakrishnan. Interacting with Large Displays from a Distance with Vision-Tracked Multi-Finger Gestural Input. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology (UIST'05)*, pp. 43–52, 2005.
- [49] Jennifer Mankoff and Gregory D. Abowd. Cirrin: A word-level unistroke keyboard for pen input. In *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology (UIST'98)*, pp. 213–214, 1998.
- [50] Nobuyuki Matsushita and Jun Rekimoto. HoloWall: Designing a Finger, Hand, Body, and Object Sensitive Wall. In *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology (UIST'97)*, pp. 209–210, 1997.
- [51] Mark T. Maybury. Intelligent Multimedia Interfaces. *AI Magazine*, Vol. 13, No. 2, pp. 35–37, 1992.
- [52] Pranav Mistry, Pattie Maes, and Liyan Chang. WUW - Wear Ur World - A Wearable Gestural Interface. In *Proceedings of the 27th International Conference on Human Factors in Computing Systems (CHI'09)*, pp. 4111–4116, 2009.

- [53] Motoki Miura and Susumu Kunifuji. RodDirect: Two-Dimensional Input with Stylus Knob. In *Proceedings of the 8th Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI'06)*, pp. 113–120, 2006.
- [54] Sachi Mizobuchi, Shinya Terasaki, Tero Keski-Jaskari, Jari Nousiainen, Matti Ryynanen, and Miika Silfverberg. Making an Impression: Force-Controlled Pen Input for Handheld Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'05)*, pp. 1661–1664, 2005.
- [55] Tomer Moscovich. Contact Area Interaction with Sliding Widgets. In *Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology (UIST'09)*, pp. 13–22, 2009.
- [56] Donald A. Norman, 野島 久雄(訳). 誰のためのデザイン? — 認知科学者のデザイン原論. 新曜社, 1990.
- [57] Kenji Oka, Yoichi Sato, and Hideki Koike. Real-Time Fingertip Tracking and Gesture Recognition. *IEEE Computer Graphics and Applications*, Vol. 22, No. 6, pp. 64–71, 2002.
- [58] Alex Olwal, Steven Feiner, and Susanna Heyman. Rubbing and Tapping for Precise and Rapid Selection on Touch-Screen Displays. In *Proceedings of the 26th Annual SIGCHI Conference on Human Factors in Computing Systems (CHI'08)*, pp. 295–304, 2008.
- [59] James Patten, Ben Recht, and Hiroshi Ishii. Audiopad: A Tag-based Interface for Musical Performance. In *Proceedings of the 2002 Conference on New Interfaces for Musical Expression (NIME'02)*, pp. 1–6, 2002.
- [60] Ken Perlin. Quikwriting: Continuous Stylus-based Text Entry. In *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology (UIST'98)*, pp. 215–216, 1998.
- [61] Thomas Pintaric, Thomas Kment, and Wolfgang Spreicer. SqueezeOrb: A Low-Cost Pressure-Sensitive User Input Device. In *Proceedings of the 2008 ACM Symposium on Virtual Reality Software and Technology (VRST'08)*, pp. 269–270, 2008.
- [62] Stuart Pook, Eric Lecolinet, Guy Vaysseix, and Emmanuel Barillot. Control Menus: Execution and Control in a Single Interactor. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'00)*, pp. 263–264, 2000.
- [63] Ivan Poupyrev and Shigeaki Maruyama. Tactile Interfaces for Small Touch Screens. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology (UIST'03)*, pp. 217–220, 2003.

- [64] Ivan Poupyrev, Jun Rekimoto, and Shigeaki Maruyama. TouchEngine: A Tactile Display for Handheld Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'02)*, pp. 644–645, 2002.
- [65] Gonzalo Ramos, Matthew Boulos, and Ravin Balakrishnan. Pressure Widgets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'04)*, pp. 487–494, 2004.
- [66] Adrian Reetz, Carl Gutwin, Tadeusz Stach, Miguel Nacenta, and Sriram Subramanian. Superflick: a Natural and Efficient Technique for Long-Distance Object Placement on Digital Tables. In *Proceedings of Graphics Interface 2006 (GI'06)*, pp. 163–170, 2006.
- [67] Jun Rekimoto. Pick-and-Drop: A Direct Manipulation Technique for Multiple Computer Environments. In *Proceedings of the 8th Annual ACM Symposium on User Interface Software and Technology (UIST'97)*, pp. 31–39, 1997.
- [68] Jun Rekimoto. SmartSkin: An Infrastructure for Freehand Manipulation on Interactive Surfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'02)*, pp. 113–120, 2002.
- [69] Jun Rekimoto, Brygg Ullmer, , and Haruo Oba. DataTiles: A Modular Platform for Mixed Physical and Graphical Interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'01)*, pp. 269–276, 2001.
- [70] Xiangshi Ren, Jibin Yin, Shengdong Zhao, and Yang Li. The Adaptive Hybrid Cursor: A Pressure-based Target Selection Technique for Pen-based User Interfaces. In *Proceedings of the 11th IFIP TC13 International Conference on Human-Computer Interaction (INTERACT 2007)*, LNCS4662, pp. 310–323, 2007.
- [71] Anne Roudaut, Eric Lecolinet, and Yves Guiard. MicroRolls: Expanding Touch-Screen Input Vocabulary by Distinguishing Rolls vs. Slides of the Thumb. In *Proceedings of the 27th International Conference on Human Factors in Computing Systems (CHI'09)*, pp. 927–936, 2009.
- [72] 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. 43–52, 2009.
- [73] Itiro Siio and Hitomi Tsujita. Mobile Interaction Using Paperweight Metaphor. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology (UIST'06)*, pp. 111–114, 2006.

- [74] G. M. Smith and m. c. schraefel. The Radial Scroll Tool: Scrolling Support for Stylus- or Touch-Based Document Navigation. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology (UIST'04)*, pp. 53–56, 2004.
- [75] Mandayam A. Srinivasan and Jyh-shing Chen. Human Performance in Controlling Normal Forces of Contact with Rigid Objects. *Advances in Robotics, Mechatronics, and Haptic Interfaces*, Vol. 49, pp. 119–125, 1993.
- [76] Sriram Subramanian, Dzimity Aliakseyeu, and Andrés Lucero. Multi-Layer Interaction for Digital Tables. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology (UIST'06)*, pp. 269–272, 2006.
- [77] Masanori Sugimoto and Keiichi Hiroki. HybridTouch: An Intuitive Manipulation Technique for PDAs Using Their Front and Rear Surfaces. In *Proceedings of the 8th Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI'06)*, pp. 137–140, 2006.
- [78] Atsushi Sugiura and Yoshiyuki Koseki. A User Interface Using Fingerprint Recognition - Holding Commands and Data Objects on Fingers -. In *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology (UIST'98)*, pp. 71–79, 1998.
- [79] Yu Suzuki, Kazuo Misue, and Jiro Tanaka. Interface for Digital Notes Using Stylus Motions Made in the Air. In *Proceedings of the 2nd International Conference on Knowledge, Information and Creativity Support Systems (KICSS'07)*, pp. 104–109, 2007.
- [80] Yu Suzuki, Kazuo Misue, and Jiro Tanaka. Stylus Enhancement to Enrich Interaction with Computers. In *Proceedings of 12th International Conference on Human-Computer Interaction (HCI International 2007)*, LNCS4551, pp. 133–142, 2007.
- [81] Yu Suzuki, Kazuo Misue, and Jiro Tanaka. Pen-based Interface Using Hand Motions in the Air. *The IEICE Transactions on Information and Systems*, Vol. E91-D, No. 11, pp. 2647–2654, 2008.
- [82] Yu Suzuki, Kazuo Misue, and Jiro Tanaka. Interaction Technique for a Pen-Based Interface Using Finger Motions. In *Proceedings of 13th International Conference on Human-Computer Interaction (HCI International 2009)*, LNCS 5611, pp. 503–512, 2009.
- [83] Yu Suzuki, Kazuo Misue, and Jiro Tanaka. Interaction Technique Combining Gripping and Pen Pressures. In *Proceedings of 14th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES'10)*, LNAI 6279, pp. 440–448, 2010.
- [84] Feng Tian, Xiang Ao, Hongan Wang, Vidya Setlur, and Guozhong Dai. The Tilt Cursor: Enhancing Stimulus-Response Compatibility by Providing 3D Orientation Cue of Pen. In

- Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'07)*, pp. 303–306, 2007.
- [85] Feng Tian, Lishuang Xu, Hongan Wang, Xiaolong Zhang, Yuanyuan Liu, Vidya Setlur, and Guozhong Dai. Tilt Menu: Using the 3D Orientation Information of Pen Devices to Extend the Selection Capability of Pen-based User Interfaces. In *Proceedings of the 26th Annual SIGCHI Conference on Human Factors in Computing Systems (CHI'08)*, pp. 1371–1380, 2008.
  - [86] Brygg Ullmer and Hiroshi Ishii. The metaDESK: Models and Prototypes for Tangible User Interfaces. In *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology (UIST'97)*, pp. 223–232, 1997.
  - [87] Marjorie Fink Vargas. *Louder Than Words: An Introduction to Nonverbal Communication*. Iowa State Press, 1986.
  - [88] Dan Venolia and Forrest Neiberg. T-Cube: A Fast, Self-Disclosing Pen-Based Alphabet. In *Proceedings of the 7th Annual ACM Symposium on User Interface Software and Technology (UIST'94)*, pp. 265–270, 1994.
  - [89] Kevin Vlack, Terukazu Mizota, Naoki Kawakami, Kazuto Kamiyama, Hiroyuki Kajimoto, and Susumu Tachi. GelForce: A Vision-based Traction Field Computer Interface. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'05)*, pp. 1154–1155, 2005.
  - [90] Daniel Vogel and Patrick Baudisch. Shift: A Technique for Operating Pen-Based Interfaces Using Touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'07)*, pp. 657–666, 2007.
  - [91] Manuela Waldner, Jörg Hauber, Jürgen Zauner, Michael Haller, and Mark Billinghurst. Tangible Tiles: Design and Evaluation of a Tangible User Interface in a Collaborative Tabletop Setup. In *Proceedings of the 18th Australia Conference on Computer-Human Interaction: Design: Activities, Artefacts and Environments (OZCHI'06)*, pp. 151–158, 2006.
  - [92] Feng Wang, Xiang Cao, Xiangshi Ren, and Pourang Irani. Detecting and Leveraging Finger Orientation for Interaction with Direct-Touch Surfaces. In *Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology (UIST'09)*, pp. 23–32, 2009.
  - [93] Feng Wang and Xiangshi Ren. Empirical Evaluation for Finger Input Properties In Multi-touch Interaction. In *Proceedings of the 27th International Conference on Human Factors in Computing Systems (CHI'09)*, pp. 1063–1072, 2009.

- [94] Malte Weiss, Julie Wagner, Yvonne Jansen, Roger Jennings, Ramsin Khoshabeh, James D. Hollan, and Jan Borchers. SLAP Widgets: Bridging the Gap Between Virtual and Physical Controls on Tabletops. In *Proceedings of the 27th International Conference on Human Factors in Computing Systems (CHI'09)*, pp. 481–490, 2009.
- [95] Pierre Wellner, 坂村 健(訳). 電脳強化環境—どこでもコンピュータの技術と展望—. パーソナルメディア, 1994.
- [96] Daniel Wigdor, Clifton Forlines, Patrick Baudisch, John Barnwell, and Chia Shen. Lucid-Touch: A See-Through Mobile Device. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology (UIST'07)*, pp. 269–278, 2007.
- [97] Andrew D. Wilson. TouchLight: An Imaging Touch Screen and Display for Gesture-Based Interaction. In *Proceedings of the 6th international conference on Multimodal interfaces (ICMI'04)*, pp. 69–76, 2004.
- [98] Andrew D. Wilson, Shahram Izadi, Otmar Hilliges, Armando Garcia-Mendoza, and David Kirk. Bringing Physics to the Surface. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology (UIST'08)*, pp. 67–76, 2008.
- [99] Mike Wu and Ravin Balakrishnan. Multi-Finger and Whole Hand Gestural Interaction Techniques for Multi-User Tabletop Displays. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology (UIST'03)*, pp. 193–202, 2003.
- [100] Ka-Ping Yee. Peephole Displays: Pen Interaction on Spatially Aware Handheld Computers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'03)*, pp. 1–8, 2003.
- [101] 川端一輝. 認識ペンによる直筆・オンライン署名同時生成方式認証システムの開発. 「IPA創造的ソフトウェア育成事業」最終成果報告発表会論文集, 1998.
- [102] 寺嶋一浩, 岩下貴司, 小室孝, 石川正俊. 携帯機器向け空中キー入力インターフェースのための手指の動作認識アルゴリズム. 第7回計測自動制御学会システムインテグレーション部門講演会 (SI'06), pp. 1378–1379, 2006.
- [103] 竹川佳成, 寺田努, 西尾章治郎. 鍵盤楽器のための実時間運指取得システムの構築. コンピュータソフトウェア, Vol. 23, No. 4, pp. 51–59, 2006.
- [104] 竹岡義樹, 味八木崇, 曙本純一. Z-touch: 指先姿勢インタラクション可能なマルチタッチシステム. 第18回インタラクティブシステムとソフトウェアに関するワークショップ (WISS 2010) 論文集, pp. 71–76, 2010.
- [105] 岩村吉晃. タッチ. 医学書院, 2001.

- [106] 鎌倉矩子. 手のかたち手のうごき. 医歯薬出版, 1989.
- [107] 山本景子, 金谷一朗, 佐藤宏介. ペンの把持情報を利用したペントアプレット型ドローイング支援インタフェース. インタラクション 2010 論文集, pp. 239–242, 2010.
- [108] 福地健太郎, 曜本純一. 人体形状センサのエンターテインメント応用とそのインタラクション技術. エンタテインメントコンピューティング 2003 論文集, pp. 70–74, 2003.
- [109] 岡田謙一, 西田正吾, 葛岡英明, 仲谷美江, 塩澤秀和. ヒューマンコンピュータインタラクション. オーム社, 2002.
- [110] 広田光一, 廣瀬通孝. 五感情報通信の実現にむけて～触覚および嗅覚情報の伝達～. 情報処理学会 ヒューマンインタフェース研究会報告, pp. 13–18, 2002.
- [111] 野中秀俊, 栗原正仁. Tilt-Stick & Spin-Stick: 特定アフォーダンスに基づくペン入力デバイス. インタラクション 2006 論文集, pp. 43–44, 2006.
- [112] 伊賀聰一郎, 伊藤英一, 安村通晃. Kirifuki: 呼気・吸気を利用した GUI 操作環境の提案. 情報処理学会 ヒューマンインタフェース研究会報告, 第 2000-HI-87 卷, pp. 49–54, 2000.
- [113] 佐藤大介, 志築文太郎, 三浦元喜, 田中二郎. Popie: フローメニューに基づく日本語入力手法. 情報処理学会論文誌, Vol. 47, No. 7, pp. 2305–2316, 2006.
- [114] 鳴海拓志, 谷川智洋, 梶波崇, 廣瀬通孝. メタクッキー: 感覚間相互作用を用いた味覚ディスプレイの検討. 日本バーチャルリアリティ学会論文誌, Vol. 14, No. 4, pp. 579–588, 2010.
- [115] 田村博. ヒューマンインタフェース. オーム社, 1998.
- [116] 福本雅朗, 外村佳伸. Wireless FingeRing: 人体を信号経路に用いた常装着型キーボード. 情報処理学会論文誌, Vol. 39, No. 5, pp. 1423–1430, 1998.
- [117] 赤羽歩, 村山淳, 山口武彦, 寺西望, 佐藤誠. 触感提示機能を持つタッチパネルのための押下感生成信号の検討. ヒューマンインタフェース学会論文誌, Vol. 8, No. 4, pp. 591–598, 2006.
- [118] 松井望, 山本喜一. バーチャルキーボード: ビデオ画像からの頑健な実時間指先検出の実現. 第 3 回プログラミングおよび応用のシステムに関するワークショップ (SPA'00) 論文集, 2000.
- [119] 鈴木優, 三末和男, 田中二郎. ペンによる自然な振る舞いを利用したインタラクション手法. 第 14 回インタラクティブシステムとソフトウェアに関するワークショップ (WISS 2006) 論文集, pp. 141–142, 2006.

- [120] 鈴木優, 三末和男, 田中二郎. ユーザに優しいディジタルノート向けスタイルスインターフェース. 第 15 回インタラクティブシステムとソフトウェアに関するワークショップ (WISS 2007) 論文集, pp. 69–74, 2007.
- [121] 鈴木優, 三末和男, 田中二郎. ショートカットアクション: 手の動きを用いたスタイルスインターフェース. 情報処理学会第 70 回全国大会講演論文集, pp. 4:19–4:20, 2008.
- [122] 鈴木優, 三末和男, 田中二郎. ペンを握る動作を用いた離散入力操作の検討. 第 17 回インタラクティブシステムとソフトウェアに関するワークショップ (WISS 2009) 論文集, pp. 7–12, 2009.
- [123] 鈴木優, 三末和男, 田中二郎. ペンを握る力と筆圧を組み合わせたインタラクション手法. 情報処理学会創立 50 周年記念 (第 72 回) 全国大会講演論文集, pp. 4:23–4:24, 2010.
- [124] 鈴木優, 三末和男, 田中二郎. ペンを握る動作を利用したインタラクション手法の検討. 情報処理学会論文誌, Vol. 52, No. 4, 2011 (to appear).
- [125] 三谷雄二. タッチパネル—開発技術の進展—. シーエムシー出版, 2009.
- [126] 黒川隆夫. ノンバーバルインタフェース. オーム社, 1994.
- [127] 館暲, 佐藤誠, 廣瀬通孝. バーチャルリアリティ学. 工業調査会, 2010.

## 著者論文リスト

### 本研究に関連する論文

#### 論文誌

- [1] 鈴木優, 三末和男, 田中二郎. ペンを握る動作を利用したインタラクション手法の検討. 情報処理学会論文誌, Vol. 52, No. 4, 2011 (to appear).
- [2] Yu Suzuki, Kazuo Misue, and Jiro Tanaka. Pen-based Interface Using Hand Motions in the Air. *The IEICE Transactions on Information and Systems*, Vol. E91-D, No. 11, pp. 2647–2654, 2008.

#### 査読付き国際会議論文

- [3] Yu Suzuki, Kazuo Misue, and Jiro Tanaka. Interaction Technique Combining Gripping and Pen Pressures. In *Proceedings of 14th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES'10)*, LNAI 6279, pp. 440–448, 2010.
- [4] Yu Suzuki, Kazuo Misue, and Jiro Tanaka. Interaction Technique for a Pen-Based Interface Using Finger Motions. In *Proceedings of 13th International Conference on Human-Computer Interaction (HCI International 2009)*, LNCS 5611, pp. 503–512, 2009.
- [5] Yu Suzuki, Kazuo Misue, and Jiro Tanaka. Interface for Digital Notes Using Stylus Motions Made in the Air. In *Proceedings of the 2nd International Conference on Knowledge, Information and Creativity Support Systems (KICSS'07)*, pp. 104–109, 2007.
- [6] Yu Suzuki, Kazuo Misue, and Jiro Tanaka. Stylus Enhancement to Enrich Interaction with Computers. In *Proceedings of 12th International Conference on Human-Computer Interaction (HCI International 2007)*, LNCS4551, pp. 133–142, 2007.

#### 査読付き国内会議論文

- [7] 鈴木優, 三末和男, 田中二郎. ペンを握る動作を用いた離散入力操作の検討. 第 17 回インタラクティブシステムとソフトウェアに関するワークショップ (WISS 2009) 論文集, pp. 7–12, 2009.

- [8] 鈴木優, 三末和男, 田中二郎. ユーザに優しいディジタルノート向けスタイルスインターフェース. 第 15 回インタラクティブシステムとソフトウェアに関するワークショップ (WISS 2007) 論文集, pp. 69–74, 2007.

### その他の論文

- [9] 鈴木優, 三末和男, 田中二郎. ペンを握る力と筆圧を組み合わせたインタラクション手法. 情報処理学会 創立 50 周年記念 (第 72 回) 全国大会講演論文集, pp. 4:23–4:24, 2010.
- [10] 鈴木優, 三末和男, 田中二郎. ショートカットアクション : 手の動きを用いたスタイルスインターフェース. 情報処理学会第 70 回全国大会講演論文集, pp. 4:19–4:20, 2008.
- [11] 鈴木優, 三末和男, 田中二郎. ペンによる自然な振る舞いを利用したインタラクション手法. 第 14 回インタラクティブシステムとソフトウェアに関するワークショップ (WISS 2006) 論文集, pp. 141–142, 2006.

## 本研究に関連しない論文

### 査読付き国際会議論文

- [12] Yu Suzuki, Kazuo Misue, and Jiro Tanaka. Visualization Technique for Linked Changes in Exchange Rates Between Three Currencies. In *Proceedings of the 3rd International Conference on Knowledge, Information and Creativity Support Systems (KICSS'08)*, pp. 136–141, 2008.

### その他の論文

- [13] 鈴木優, 島村祐介, 武田嵩太朗, 朴春子, 大和田創, 三末和男. 未来のキッチンプロジェクト “Kitchen of the Future”～調理者の状況に応じた適切な入力インターフェースの実現～. 筑波大学 大学院システム情報工学研究科 コンピュータサイエンス専攻ソリューション型研究開発プロジェクト 2009 年度 研究成果報告, pp. 61–70, 2010.
- [14] 武田嵩太朗, 鈴木優, 島村祐介, 朴春子, 大和田創, 三末和男, 田中二郎. キッチンにおける調理者の状況に適したインターフェース—まな板への情報提示とそのタッチ操作手法の開発—. 情報処理学会 創立 50 周年記念 (第 72 回) 全国大会講演論文集, pp. 3:183–3:184, 2010.
- [15] 島村祐介, 鈴木優, 武田嵩太朗, 朴春子, 大和田創, 三末和男, 田中二郎. キッチンにおける調理者の状況に適した入力インターフェース—調理者の状況分析と入力インターフェースの開発及び評価—. 情報処理学会 創立 50 周年記念 (第 72 回) 全国大会講演論文集, pp. 3:185–3:186, 2010.

- [16] Atia Ayman, Yu Suzuki, Sewang Seo, Jie Gao, Atsutomo Kobayashi, and Jiro Tanaka. Smart Team Management System. CS Technical Reports Special Issue on “Human Resource Development with Practical IT Skills in Advanced Information Science” Program, University of Tsukuba, pp. 19–28, 2008.
- [17] 村上愛淑, 早樋沙織, 鈴木優, 佐藤修治, 三末和男, 田中二郎, 椎尾一郎. 塩味センサによる調味支援. ヒューマンインターフェースシンポジウム 2006 (HIS'06), pp. 659–662, 2006.