

ヒューマンコンピュータ
インタラクションにおける
非明示的挙動から
ユーザ状態を推定するモデルの獲得

谷 堯尚

博士(情報学)

総合研究大学院大学

複合科学研究科

情報学専攻

平成 26 年度

2015 年 3 月

本論文は総合研究大学院大学複合科学研究科情報学専攻に
博士（情報学）授与の要件として提出した博士論文である。

審査委員会

山田 誠二 (主査)	総合研究大学院大学 / 国立情報学研究所 / 東京工業大学
相原 健郎	総合研究大学院大学 / 国立情報学研究所
稲邑 哲也	総合研究大学院大学 / 国立情報学研究所
市瀬 龍太郎	総合研究大学院大学 / 国立情報学研究所
藤田 欣也	東京農工大学

Acquisition of the model estimating
the user state from implicit behaviors
in a human computer interaction

Takahisa Tani

DOCTOR OF
PHILOSOPHY

Department of Informatics

School of Multidisciplinary Science

The Graduate University for Advanced Studies (SOKENDAI)

March, 2015

A dissertation submitted to the Department of Informatics,
School of Multidisciplinary Sciences,
The Graduate University for Advanced Studies (SOKENDAI)
in partial fulfillment of the requirements for
the degree of Doctor of Philosophy

Advisory Committee

Seiji Yamada (Chair)	National Institute of Informatics/ The Graduate University for Advanced Studies/ Tokyo Institute of Technology
Kenro Aihara	National Institute of Informatics/ The Graduate University for Advanced Studies
Tetsunari Inamura	National Institute of Informatics/ The Graduate University for Advanced Studies
Ryutaro Ichise	National Institute of Informatics/ The Graduate University for Advanced Studies
Kinya Fujita	Tokyo University of Agriculture and Technology

内容梗概

ヒューマンコンピュータインタラクションにおいて、人の意図をコンピュータが把握することは、円滑なインタラクションを実現する1つの方法となりうる。一方、人がコンピュータに対して能動的、明示的、あるいは意識的に与えることができる情報には限りがある。労力や時間による制約や、インターフェイスの機能そのものによる制約などが、その主な理由である。一方、ヒューマンヒューマンインタラクションにおいては、ユーザが必ずしも全ての情報を明示的に発しているわけではなく、非明示的情報をユーザが利用する場合や、ユーザが意識しない情報がやりとりされている場合もある。これをヒューマンコンピュータインタラクションにも応用する、すなわち人が無意識に発している情報である非明示的情報をコンピュータが収集、利用することで、より円滑なインタラクションを実現できる可能性がある。

本研究では、新たに有用な非明示的情報を2件提案し、その有用性について考察する。1件目はデスクワーク中のユーザが机上に与える圧力である。パーソナルコンピュータ(PC)を用いたデスクワーク中、ユーザはキータイピングやマウス操作、腕を置くなど、様々な行動により机上に圧力を加えている。一方、PCを用いた作業中、PCはユーザに対し情報通知を行うことがあるが、このタイミングが適切でないと、作業効率が低下することが報告されている。そこで本研究では、デスクワーク中にユーザが机上に与える押圧という非明示的情報を利用し、ユーザの繁閑を推定することを提案し、その有用性を示す。

2件目はタッチパネル操作におけるタップ座標である。タッチパネルは近年様々な機器に用いられる有用なインターフェイス装置であるが、物理キーボードやマウスなどと比べ入力精度が低いことが指摘されている。先行研究ではキネマティック的な入力座標の変化をモデル化し補正することが提案されているが、タッチパネルに表示するボタンなどのオブジェクトUIの形状を人間が認識した時の入力座標の変化については考察されていない。そこで本研究では、UIの形状による人間の行動特性の変化である認知的エラーをモデル化し補正することで、タッチパネルの入力精度のさらなる向

上を目指す．

参加者の協力を得た実験により，両手法ともインタラクションの円滑化に有用な情報を得られることを支持する結果が得られた．これにより，提案した2件の非明示的情報が，コンピュータがユーザの意図を把握するために有用であることを示した．

目次

第1章	序論	1
1.1	本研究の背景と目的	1
1.2	本論文の構成	3
第2章	関連研究	5
2.1	インタラクションの円滑化	5
2.1.1	インタラクションにおけるモデル	5
2.1.2	ヒューマンコンピュータインタラクションにおけるモデル	6
2.1.3	非明示的情報	6
2.2	情報通知	6
2.2.1	ユーザの状態推定	8
2.2.2	情報通知のためのユーザモデル	8
2.2.3	ペリフェラルディスプレイ	9
2.3	タッピングのエラーモデル	9
2.3.1	タップ座標の誤差	10
2.3.2	タップエラーの改善	11
第3章	机にかかる圧力を用いたユーザの割り込み可能性推定	19
3.1	圧力センサによるユーザ状態推定	21
3.1.1	圧力センサ	21
3.1.2	特徴量抽出	22
3.1.3	推定方法	24

3.2	使用キーボード差の影響評価実験	27
3.2.1	実験方法	27
3.2.2	実験結果	34
3.2.3	考察	37
3.3	評価実験	39
3.3.1	実験環境とタスク	39
3.3.2	参加者と実験手順	42
3.3.3	実験結果	43
3.4	考察	45
3.4.1	机上の圧力を用いた状態推定の評価	45
3.4.2	キーボード, マウスの非操作時における状態推定	45
3.4.3	時系列特徴の効果	46
3.4.4	より実用的な特徴選択	48
3.4.5	本研究の適用範囲	51
第4章	タッチパネルにおけるUIデザインを考慮した操作特性モデル	53
4.1	タッチパネル入力の問題点	53
4.2	タップ位置補正による入力精度向上	55
4.2.1	インターフェイス形状によるタップモデルへの影響	55
4.2.2	エラーモデル	56
4.3	ガウス過程回帰	57
4.3.1	多変量ガウス分布	57
4.3.2	ガウス過程	58
4.3.3	ガウス過程の回帰への応用	58
4.3.4	ガウス過程回帰の多次元出力への拡張	59
4.4	評価実験	62
4.4.1	実験方法	62

4.4.2	評価方法	62
4.4.3	実験結果	66
4.5	考察	75
4.5.1	得られたエラーモデル	75
4.5.2	タップ精度	75
第5章	結論	79
	謝辞	81
	参考文献	81
	研究業績	88

目次

2.1	情報通知の例	7
2.2	人間の指のタップ特性 [Holz 11]	15
2.3	目標座標とセンサが出力する座標の差 [Weir 12]	15
2.4	デバイスを保持する姿勢とエラーの関係 [Goel 13]	16
2.5	入力座標の誤差 [Weir 12]	16
2.6	次に押される可能性が高いキーを大きく表示する [Al Faraj 09]	17
2.7	タップモデルを元に設計されたソフトウェアキーボードの例 [Findlater 12]	17
3.1	LL Sensor ¹	22
3.2	LL Sensor の原理図 ¹	22
3.3	LL Sensor の出力	23
3.4	線形しきい素子の分離超平面とマージン (\mathbf{g} がクラス 1 のサンプル, $-\mathbf{g}$ がクラス -1 のサンプルを示す. \mathbf{h} と $-\mathbf{h}$ はサポートベクタを示す.)	26
3.5	実験環境の概観	28
3.6	タスクの画面 (メイン画面)	28
3.7	タスクの画面 (情報通知質問画面)	29
3.8	センサ領域の分割模式図	31
3.9	キーボード a: FK-EA4XA	32
3.10	キーボード b: Realforce 91 NE0100	32
3.11	決定木の例 (参加者 c キーボード b)	35
3.12	決定木の例 (参加者 b)	36
3.13	決定木の例 (キーボード b)	36

3.14	実験に使用したキーボード：SKB-LKG3BK	39
3.15	実験に使用したマウス：LS-1BK	40
3.16	実験環境	41
3.17	実験環境のフットスイッチ	41
3.18	2つのタスクにおけるタスクウィンドウ	42
3.19	情報通知ダイアログ	42
3.20	圧力測定のタイムライン	46
3.21	小型で安価なセンサ	48
3.22	直線状にセンサを配置した場合のシミュレート	49
3.23	直線状にセンサを配置した場合の推定精度	50
4.1	スマートフォンのソフトウェアキーボードの例	54
4.2	タップ対象とタップ位置	55
4.3	実験環境の概観	63
4.4	タスク画面	64
4.5	実験に使用したスマートフォン：Nexus 4	65
4.6	周辺マーカ番号	66
4.7	タップ位置とキネマティックエラーの関係	67
4.8	マーカの表示位置と認知的なエラーの関係（周辺マーカが1つのみの場合）	69
4.9	周辺マーカパターンと認知的なエラーの関係（全ての周辺マーカパターンに対するタップ座標誤差の平均）	70
4.10	周辺マーカパターンと認知的なエラーの関係（全ての周辺マーカパターンに対する回帰結果）	71

表目次

3.1	使用する特徴量	24
3.2	使用する特徴量	30
3.3	センサ領域の分割	30
3.4	データ例	35
3.5	10-fold cross validation 結果	38
3.6	データの例 (左腕)	44
3.7	3つの分類学習アルゴリズムの精度 (下線が各タスクの最良値, 太字が すべてにおける最良値)	45
3.8	時系列データを使用した場合の推定精度 (下線が各学習アルゴリズムの 最良値, 太字はすべてにおける最良値)	47
3.9	局所的にセンサを配置した場合の推定精度	49
4.1	周辺マーカパターンと認知的なエラーの関係	72
4.2	タップの成功率	74

第1章 序論

1.1 本研究の背景と目的

ヒューマンコンピュータインタラクションにおいて，人の意図をコンピュータが把握することは，円滑なインタラクションを実現する1つの方法となりうる．一方，人がコンピュータに対して能動的，明示的，あるいは意識的に与えることができる情報には限りがある．労力や時間による制約や，インタフェースの機能そのものによる制約などがその主な理由である．一方，ヒューマンヒューマンインタラクションにおいては，情報発信者が必ずしも全ての情報を明示的に発しているわけではなく，非明示的な情報を情報受信者が利用する場合や，情報発信者も受信者も意識しない情報がやりとりされている場合もある．

本研究では，情報発信者が明示的，意識的，あるいは能動的に発信する情報，例えば言葉や身振り手振り（ジェスチャー），目配せなどを明示的情報と呼ぶ．これに対し，非明示的あるいは無意識的に発信する情報を非明示的情報と呼ぶ．例えば発音の大小は，ときに表現手法の1つとして意識的に変化をつけられ，明示的情報となりうるが，無意識に変化する場合もある．身振り手振りは意識的に行う場合が多いが，頬を掻くなどの身体操作は無意識に行われる場合が多い．視線は目配せとして意識的に情報を表す場合もあるが，無意識的に視線が変化する場合もある．非明示的情報は，明示的情報には含まれていない情報を含んでいることが多い．なぜなら，非明示的情報はユーザが自分の持つ情報の中から発信する情報として選ばれた情報と言えるからである．これをヒューマンコンピュータインタラクションにも用いる，すなわち非明示的情報をコンピュータが収集，利用することで，ユーザの潜在的な要求をコンピュータがくみ取り，より円滑なインタラクションを実現できると考えられる．

ヒューマンヒューマンインタラクションにおいて明示的 / 非明示的情報の分類は見かけ上難しく、同じ動作がどちらにも属する場合があるが、本研究においては情報発信者の意識の有無により分類するものとする。一方、ヒューマンコンピュータインタラクションにおいては、コンピュータのインタフェースははっきり定義されているため、ユーザはその定義通りの情報を明示的情報として与え、他は非明示的情報となることが期待できる。例えばインタフェースとしてキーボードを用いる場合、押すキーとその順番、そして稀に押す間隔の長さが明示的情報となるが、キーを押す強さや、キーの中の位置（キーの右端を押すか中央を押すかなど）を意識することはほぼ無いと考えられる。

本研究では、新たに有用な非明示的情報を2件提案し、その有用性について考察する。1件目はデスクワーク中のユーザが机上に与える圧力である。パーソナルコンピュータ(PC)を用いたデスクワーク中、ユーザはキータイピングやマウス操作、腕を置くなど、様々な行動により机上に圧力を加えている。一方、PCを用いた作業中、PCはユーザに対し情報通知を行うことがあるが、このタイミングが適切でないと、作業効率が低下することが報告されている。そこで本研究では、デスクワーク中にユーザが机上に与える押圧という非明示的情報を利用し、ユーザの繁閑を推定することを提案し、その有用性を示す。

2件目はタッチパネル操作におけるタップ座標である。タッチパネルは近年様々な機器に用いられる有用なインタフェース装置であるが、物理キーボードやマウスなどと比べ入力精度が低いことが指摘されている。先行研究ではキネマティクスの入力座標の変化をモデル化し補正することが提案されているが、タッチパネルに表示するUIの形状を人間が認識した時の入力座標の変化については考察されていない。そこで本研究では、UIの形状による人間の行動特性の変化をモデル化し補正することで、タッチパネルの入力精度のさらなる向上を目指す。

1.2 本論文の構成

第2章「関連研究」では、本研究の枠組みである非明示的情報とヒューマンコンピュータインタラクションについて述べ、更に本研究で取り上げる2つの課題について取り組む先行研究を説明する。

第3章「机上のにかかる圧力を用いたユーザの割り込み可能性推定」では、本研究で提案する、デスクワーク中の割り込み可能性を机上にかかる圧力から推定する方法について説明する。更に2件の参加者実験を行い、その結果について考察する。

第4章「タッチパネルにおけるUIデザインを考慮した操作特性モデル」では、本研究で提案する、タッチパネルのタップ誤差と表示されるUIの形状の関係について述べ、参加者実験を行い、その結果について考察する。

第5章では本論文の成果をまとめる。

第2章 関連研究

2.1 インタラクションの円滑化

人同士，あるいは人とコンピュータから成る集団があるとき，インタラクションを行う目的として，集団としてのパフォーマンスを向上することがしばしば挙げられる．本研究では，このパフォーマンスをより高くすること，あるいは，高いパフォーマンスへの到達速度を向上することを，インタラクションの円滑化とし，これを目指す．同様の事例として，アンビエントインタフェース [Tripathi 08] が挙げられる．アンビエントインタフェースは，従来のようなユーザが各種周辺機器を自ら調整するシステムではなく，環境がユーザの状況をセンシングし，より適当な状態に移行するシステムである．

2.1.1 インタラクションにおけるモデル

円滑なインタラクションを実現する手段の1つとして，当事者それぞれが相手のモデルを持ち，それに適した言動を選択することが挙げられる．ここで使用されるモデルとしては，メンタルモデルや認知モデルなどが挙げられる．

こうした比較的高次元な思考におけるモデルに対し，現象に対して状態を判断するような，思考を伴わないモデルも利用される．例えば，目線を逸らすことに対して後ろめたさがあると判断する，頬杖をついているのを見てリラックスしていると判断する，などである．多くの場合，これらのモデルによって得られた情報は，より高次元な判断に用いられる．

2.1.2 ヒューマンコンピュータインタラクションにおけるモデル

ヒューマンコンピュータインタラクション (HCI : Human-Computer Interaction) においては、しばしば人間同士のインタラクションで用いられるモデルが応用される。前述したモデルのうち比較的次元な判断としてのモデルの応用、あるいは代替として、例えばセンサー値に対して分類学習やクラスタリングを用いてラベル付けを行うことや、回帰によって出力値を予想するなどの手法がしばしば用いられる。本研究ではこのような、分類学習あるいは回帰を用いた写像の同定を対象とし、モデル化と呼ぶことにする。

2.1.3 非明示的情報

インタラクションが行われることは、情報がやりとりされることであると言える。この情報は、明示的情報と非明示的情報とに分類できる。

明示的情報とは、情報発信者が情報受信者に対して提示する意思を持って提示する情報を指す。例えば会話における会話の内容や、キーボード入力における入力されたキーは明示的情報である。一方、非明示的情報とは、情報発信者が情報受信者に対して提示する意思がないが発信される情報を指す。会話における会話の速度や、キーボード入力におけるタイピングの速度は、非明示的情報である場合がある。会話における速度は抑揚として情報発信者が明示的に発する情報にもなりうるなど、現象だけを指して非明示的情報であると断定することはできない。

非明示的情報の代表的な例として、視線情報が挙げられる。Klamiら [Klami 08] は、視線情報を画像の関連付けに応用することを提案している。

2.2 情報通知

近年の情報化社会において、ユーザは図 2.1 のような、多くの情報通知 (notification) を受け取りながら仕事や生活をしている [Iqbal 10]。その例として、メールやインスタントメッセージなどの着信、様々なアプリケーションの Update などのアラートが挙げ

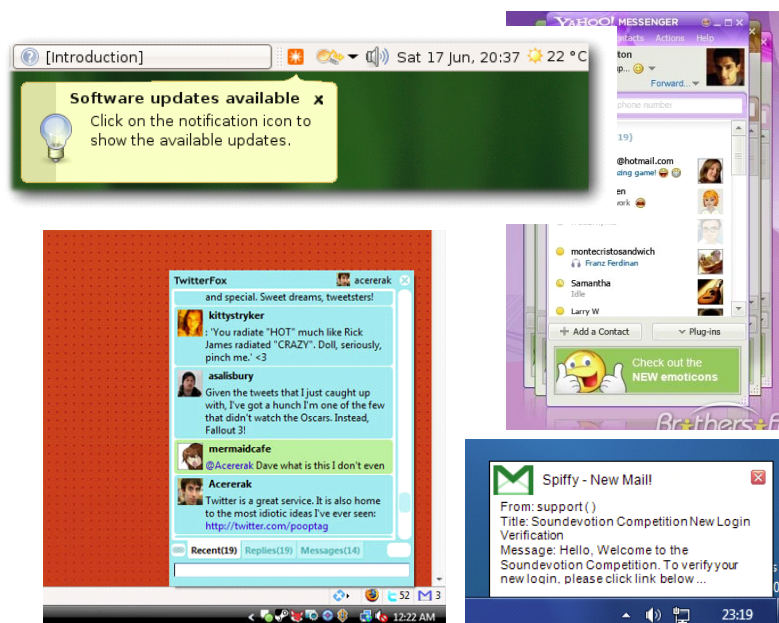


図 2.1: 情報通知の例

られる。これらの情報通知は、ユーザが能動的に情報を入手するのではなく、システムがユーザに対して情報を提示する形式を取る。

ここで生じる問題として、情報通知タイミングの妥当性が挙げられる。情報通知のタイミングが適切でなければ、ユーザに不要なストレスを与えることになるのみならず、知的生産性が低下する可能性が指摘されている [Bailey 01]。これは主に、人間のマルチタスクにおいて、タスクの切り替えに時間が掛かることに起因すると言われてしている。人間は、言語理解や数値処理など、全く異なる種類のタスクを並列に処理することが可能だが、同種のタスクを同時に行うことは困難である。更に同種のタスクを連続して行う場合、タスクの切り替えに数秒から数十秒を要する。そのため、切り替えに必要な時間の分、作業効率が低下する。

本研究では、この問題について解決することで、作業の効率化を図り、インタラクションの円滑化を目指す。

2.2.1 ユーザの状態推定

前述の問題を解決する手法の一つとして、ユーザ状態が情報通知に適したタイミングで情報通知を行うことが考えられる。そのために、システムがユーザのモデルを持ち、それをを用いて現在ユーザがどのような状態であるか推定する。

キーやマウスの入力の頻度やタイミングを用いてユーザの繁閑や感情の状態を推定する手法が提案されている [Epp 11, 水口 04, 本田 98]。しかしこの手法では、入力の頻度が必ずしもユーザの繁閑を反映していない場合に情報提示タイミングの推定ができないことや、ユーザが入力装置を利用していない場合には推定できないなどの問題がある。

また、PC 作業の切れ目度合いを推定する手法 [Iqbal 08] や、アプリケーションの切り替えタイミングを用いる手法 [田中 10] も研究されている。しかし、これらの方法は web 閲覧等のアプリケーションを切り替えない作業には適用できない。

他に、各種センサを利用したユーザの作業の種類に基づく忙しさの推定に関する研究や Web カメラを用いてユーザの表情や姿勢を撮影し推定に用いる研究があるが [Fogarty 05, Chen 07, Jaimes 06]、顔の撮影やセンサをユーザに取り付けることは、観測されていることに対するユーザの心理的負荷や、センサ自体の煩わしさなどが問題になる。

2.2.2 情報通知のためのユーザモデル

情報通知を適切なタイミングで行う際に必要な情報は、ユーザの状態、ユーザの置かれた環境の状態、通知する情報の性質、あるいは重要性などが挙げられる。

例えば、地震速報のような緊急性の高い情報を通知する場合、ユーザ状態に関わらず即時通知することが適切であるが、翌週の予定に関するメールのような緊急性の低い情報を通知する場合、ユーザが作業に集中している場合や、他者との会話中である場合に通知すると、それらのメインタスクを不必要に阻害し、ユーザにストレスを与えたり、作業効率を低下させたりする。

本研究では、通知する情報の緊急性が低く、ユーザの状態やユーザの置かれた環境の状態を考慮して情報通知する必要がある場合を想定する。また、ユーザは1人でデスクワークを行い、他者の介入はない環境を想定する。

2.2.3 ペリフェラルディスプレイ

ユーザモデルレスにこの問題を解決する手法の一つとして、ペリフェラルディスプレイが挙げられる。人間の視野は、集中度が高まれば狭く、下がれば広くなることが知られており、その差の領域を周辺視、または周辺視野などと呼ぶ [McCrickard 01]。周辺視野はユーザが集中している時には視野に入らず、集中していない時に視野に入る領域である。これを利用し、周辺視野内にディスプレイなどを配置することで、情報通知を表示していても、集中時には気付かず、集中が途切れた時に初めて情報通知に気づく、というシステムを構築することができる。このシステムをペリフェラルディスプレイと呼ぶ [Kim 10]。ペリフェラルディスプレイでは、ユーザの状態を推定する必要がない、すなわちシステムがユーザのモデルを持つことなく、適切なタイミングでユーザが情報を受け取ることを可能にしている。

2.3 タッピングのエラーモデル

タッチパネルに対する操作、すなわち、タッチパネルを入力装置として使用する場合を考える。近年のタッチパネルに対する主な操作は、タップ、スワイプ、ピンチアウト/イン、フリックなどに分類される。本研究ではこのうち最も基本的かつ単純なタップについて考える。

タッチパネルをタップする際のエラーとして、タップ座標の誤差、タップしたが反応しない、タップしていない(したつもりがない)のに反応する、の3種類のエラーがしばしば挙げられる。アンケート(10名)の結果、この3種類の中ではタップ座標の誤差が最も頻度が高いと全員が答えた。そこで本研究では、タップ座標の誤差について考える。

2.3.1 タップ座標の誤差

システムが判定するタップ座標とユーザの意図したタップ座標に差があることが多い。この差をタッピングのエラーと呼ぶ。エラーの発生する要因として、タッチパネルの厚みや人間のタップ特性などが挙げられている。

タッチパネルの構造によるエラー

タッチパネルは発光素子の表面にガラスが配置される構造になっている。通常このパネルを用いる場合、パネルに対して垂直に視認することはあまりない。このとき、表示されたタップ対象とタップ位置に差が生じる。通常この差はタッチパネルのキャリブレーションにより大部分が吸収されるため、現在の実用において問題になることは稀である。

人間のタップ特性によるエラー

タッチパネルをタップする場合、自身の指でタップする他、タッチペンなどの補助装置を用いる場合もあるが、補助装置を持ち歩く煩わしさや、ピンチアウトのような複数点を同時にタッチする操作との互換性などに問題が生じるため、自身の指でタップする需要は高い。

一方、人間の指はタッチペンのような様な形状ではないため、その接触面はいくつかの要因により非常に複雑に変化する。Holzら [Holz 11] はこのタップ特性について詳細に調査している。図 2.2 に、Holzらの提案するエラーの概念図を示す。これによると、タップしようとする座標 (a) に対し、指 (b) が触れる位置 (c) は操作する手の根本方向に差が生じ、タップ座標 (d) も同様に差が生じるというものであり、実験によってこれを示している。

Weirら [Weir 12] もどのような報告をしている。図 2.3 に、この論文に示されたタップの例を示す。白黒の濃淡はタッチパネルのセンサである静電容量センサの値で、黒い領域は非反応領域すなわち指が触れていない領域、白い領域は反応領域すなわち指

が触れている領域を示している。更に、白丸はタップしようとするオブジェクトが表示されている座標、黒丸はシステムがタップされたとして出力した座標を示している。

要因としては、デバイスを保持する姿勢、すなわちデバイスを横長に保持するか縦長に保持するか、左右どちらの手で保持するか、左右どちらの手で操作するか、などの他、タッチパネルに対する接触角度や指の状態、すなわち爪の長さや乾湿状況などが挙げられる。

図 2.4 に、[Goel 13] に示される、デバイスを保持する姿勢によるタップ座標の誤差の変化を示す。この研究は様々な保持姿勢でソフトウェアキーボードを操作する際のエラーを補正するものである。図にはそれぞれ (A) 左手親指 (B) 右手親指 (C) 人差し指 (D) 両手親指で操作した場合の、各キーに対するタップの平均座標を黄色点で示している。

これらのエラーを考慮してシステムが得るタップ座標を正確に把握し正確な座標をタップするユーザは皆無と言える。言い換えると、現行のタッチパネルシステムは、これら人間のタップ特性を反映していない。

2.3.2 タップエラーの改善

前述のようなタップ座標の誤差に対して、システムを改良することにより対処する研究が行われている。

対処の方法は、おおまかに 2 種類に分類できる。1 つは入力された内容に対し、モデルを用いて補正・修正を行う方法である。もう 1 つはインタフェースの形状を変化させ、エラーの発生率を抑制する、あるいはエラーによる影響を抑制する方法である。また、それらの方法に用いられるモデルも 2 種類に分類でき、言語モデルを用いる場合と、タップ座標モデルを用いる場合がある。

言語モデルを用いた対処

タッチパネル上に表示されるソフトウェアキーボードを用いた文字入力をする場合を考える。

このとき、タップ座標の誤差は、入力文字の間違いとして顕在化する。入力文字の間違いの問題として考えるとき、本来入力されるべき文字列は、各言語の辞書に存在する単語、あるいは各言語の文法に則った単語列である確率が高いと考えられる。

例えば、直前に入力された文字を用いてマルコフ過程により次に入力される文字を予測することができる。

この手法は、言語の持つ誤り訂正能力を利用していると言える。一方、言語モデルをタップ系列のモデルに拡張することで適用範囲を広げることができると考えられるが、タップが何らかの系列でなければ適用できないという問題がある。

タップモデルを用いた対処

前述のように、タッチパネルに対するタップの誤差は様々な影響により生じる。その誤差をモデル化、すなわちタップモデルを構築する。

タップが系列である必要はないが、タップ座標の誤差がどのような要因の影響を受けているか検討する必要がある。

入力された内容に対する補正

入力された内容に対し、本来入力される確率が高い入力候補を提示、あるいは置き換える。多くの場合、本来入力される候補に対し、入力された内容を条件とする条件付き確率を定義し、確率の高い候補を選択する。

以下に、この手法を適用するときの例を挙げる。ただし、理解を助けるために非常に単純化した例である。

この補正に言語モデルを適用する場合、例えば、英語入力中に”tes”と入力されたとき、その次にタップされる確率が高い文字として”t”などが挙げられる。実際の入力で”y”と入力された場合、“tes”の次に“y”が入力される確率は低いため、入力候補と

して“t”と“y”(あるいは“test”と“tesy”)を表示しユーザに選択させる,あるいは“t”が入力されたものとして扱うことができる.

この補正にタップモデルを適用する場合,例えば,画面上の座標 (x, y) がタップされ,かつ左手で縦長にデバイスを保持し右手の人差し指で操作している場合,タップ座標は (x_d, y_d) 程度の誤差が生じることが事前にわかっているならば,座標 $(x - x_d, y - y_d)$ がタップされたものとして扱うことができる.あるいは,表示されるタップ対象に対し,タップ判定の領域を表示領域とは別に設定する.

図 2.5 に, [Weir 12] で示された入力座標の誤差の図を示す. 実験はスマートフォンを両手で横長に保持した状態で両手親指で操作することを前提として行われた. 提案手法はタッチパネルの入力装置である静電容量センサ値を入力, タップ座標を出力とする関数を GPR により獲得し, この関数によって得られたタップ座標が, 既存の手法より正確な座標であることを示している. 図において, 長方形はタッチパネルを表しており, 実験に使用されたスマートフォン (Nokia N9) に実装された OS (MeeGo) が出力した座標と提案手法が出力した座標の差のうち, 横方向の差を白黒の濃淡で表している. 白い領域は既存手法が左にシフトしている領域, 黒い領域は既存手法が右にシフトしている領域である. 画面左半面は左手親指で操作しているため, 操作する指の方向である左方向に, 画面右反面は右手親指で操作しているため, 操作する指の方向である右方向に, それぞれシフトしている様子が見て取れる. 更に, 画面中央付近は操作する指から遠いため, 白黒どちらとも濃く, すなわち大きくシフトしている. この研究では最終的に, 直径 2mm の円形のターゲットに対するタップタスクにおいて, ターゲットをタップする成功率が 23.47% 向上したとしている.

2.3.2.1 表示するインタフェースの設計によるエラーの抑制

タッチパネルにおいて, タップをする場合は常になんらかのインタフェースを表示した上で, そのインタフェースに対してタップすることになる. すなわち, タップ座標の分布はインタフェース形状により制御されることになる. このとき, タップ座標と表示されたインタフェース中のタップ対象の座標は多くの場合一致せず, またそれ

らの座標の差が全てのタップ対象に対して一樣になることも少ない。

そこで、タップされる確率が高い座標同士がお互いに十分な距離を置けるよう、表示するインタフェースの形状を変化させることが有効となる。

適用例としては、この手法に言語モデルを適用し、ソフトウェアキーボードにおいて、次に入力される確率が高い文字のキーを大きく表示する。図 2.6 に [Al Faraj 09] が提案する、”BigKey”の概観を示す。この例では、現在”t”が入力されている状況(画面中段)において、システムは次に”the”が入力されることを予想し、”h”や”e”のキーを大きく表示している。

あるいは、タップモデルを適用し、タップ座標の誤差が小さくなるようにキーを配置する。図 2.7 に、[Findlater 12] が提案する”Personalized Input”の例を示す。この研究では、十分大きなタッチパネル上を両手全ての指で操作することを前提とし、ソフトウェアキーボードを用いて入力する際の誤入力が最小になるよう、各ユーザ毎にキーボードを設計するというものである。

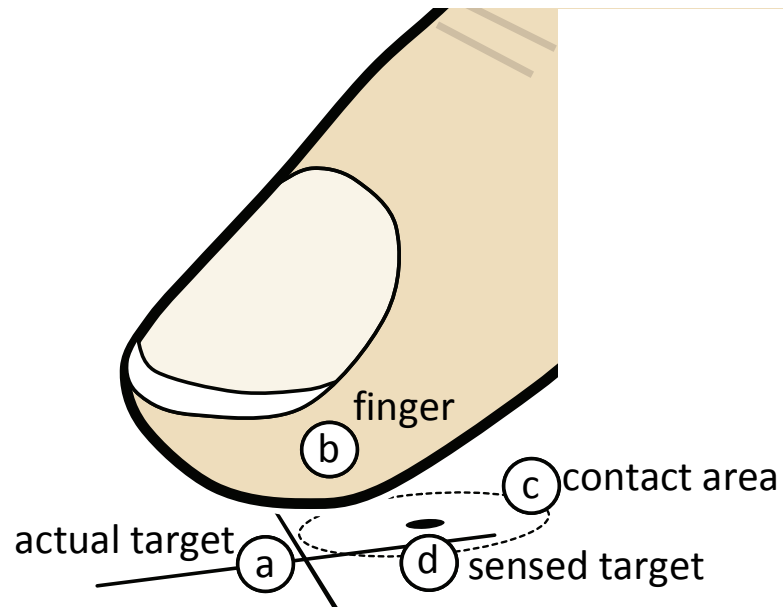


図 2.2: 人間の指のタップ特性 [Holz 11]

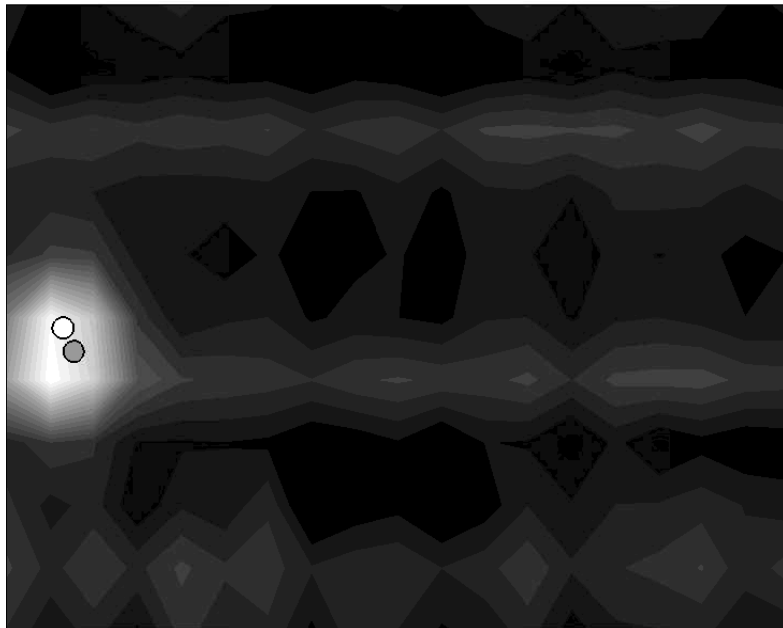


図 2.3: 目標座標とセンサが出力する座標の差 [Weir 12]

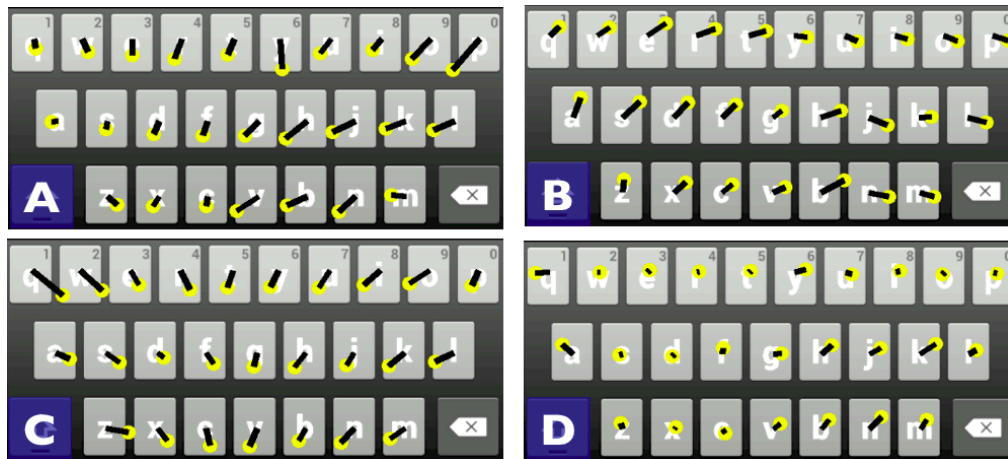


図 2.4: デバイスを保持する姿勢とエラーの関係 [Goel 13]

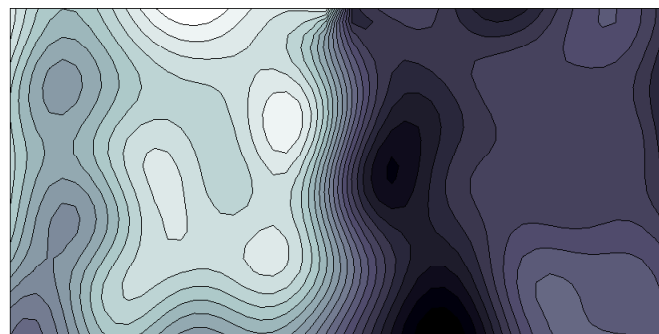


図 2.5: 入力座標の誤差 [Weir 12]



図 2.6: 次に押される可能性が高いキーを大きく表示する [Al Faraj 09]

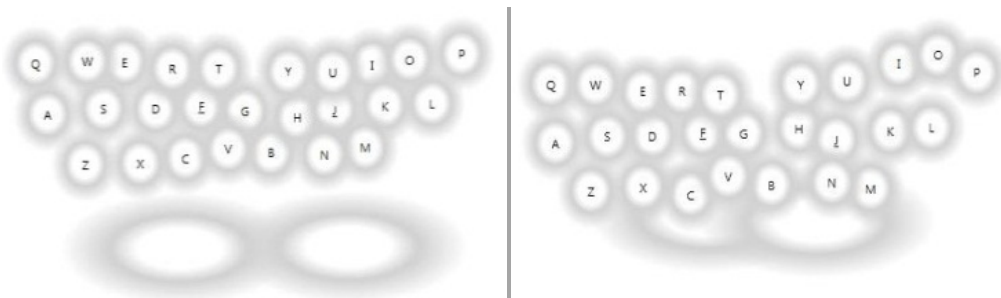


図 2.7: タップモデルを元に設計されたソフトウェアキーボードの例 [Findlater 12]

第3章 机上にかかる圧力を用いたユーザの割り込み可能性推定

近年の情報化社会において、ユーザは多くの情報通知 (notification) を受け取りながら仕事や生活をしている [Iqbal 10]。その例として、メールやインスタントメッセージなどの着信、様々なアプリケーションの Update などのアラートが挙げられる。これらの情報通知は、ユーザが能動的に情報を入手するのではなく、システムがユーザに対して情報を提示する形式を取る。ここで生じる問題として、情報通知タイミングの妥当性が挙げられる。情報通知のタイミングが適切でなければ、ユーザに不要なストレスを与えることになるのみならず、知的生産性が低下する可能性が指摘されている [Bailey 01]。そこで、ユーザの状態を考慮した情報通知の制御が必要となる。つまり、ユーザが情報通知を受理できる割り込み可能状態 (interractable state) であるかどうかを推定し、その状態のときに情報を通知することが望ましい。

また一方、ユーザの割り込み可能状態を推定しないアプローチもある。メインタスクのウィンドウやディスプレイの横にサブウィンドウやサブディスプレイをメインタスクの遂行に干渉しないように配置して、そこに情報通知内容を表示するペリフェラルディスプレイ [McCrickard 01, Kim 10] がそのようなアプローチの典型であるが、本研究では、情報通知以外の目的にも適用することを想定して、ユーザ状態を推定する方法を採用する。

ユーザ状態推定のためにはユーザを観測することが必要である。キーやマウスの入力の頻度やタイミングを用いてユーザの繁閑や感情の状態を推定する手法が提案されている [Epp 11, 水口 04, 本田 98]。しかしこの手法では、入力の頻度が必ずしもユーザの繁閑を反映していない場合に情報提示タイミングの推定ができないことや、ユーザ

が入力装置を利用していない場合には推定できないなどの問題がある。

また、PC作業の切れ目度合いを推定する手法 [Iqbal 08] や、アプリケーションの切り替えタイミングを用いる手法 [田中 10] も研究されている。しかし、これらの方法は web 閲覧等のアプリケーションを切り替えない作業には適用できない。

他に、各種センサを利用したユーザの作業の種類に基づく忙しさの推定に関する研究や Web カメラを用いてユーザの表情や姿勢を撮影し推定に用いる研究があるが [Fogarty 05, Chen 07, Jaimes 06]、顔の撮影やセンサをユーザに取り付けることは、観測されていることに対するユーザの心理的負荷や、センサ自体の煩わしさなどが問題になる。

そこで本研究では、机上にかかる圧力を利用した新しいユーザ状態推定手法を提案する。PC を用いる作業において、机上にはキー入力の押圧や腕を乗せた際の力、マグカップの重さなど様々な圧力が加わっている。また、見せられた文章の内容が肯定的か否定的かによって、無意識のうちに握力の強弱に影響を与えるという研究結果がある [Aravena 12]。これらのことは人間の心理状態が筋力の強弱として表出する可能性を示唆している。そのため、それらの大きさと変化、すなわちキー押圧の強弱、腕に体重を乗せる度合いや位置、もしくはマグカップの中身の減り方などには、ユーザ状態を推定するために有用な情報が含まれていると考えられる。例えば、人は熱中すれば自然と力が入り、キー押圧も普段より強い力で行ったり、飲み物を口にする頻度が下がったりすることが考えられる。作業に集中している間は前傾姿勢になるという報告 [木村 11] から、集中時は非集中時に比べ、腕に体重を多くかけていることも考えられる。

また、机上にかかる圧力の測定はユーザにセンサを取り付ける必要がなく、顔撮影のように個人情報扱うものでもないため、ユーザに物理的、心理的負荷を与えることなく測定することができる。

本研究は、机上にかかる圧力によって、割り込み可能か否かというユーザ状態の推定を実現することを目的とする。従来研究には、机上にかかる圧力によってユーザの割り込み可能性を推定する研究は見当たらないため、新たなユーザ状態の推定方法を開

発するという意味において，本研究は方法論的な新規性をもつと考える．そして，本研究は，ユーザに物理的・心理的な負担を与えない，新しいユーザ状態推定法を提案することにより，ユーザ状態に基づく情報通知の実現に貢献できる価値をもつ．

3.1 圧力センサによるユーザ状態推定

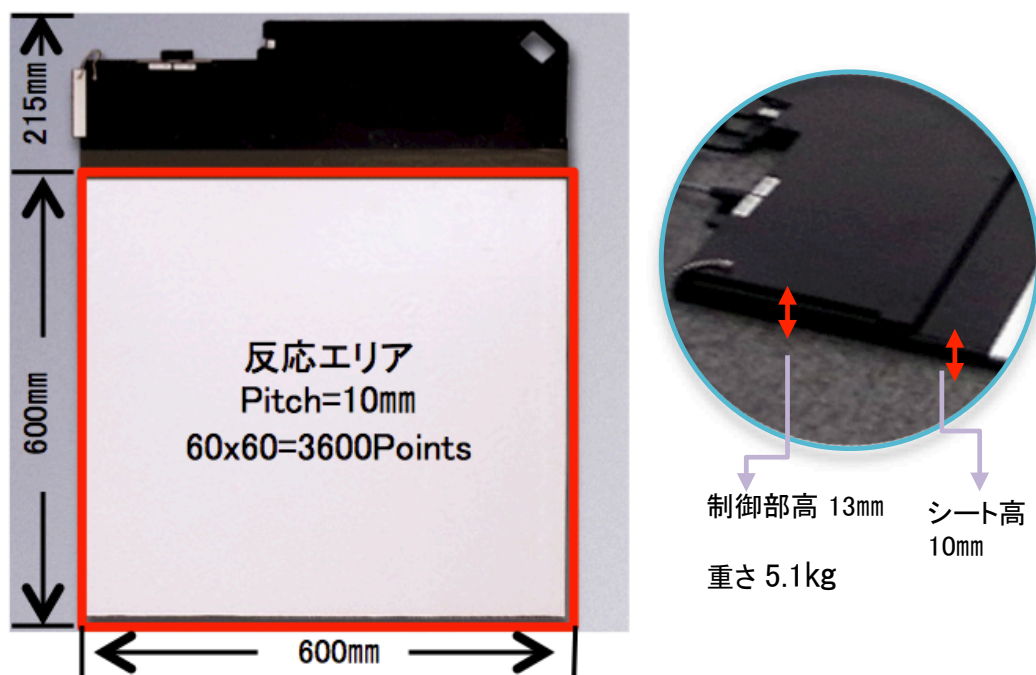
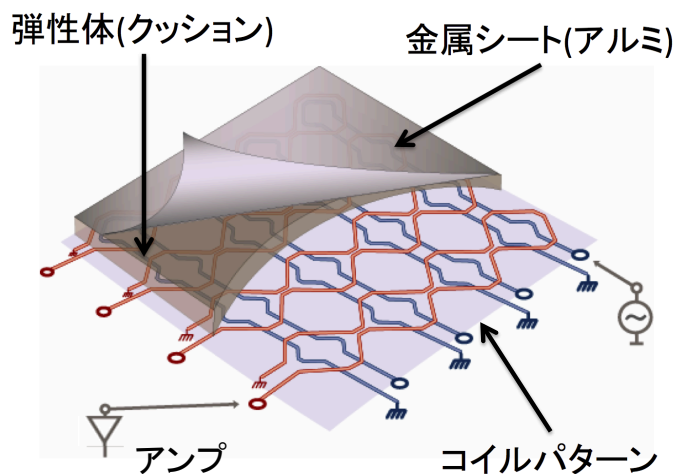
3.1.1 圧力センサ

机にかかる圧力を測定するため，机上に格子状に測定点を持つ圧力センサシートを敷く．これにより机の上の平面上の圧力分布を測定する．キーボード等机上に置く物はすべてこのセンサシート上に置き，作業もシート上で行う．実際に2種類のキーボード(KFK-EA4XA(株)ミツミ電機，Realforce 91 NE0100(株)東プレ)を用いて，その位置を日常的に妥当な範囲ですらしながら使用時の圧力を測ってみるという予備調査を行った．その結果，主にキーボードのタイピング圧力を計測するためには，センサシートの大きさは1m四方程度，圧力分解能は10g程度が必要であることがわかった．

これらの分解能を満たす圧力センサシートとして，本研究では図3.1に示すLL Sensor((株)シロク，大きさ：600mm×600mm)を利用した．一般的に知られる圧力センサには感圧抵抗方式や静電容量方式などがあるが，LL Sensorは電磁誘導を利用した圧力センサシートで，静電式や抵抗式と比較して大面積のセンサを低価格に実現できる特徴がある．また，シート自体はごく薄いもの(厚さ10mm)であり，その上での作業に違和感はない．図3.2にLL Sensorの原理図を示す．上方から圧力が加わることで弾性体の厚みが変わり，金属シートとコイルパターンとの距離が短くなる．これにより行方向コイルと列方向コイルとの間の相互誘導が変化し，圧力変化を電気信号として取り出すことができる．

また，圧力分解能は使用する弾性材により調整可能であり，既にキー押圧の強弱検知が可能なように調整済みである．センサ出力の例を図3.3に示す．ただし，センサの出力は物理量ではなくセンサ独自の尺度である．この図では，出力値の低い方から

¹<http://www.lsensor.com/>

図 3.1: LL Sensor¹図 3.2: LL Sensor の原理図¹

順に白(指示値 0~5), 青(5~10), 緑(10~15), 赤(15~20), 黒(20~)に着色してある。この例ではセンサシート上にキーボードとマウスを配置しタイピングを行なっている。楕円で示した部分に腕が, 四角で示した部分にキーボードが, その右側にマウ

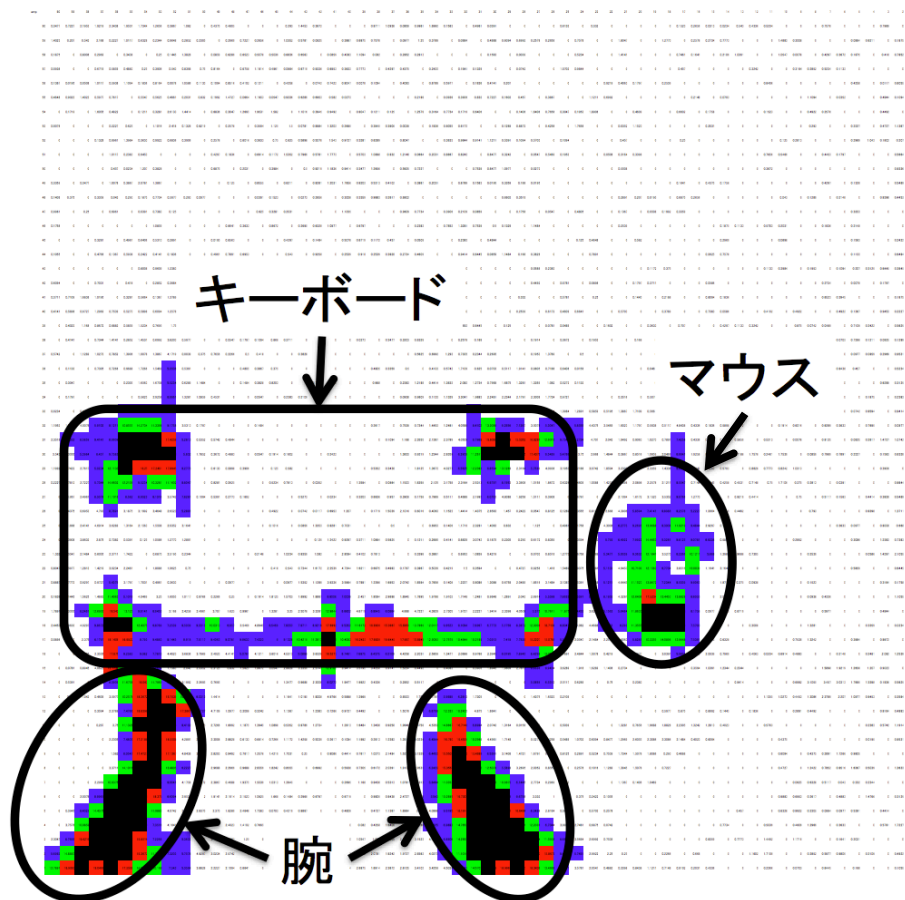


図 3.3: LL Sensor の出力

スが置かれていることが見て取れる。

3.1.2 特徴量抽出

次に、圧力データから特徴量を抽出する。特徴量は、前述のようなキー押圧や乗せている物体の重さ、位置、それらの変化などである。

本研究では、机上に圧力を与える物体はキーボードとマウス、ユーザの腕のみであるとし、モニタは圧力センサのゼロ点補正により影響を排除した。ただし、将来的には本やマグカップなど、多くの物体が置かれた環境に適用できると考えられる。

後述する実験ではタイピングとマウス操作のそれぞれについて典型的なタスクを使用するが、PC を用いた一般的なオフィスワークでは、マウスの利用が必須であり、マ

表 3.1: 使用する特徴量

		圧力	面積	重心				
				x 軸	y 軸			
キーボード	左奥足	24 特徴量	...					
	右奥足							
	手前足							
マウス								
ユーザ	左手					...		
	右手							

ウスの圧力の特徴はユーザ状態推定にとって重要と考えられる。さらに、腕から得られる情報は重要であると考えられる。

以上のような考察から、本研究では表 3.1 のように、以下の 24 個の値を特徴量として使用する。キーボードの左奥足・右奥足・手前足、マウス、ユーザの左腕・右腕の計 6 個の圧力それぞれに対し、圧力値 (Pressure)、面積 (Area)、重心の x 座標 (CoG_x)、重心の y 座標 (CoG_y) の 4 個のパラメータを抽出した。

元の圧力データから特徴を抽出するために、単純なパターンマッチングアルゴリズムを使用した。事前にキーボードとマウスそれぞれの圧力マップを取得、正規化し、マッチングのパターンとした。そして、パターンを用いて圧力データを走査することで、パターンに対応する領域を抽出した。また、左右の腕の位置を得るために、データ全体を右下及び左下から走査し、しきい値を使用して領域を抽出した。なお、領域抽出の精度を一部のデータを用いて調査したところ、約 90 % 程度であった。その後、各領域に対し、圧力、面積、重心を算出する。

圧力データは 30ms 毎に 1 フレーム得られる。また、特徴は割り込みの 1 秒前から 9 フレーム分のデータを平均して使用する。

3.1.3 推定方法

特徴抽出後、得られた特徴をベクトルとして、分類学習アルゴリズムに入力して、分類学習を実行し、割り込み可能か否かを分類できるクラシファイアを学習により得る。

実稼働時には、ユーザによる机上の圧力をモニターしながら、事前に得られたクラシファイアを適用することで、割り込み可能か否かの判定が可能となる。

実験では、高精度のアルゴリズムとして SVM(Support Vector Machines)[Vapnik 95] と Random Forests[Breiman 01] の 2 つと、従来からよく用いられる C4.5[Quinlan 93] を使用し、それらのパフォーマンスを比較検討することにした。

Support Vector Machines

SVM は、教師あり学習を用いるパターン認識モデルの 1 つで、線形しきい素子を用いて 2 クラスのパターン識別器を構成する手法である。訓練データ集合から、「マージン最大化」という基準で線形しきい素子のパラメータを学習する。線形しきい素子は、入力特徴ベクトルに対し、識別関数

$$y = \text{sign}(w^T x - h) \quad (3.1)$$

により 2 値の出力値を計算する。ここで、 w はシナプス荷重に対応するパラメータであり、 h はしきい値である。また関数 $\text{sign}(u)$ は、 $u > 0$ のとき 1 をとり、 $u \leq 0$ のとき -1 をとる符号関数である。このモデルは、入力ベクトルとシナプス荷重の内積がしきい値を超えれば 1 を出力し、超えなければ -1 を出力する。これは幾何学的には、判別超平面により入力特徴空間を 2 つに分けることに相当する。図 3.4 に、この幾何学的解釈の模式図を示す。ここで、2 本の点線 ($H1$, $H2$) により訓練データが完全に分離されており、その中間にある実線が判別超平面である。SVM では、判別超平面と他の 2 つの超平面との距離 (マージン) が最大となるように、これらの超平面を選択する。

この手法では、本質的に非線形で複雑な識別課題に対して良い結果が得られない可能性がある。そこで特徴ベクトルを非線形変換し、その空間内で線形の識別を行う、カーネルトリックと呼ばれる手法を用いることで、非線形な識別課題に対応することができる。

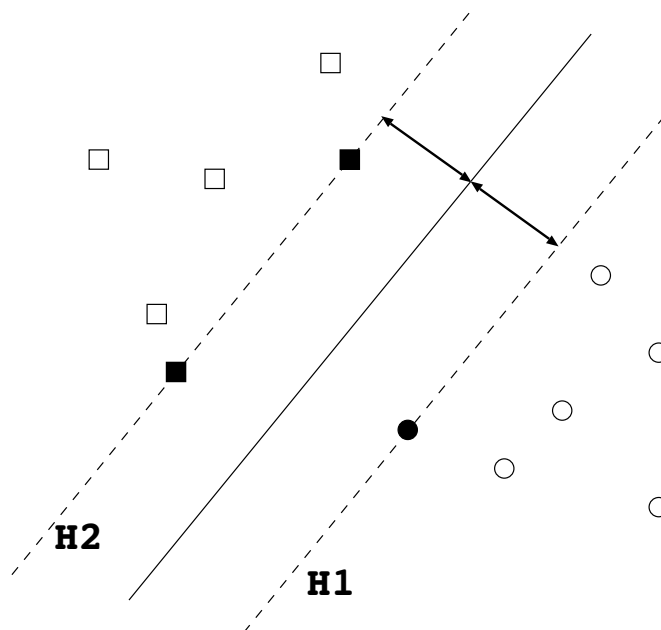


図 3.4: 線形しきい素子の分離超平面とマージン(\square がクラス 1 のサンプル, \circ がクラス-1 のサンプルを示す. \blacksquare と \bullet はサポートベクタを示す.)

Random Forests

Random Forests は、機械学習アルゴリズムの 1 つで、識別、回帰、クラスタリングに用いられる。決定木を弱学習器とする集団学習アルゴリズムであり、この名称は、ランダムサンプリングされたトレーニングデータによって学習した多数の決定木を使用することによる。特徴量の重要度が学習とともに計算できる、学習が高速である、過学習が起きにくいなどの特徴がある。

集団学習とは、モデルを複数生成し、結果を統合することにより、精度を向上させるアルゴリズムである。このとき、各モデルは異なるサンプルや異なる重みを用いることで、異なるモデルとなる。Random Forests を分類学習に用いる場合、結果の統合には多数決が用いられる。

C4.5

C4.5 は ID3[Quinlan 86] を拡張した，決定木による分類アルゴリズムの 1 つである．決定木による学習において，データに内蔵する本質をつかんだ決定木が得られることが理想であるが，有限のデータからそれを実現することは困難である．データに矛盾がない場合，正しい決定木は多数あるが，そのうち簡潔なものほど良いとされることが多い．このような性質を持った決定木を得るために，頻度情報を利用して属性の重要度を推定する．

C をクラスの数，データセット D のうち j 番目のクラスに属するデータ割合を $p(D, j)$ とする．クラスを同定するために必要な平均情報量 $Info(D)$ は次式で与えられる．

$$Info(D) = - \sum_{j=1}^C p(D, j) \times \log_2(p(D, j)) \quad (3.2)$$

k 番目の分類 T における平均情報量の利得 $Gain(D, T)$ は次式で与えられる．

$$Gain(D, T) = Info(D) - \sum_{i=1}^k \frac{|D_i|}{|D|} \times Info(D_i) \quad (3.3)$$

分類に必要な情報量 $Split(D, T)$ は次式で与えられる．

$$Split(D, T) = - \sum_{i=1}^k \frac{|D_i|}{|D|} \times \log_2 \left(\frac{|D_i|}{|D|} \right) \quad (3.4)$$

$Gain(D, T)$ を $Split(D, T)$ で正規化した値を利得比と呼び，これが最大となる T を選択し，決定木を展開していく．

学習が進むと分類による情報量の利得は減少していく．この利得が非常に小さい分類は過学習 (over-fitting) を生じ，学習の汎化性能を低下させる．そこで一定以下の利得しか得られない分類に対して枝刈り (pruning) を行う．

3.2 使用キーボード差の影響評価実験

机上にかかる圧力を用いてユーザ状態を推定する上で重要になると予想される，使用するキーボードの変化の影響を調査するため，実験を行う．



図 3.5: 実験環境の概観

3.2.1 実験方法

実験環境

本研究ではデスクワーク環境を想定するが、本実験では研究の初期段階として、複雑な要因を排除した実験を行う。そのため、単純化したデスクワーク環境を構築した。図 3.16 に実験の様子を示す。

具体的には、PC を用いたタイピングタスクを用意し、そのタスクを行なっている間にシステムからの情報通知を許容するか否かを問う。タスクは、画面上に課題となる文字列がスクロール表示され、その読み仮名に従ってタイピングするものである。スクロールは右から左に向かって文字列が連続的に流れるものとし、タイピングした内容は表示しないものとした。これは仮名と漢字の違いによってタスクの負荷が変動しないようにするためである。図 3.6 にタイピングタスクの画面を示す。画面サイズは $500[\text{pixel}] \times 300[\text{pixel}]$ 、フォントサイズは $30[\text{pixel}]$ である。スクロール速度は (a) $0[\text{pixel/s}]$ (停止)、(b) $33[\text{pixel/s}]$ ($30[\text{ms/pixel}]$)、(c) $50[\text{pixel/s}]$ ($20[\text{ms/pixel}]$)、(d) 100

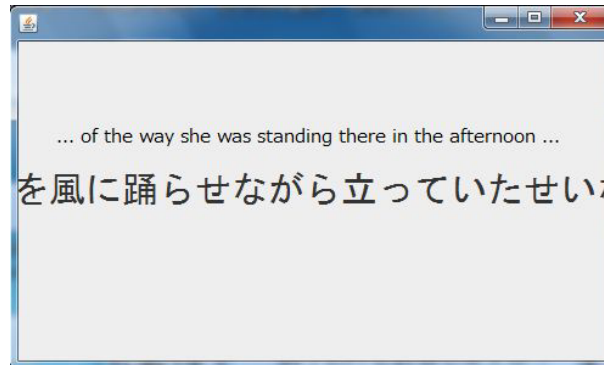


図 3.6: タスクの画面 (メイン画面)

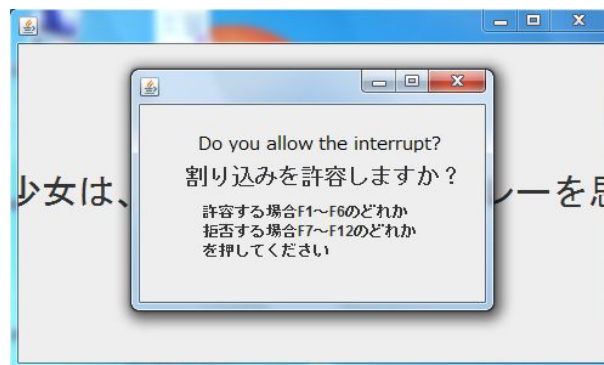


図 3.7: タスクの画面 (情報通知質問画面)

[pixel/s] (10[ms/pixel]) の4段階とする。(c) (b) (d) (c) (b) (a) (c) (d)

(b) (d) の順で1分毎に切り替え, 計10分のタスクとする。速度が変わらない1分の中に, 開始から10秒, 30秒, 50秒のタイミングで情報通知を許容するか否かを問うダイアログを表示する。ダイアログを図3.19に示す。ダイアログのサイズは300[pixel]×200[pixel]とした。ユーザはキーボードのファンクションキーを押すことで, その時点で情報通知を受け取ることが出来る状態か否かを答える。受理できる場合はF1-F6, 拒否の場合はF7-F12を押してもらおう。答えるとダイアログは消え, 元のタスクに戻ることができる。本実験では実際には情報通知の提示を行わず, 教示により想定させるのみとした。

タスク期間中は机上の圧力を測定し続ける。圧力のうち, 最も周期の速いものは, タイピングによる押圧と考えられる。従って, タイピングの押圧の周期より十分速い周期でサンプリングを行えばよい。予備実験の結果, キーを押している時間が約100[ms]

とわかったため、測定の間隔は30[ms]とした。

タスク期間中、机上にはキーボードとディスプレイ、ユーザの腕のみを載せる。ディスプレイのみを置いた状態の圧力をゼロ点とし、測定される圧力にはキーボードの重量とタイピングの加圧、腕の重量（加圧）のみが含まれるようにする。

特徴量抽出

本実験では作業内容をタイピングに限定しており、かつ机上に置く物をキーボードとユーザの腕のみに限定している。そのため特徴量として、表3.2の5つを採用する。

表 3.2: 使用する特徴量

	特徴量	内容
1	left	キーボードの左足
2	right	キーボードの右足
3	bottom	キーボードの手前側の足
4	left hand	ユーザの左手
5	right hand	ユーザの右手

机上に置く物を限定しているため、上記の特徴が現れる机上の位置（領域）はほぼ一定となる。そこでまず、センサを表3.3のように5つの領域に分割する。ここで、60×60のセンサに対し、横方向を x 、縦方向を y とし、左奥のセンサを $x = 0, y = 0$ 、右手前のセンサを $x = 59, y = 59$ などのように、座標で示す。平面上で示すと図3.8のようになる。

表 3.3: センサ領域の分割

	左上		右下	
	x 座標	y 座標	x 座標	y 座標
1	0	0	29	34
2	30	0	59	34
3	0	35	59	44
4	0	45	24	59
5	25	45	59	59

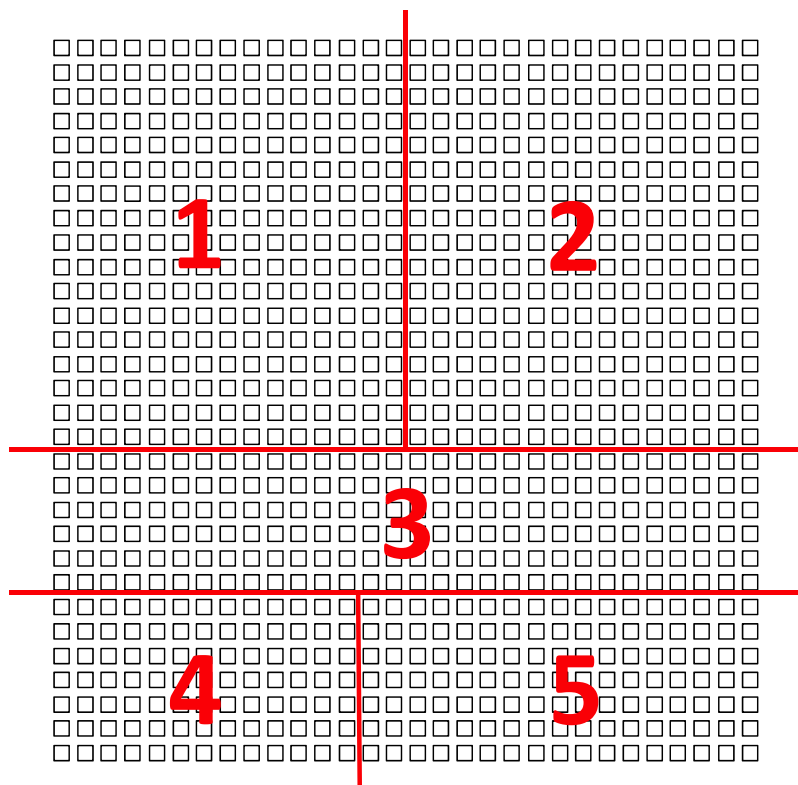


図 3.8: センサ領域の分割模式図

分割した5つの領域のそれぞれの中にあるセンサのうち、閾値より高い値を示すセンサに各特徴を表す圧力が加わっているものとみなした。閾値は経験的に20[-]とした。閾値によって得られた範囲の圧力を平均したものを、その瞬間における特徴量とした。そして、情報通知を行う1[s]前から30フレーム分(900[ms])の圧力データの平均をとり、そこにユーザが付与した可否のラベルを加えた計6つの属性を1つの訓練データとした。

分類学習アルゴリズム

本研究では、C4.5の実装として、weka3.6.4[Bouckaert 10]のJ48¹を使用した。wekaはJavaで記述されたGeneral Public Licenseの機械学習ソフトウェアであり、データ解析と予測モデリングのための視覚化ツールとアルゴリズムから成る。クラスはaccept

¹wekaに実装されているクラス名はweka.classifiers.trees.J48。



図 3.9: キーボード a : FK-EA4XA

と reject の 2 値，危険率 (confidence factor) は C4.5 で最もよく用いられる 0.25 を使用した。

キーボード

本研究ではキーボードのタイピング押圧を利用した推定を行うため，キーボードの特性，特にキーの反応押圧や重量，足の形状や面積などが推定に影響することが予想される．そこで本実験では，以下に示す 2 つのキーボードを用いて実験を行った．

- KFK-EA4XA((株) ミツミ電機:以後キーボード a とする，図 3.9)
- Realforce 91 NE0100((株) 東プレ:以後キーボード b とする，図 3.10)

キーボード a は一般的に用いられているキーボードの一つである．キーボード b はテンキーを持たず，人間工学的に設計された，特殊なキー押圧を持つキーボードであり，以下の様な特性を持つ．静電容量無接点方式のキーであるため，キーを最奥まで押し込まなくても入力される．キー毎に荷重設定，すなわち，入力に必要なキー押圧の強さが調整され，全体的に一般的なキーボードより軽い荷重で入力されるよう設計されている．これらの特性は，本研究で使用する特徴量として特に重要と考えられるキー押圧に対して直接的に影響することが予想されるため，異なる特性のキーボードとしてこれを採用した．



図 3.10: キーボード b : Realforce 91 NE0100

参加者

参加者は情報系の学生と教員，事務職で 23 歳から 51 歳 (平均 35.4 歳，標準偏差 11.7 歳) の男性 7 名女性 1 名とした．いずれも普段から PC を用いたデスクワークを頻繁にしており，タイピングに慣れている．

実験手順

各参加者に対し，まず教示を行った．教示内容は，以下の通りである．

教示内容

文字がスクロール表示されるので，それをできるだけ正確にタイピングしてください．変換は気にせず，読み仮名だけで構いません．タイピングの内容は記録されています．スクロール速度は途中で切り替わったり，止まったりします．タイピングしている途中で，割り込みを許容するかどうか尋ねられます．1 行程度の短い情報 (天気予報など) を読むことを想定し，受け取れるなら「許容 = F1 ~ F6 のいずれか」，受け取れないなら「拒否 = F7 ~ F12 のいずれか」を押してください．
ウィンドウが閉じたら実験終了です．

次に，片方のキーボードについて実験を行う．キーボードの使用順は，参加者間でカウンターバランスをとった．任意の時間休憩した後，もう片方のキーボードで実験を行う．

以上の手順を全参加者に対して行った。この時点で、各試行について、圧力センサの出力結果とユーザが回答した割り込み可否の記録、及びキータイピングの履歴が得られる。

次に、圧力センサの出力結果とユーザが回答した割り込み可否の記録を元に、特徴量を抽出した。

次に、以下の4パターンのデータセットを作成した。ここで、各パターンに属するデータセットの数と、各データセットに含まれる訓練データ数も合わせて記す。

- 各試行別（参加者別かつキーボード別：データセット数 16，訓練データ数 30）
- 参加者別（参加者別に2つのキーボードのデータをマージ：データセット数 8，訓練データ数 60）
- キーボード別（キーボード別に全参加者のデータをマージ：データセット数 2，訓練データ数 240）
- 合計（全てのデータをマージ：データセット数 1，訓練データ数 480）

ここで、accept と reject のデータ数の差の影響を取り除くため、weka に実装されている訓練データフィルタの1つである Resample² により、訓練データ数の調整を行った。

次に分類学習と分類性能の評価を行った。評価は 10-fold cross validation を用いた。

なお、ディスプレイは DELL 2001FP(20.1 インチ 解像度 1600×1200) を、PC は Let's note CF-R9((株)Panasonic CPU：Core i7 620UM/1.06GHz メモリ：2GB OS：Windows 7 Professional SP1 32bit) を使用した。

3.2.2 実験結果

参加者 c がキーボード b を使用した時に得られたデータの一部を表 3.4 に、得られた決定木を図 3.11 に示す。

²weka に実装されているクラス名は weka.filters.supervised.instance.Resample .
フィルタのパラメータは以下の通りである .
biasToUniformClass:1.0 invertSelection:False
noReplacement:False randomSeed:1 sampleSizePercent:100.0 .

表 3.4: データ例

left	right	bottom	left hand	right hand	class
26.40641	28.76054	27.045826	30.179441	25.326704	accept
26.664957	28.779667	27.27016	30.06281	26.796328	accept
26.558939	28.839512	27.280691	29.86009	27.982447	accept
26.603634	28.884605	27.049152	30.144327	27.759874	accept
26.592274	28.925966	26.955412	30.257694	27.309122	accept
26.874556	28.923357	27.190588	29.870623	23.86529	accept
26.929892	29.08362	26.964052	29.666239	27.913454	accept
26.675053	29.140356	27.037712	29.57113	25.158352	accept
26.697132	28.952013	27.28756	29.818966	27.815523	reject
26.713459	28.934538	27.142092	29.862135	26.681253	reject
26.842806	28.92776	27.214214	29.911165	25.62224	reject
26.822918	29.093777	27.181023	29.883703	22.817257	reject
26.815752	29.032724	27.108206	29.522215	24.039318	reject

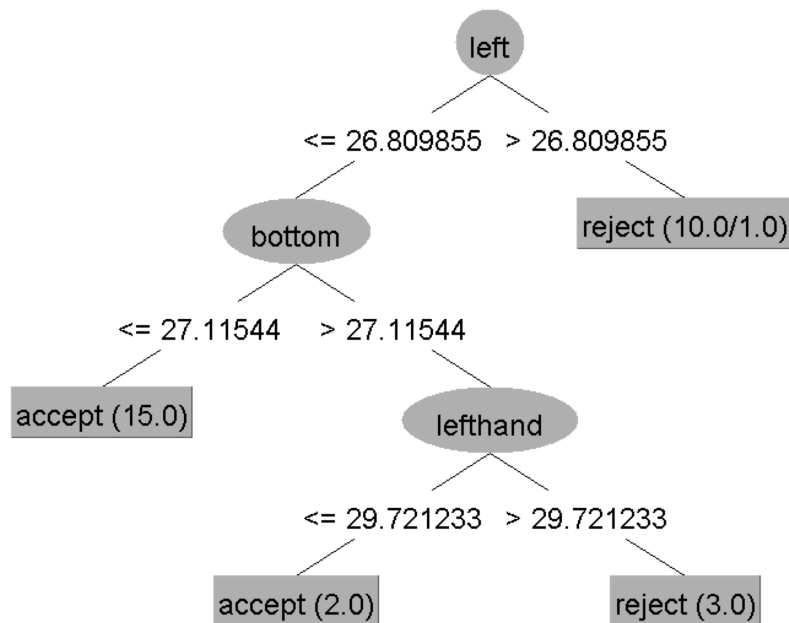


図 3.11: 決定木の例 (参加者 c キーボード b)

決定木を見ると、各ノードでは、値が低い場合に accept、高い場合に reject と判定されている様子が見える。

次に参加者 b の全てのデータを使用して得られた決定木を図 3.12 に示す。



図 3.12: 決定木の例 (参加者 b)

使用されている特徴量が、前の例と大きく異なっている。

次にキーボード b を用いた時の全てのデータを使用して得られた決定木を図 3.13 に示す。

前 2 つの例と比較すると、非常に複雑な決定木となっていることがわかる。これは、学習に用いたインスタンスの数の違いによるものと思われる³。

次に、10-fold cross validation の結果を表 3.5 に示す。ここで、TP Rate は正しく分類されたデータの割合、FP Rate は誤って分類されたデータの割合を表す。Precision(適合率)と Recall(再現率)、F-Measure は、それぞれ次式で求められる。

$$Precision = \frac{R}{N} \quad (3.5)$$

$$Recall = \frac{R}{C} \quad (3.6)$$

$$F = \frac{Precision \times Recall}{Precision + Recall} \quad (3.7)$$

³インスタンス数を揃えて学習を行った結果、決定木のサイズは前 2 つの例に近くなった。

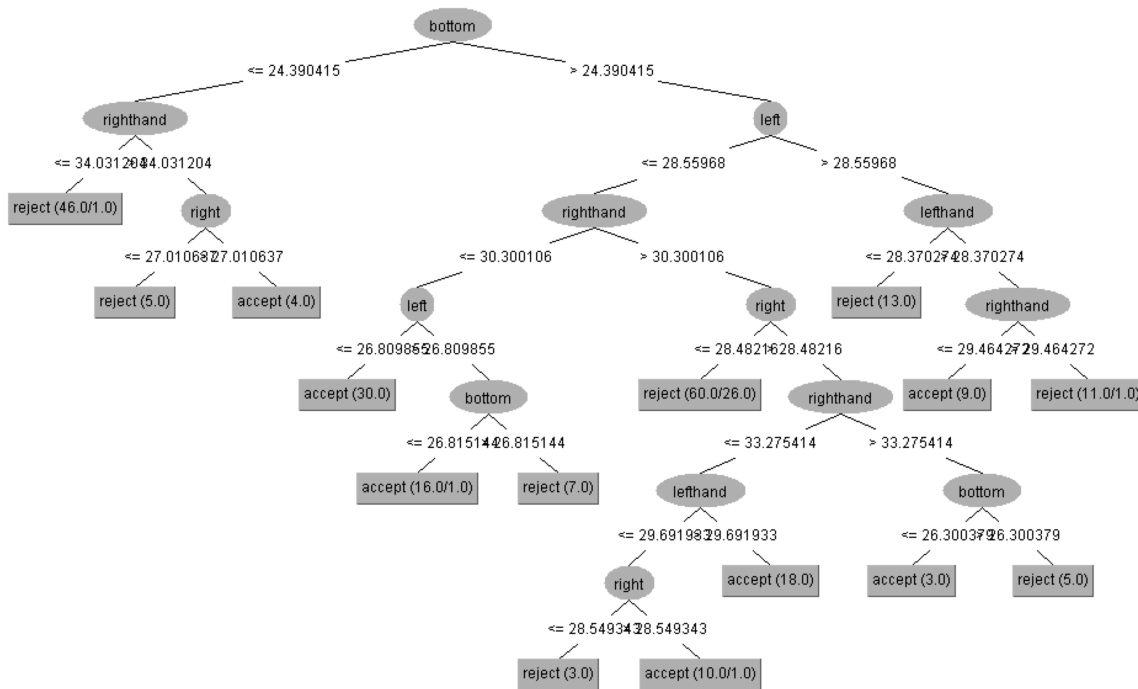


図 3.13: 決定木の例 (キーボード b)

ここで、 R は正しく分類されたデータ数、 N はそのクラスに分類されたデータ数、 C はそのクラスに属するデータ数である。

表 3.5 の各値は、各クラスに対する指標の重み付け平均であり、以下のように計算される。

$$x_{WA} = \frac{x_{accept} \times n_{accept} + x_{reject} \times n_{reject}}{n_{accept} + n_{reject}} \quad (3.8)$$

ここで、 x は各クラスに対する指標の値、 n は各クラスに属するデータ数である。

参加者毎、キーボード毎に学習した場合、うまく学習できない場合があることがわかった。これはインスタンス数が少ないためと考えられる。一方、参加者毎に決定木を生成した場合、83 % 程度の精度で分類を行えることがわかった。キーボード毎に決定木を生成した場合、80 % 程度の精度となった。そして、全てのインスタンスを用いて決定木を生成した結果、精度は 66 % まで低下した。

3.2.3 考察

異なるキーボードを用いた試行から得られた押圧を混合して学習，推定した場合でも，推定精度は低下しなかった．これはキーボードの差が押圧に影響しないことを示唆している．

表 3.5: 10-fold cross validation 結果

条件	TP Rate	FP Rate	Precision	Recall	F-Measure
参加者 a キーボード a	0.733	0.206	0.794	0.733	0.733
参加者 a キーボード b	0.833	0.194	0.833	0.833	0.832
参加者 b キーボード a	0.633	0.328	0.673	0.633	0.635
参加者 b キーボード b	0.900	0.094	0.903	0.900	0.901
参加者 c キーボード a	0.600	0.406	0.613	0.600	0.604
参加者 c キーボード b	0.833	0.167	0.837	0.833	0.834
参加者 d キーボード a	0.867	0.117	0.877	0.867	0.868
参加者 d キーボード b	0.833	0.111	0.832	0.833	0.834
参加者 e キーボード a	0.867	0.117	0.877	0.867	0.868
参加者 e キーボード b	0.867	0.200	0.891	0.867	0.860
参加者 f キーボード a	0.933	0.100	0.940	0.933	0.932
参加者 f キーボード b	0.933	0.100	0.940	0.933	0.932
参加者 g キーボード a	0.767	0.267	0.765	0.767	0.765
参加者 g キーボード b	0.767	0.267	0.765	0.767	0.765
参加者 h キーボード a	0.933	0.044	0.943	0.933	0.934
参加者 h キーボード b	0.900	0.150	0.914	0.900	0.897
平均	0.825	0.179	0.837	0.825	0.825
参加者 a 合計	0.850	0.158	0.853	0.850	0.849
参加者 b 合計	0.867	0.135	0.867	0.867	0.867
参加者 c 合計	0.733	0.274	0.733	0.733	0.732
参加者 d 合計	0.850	0.149	0.851	0.850	0.850
参加者 e 合計	0.767	0.235	0.767	0.767	0.767
参加者 f 合計	0.950	0.044	0.955	0.950	0.950
参加者 g 合計	0.800	0.202	0.800	0.800	0.800
参加者 h 合計	0.800	0.206	0.800	0.800	0.799
平均	0.827	0.175	0.828	0.827	0.827
キーボード a 合計	0.821	0.178	0.822	0.821	0.821
キーボード b 合計	0.783	0.212	0.791	0.783	0.783
平均	0.802	0.195	0.807	0.802	0.802
総計	0.665	0.352	0.670	0.665	0.657



図 3.14: 実験に使用したキーボード : SKB-LKG3BK

3.3 評価実験

3.3.1 実験環境とタスク

本研究ではユーザがPCを利用したデスクワークをしている環境を想定しているが、本実験ではできるだけ一般的で単純化したデスクワーク環境とタスクを構築した。図 3.16 に実験の様子を示す。キーボードはSKB-LKG3BK((株)サンワサプライ, 図 3.14), マウスはLS-1BK((株)Logitech, 図 3.15), ディスプレイはDELL 2001FP(20.1 インチ 解像度 1600×1200), PC は Let 's note CF-R9((株)Panasonic CPU: Core i7 620UM/1.06GHz メモリ: 2GB OS: Windows 7 Professional SP1 32bit) を使用した。この PC は、ディスプレイ描画及び圧力データの記録に使用した。タスクは、PC によるデスクワークとして考えられる典型的なタスクとして、情報通知の先行研究 [Iqbal 08] で用いられているタスクを参考にして、タイピングタスクとマウス操作タスクの2つを用意した。タイピングタスクは、画面中段を右から左にスクロールする文字列をできるだけ正確に書き写すタスクである(図 3.18(a))。表示される文字はランダムなローマ字文字列で、タイピングされた文字列は画面下段に表示される。画面サイズは 500pixel×500pixel、表示文字は MSP ゴシック、30pixel である。押下されたキーはキーコードと時刻をすべて記録される。スクロール速度を変化させることにより、タスクの負荷を調整する。タイピング中、割り込みを許容するか否かを尋ねるダイアログ(図 3.19)が 20 秒間隔



図 3.15: 実験に使用したマウス : LS-1BK

で画面中央に表示される。割り込みに対する回答には、回答の動作が机上の圧力に与える影響を最小限に留めるために、図 3.17 のようなフットスイッチを用いた。参加者は用意された 2 つのフットスイッチのうち、許可 (accept) なら左のフットスイッチを、拒否 (reject) なら右のフットスイッチを踏むことで回答する。参加者には、食事の誘いのような、緊急性が低い情報が提示されることを想定して可否を決定するよう教示を与えた。回答するとダイアログは消え、メインタスクを再開する。

マウス操作タスクは、図 3.18(b) の画面を使用し、青く四角いマーカをクリックするタスクである。マーカは一定周期 (数秒間隔) でランダムな位置に表示され、次の四角が表示されると同時に消える。消えるまでにクリックに成功すると、色が赤に変化してから消える。表示周期を変化させることにより、タスクの負荷を調整する。クリックした位置と時刻は四角の内外に関わらずすべて記録される。画面サイズは 500pixel×500pixel、マーカサイズは 20pixel×20pixel である。タイピングタスクと同様にダイアログが表示される。



図 3.16: 実験環境



図 3.17: 実験環境のフットスイッチ

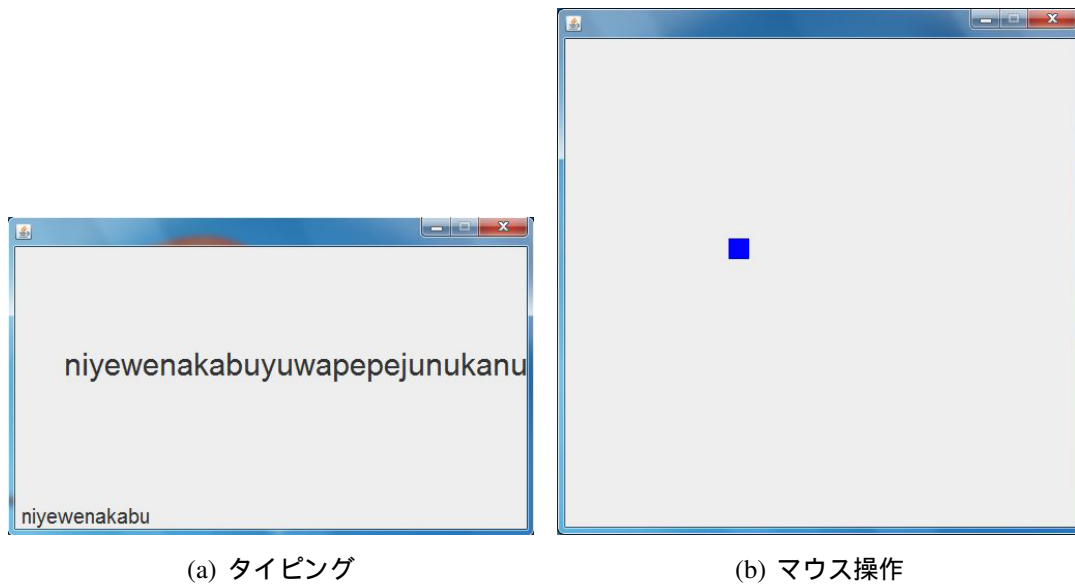


図 3.18: 2つのタスクにおけるタスクウィンドウ

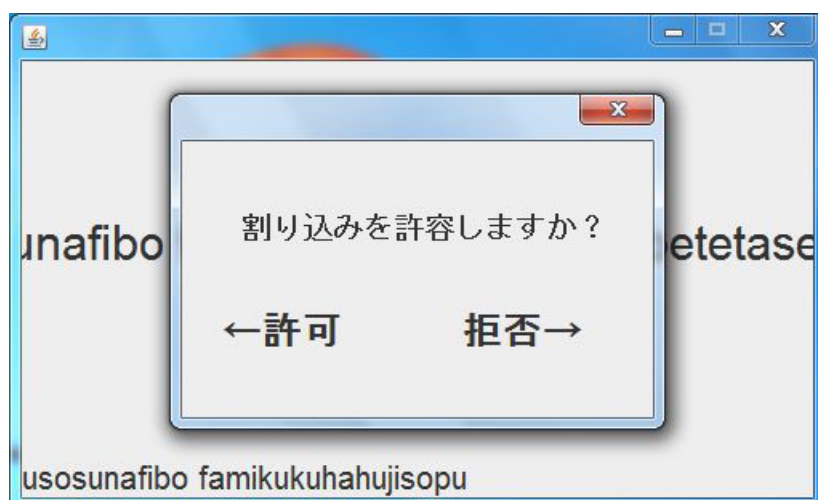


図 3.19: 情報通知ダイアログ

3.3.2 参加者と実験手順

参加者は情報系専攻の大学院生と研究所職員の20名(23歳～54歳,平均33.7歳,標準偏差9.9歳,男性9名,女性11名)であった。

許可と拒否の両方のデータを得るために,メインタスクの負荷が高いフェーズと低いフェーズが必要である。負荷が高いフェーズでは,メインタスクに集中する必要があるため,参加者は通知を拒否し,負荷が低いフェーズでは,メインタスクに集中する

必要がないため、参加者は通知を許可することが予想されるからである。そこでタイピングタスクではスクロール速度を 1/12 pixel/ms , 1/20 pixel/ms , 1/30 pixel/ms , 1/40 pixel/ms , 1/50 pixel/ms , 1/60 pixel/ms の6段階、マウス操作タスクでは表示間隔を 1.2s , 1.5s , 2s , 3s の4段階の負荷を用意した。これらの負荷は、予備実験により、最も熟練した参加者がかるうじて追従できる速度を上限とし、その近傍では微調整が可能なように間隔を決定した。そして、各参加者に対し、実験に入る前にタイピング能力やマウス操作能力を調査し、負荷にかるうじて追いつけるものを中負荷、それより一段階低いものを低負荷、一段階高いものを高負荷として、3段階の負荷を与えた。

負荷調整の後、本実験を行った。各参加者に対し、2つのタスクをそれぞれ2回試行、すなわち計4試行とした。各試行では、1分を1フェーズとして負荷が高いフェーズと低いフェーズを織り交ぜ、5フェーズ行った。各フェーズ中、20秒間隔で3回通知を行った。総計で20分間タスクを行うことになる。

分類学習の実装として、一般的なデータマイニングプログラムライブラリである weka3.5.4⁴ 上に実装された、Random Forests(number of trees = 10) , SVM の実装である LibSVM(RBF kernel , $\gamma = 0.042$) , C4.5 の実装である J48(confidence factor = 0.25) を使用した。

3.3.3 実験結果

表3.6に得られた特徴量を抜粋して挙げる。この表からは、accept と reject の間で、各特徴量の値に大きな差は見られない。得られたデータ数は全体で1199個(許可722, 拒否477)、内タイピングタスクは600個(許可333, 拒否267)、マウス操作タスクは599個(許可389, 拒否210)である。なお今回の実験では、全参加者のデータをまとめたものを使用して分類を行う。許可と拒否のデータ数に大きな差が見られなかったため、正データ・負データの量のインバランスを補正するアンダーサンプリング [Drummond 03] は適用しなかった。

次に、得られたデータを3つの分類学習に与え、交差検定(10-fold)による学習実験

⁴<http://www.cs.waikato.ac.nz/ml/weka/>

表 3.6: データの例 (左腕)

	Area	Pressure	CoG _x	CoG _y	クラス
左腕	29.6	36.4	54.9	341.4	<i>reject</i>
右腕	15.6	22.4	7.3	43.3	
左足	77.4	83.6	1616.9	3027.5	
右足	70.2	122.0	1089.3	1362.1	
手前足	28.6	58.5	346.1	113.7	
マウス	3.9	9.1	12.8	61.3	
左腕	29.4	36.7	54.2	341.9	
右腕	18.4	23.6	10.0	44.5	
左足	75.5	85.3	1615.7	3027.0	
右足	69.1	125.0	1088.3	1363.9	
手前足	31.7	56.8	343.0	1363.9	
マウス	4.9	9.8	12.3	61.4	
左腕	28.4	37.5	55.5	341.2	<i>accept</i>
右腕	17.8	24.2	9.8	44.2	
左足	76.0	83.6	1615.2	3027.1	
右足	69.9	121.9	1088.6	1364.1	
手前足	30.4	56.4	344.3	61.4	
マウス	5.4	10.4	11.8	61.4	
左腕	27.9	38.0	54.7	341.2	
右腕	17.7	23.7	9.5	44.4	
左足	77.7	86.4	1616.0	3027.0	
右足	70.5	124.1	1089.5	1364.8	
手前足	30.8	58.0	345.6	113.8	
マウス	5.0	10.1	11.8	61.4	

を行った。実験結果として、各学習アルゴリズムにより学習されたクラシファイアの精度を表 3.7 に示す。ここで精度は、許可 / 拒否の 2 クラスについて正しく分類された訓練データの割合である。

表 3.7 から、タイピングタスク、マウス操作タスク、両タスク総計のいずれの条件でも、SVM が最も高い精度を達成しており、本研究では SVM を適用すべきである。また、タイピングタスクの方がマウス操作タスクよりも、高い精度で割り込み可能性を判定できていることがわかる。

表 3.7: 3つの分類学習アルゴリズムの精度(下線が各タスクの最良値, 太字がすべてにおける最良値)

分類学習	精度 [%]		
	タイピング	マウス操作	全体
SVM	<u>76.8</u>	<u>72.3</u>	<u>73.6</u>
Random Forests	75.8	69.6	73.1
C4.5	71.7	71.6	71.3

3.4 考察

3.4.1 机上の圧力を用いた状態推定の評価

実験の結果, タイピングタスクでは約 77%, マウスタスクでは約 72% の精度でユーザの割り込み可能性を推定することができた(表 3.7). 職場での PC を用いたデスクワークの大半は文書作成であると考えられるため, タイピングタスクの結果である約 77% が重要である. チャンスレベルは約 50% であるため, 机上の圧力から得た情報に, 割り込み可能性推定に有用な情報が含まれていることが示唆された. これは, 机上の圧力を用いたユーザの状態推定を目指した先駆的な研究として, 今後の発展を見込める結果だと考えられる. また, 提案手法はコンピュータビジョン [Fogarty 05] やタイピングパターン [Epp 11] を利用した既存の手法とモダリティが干渉しないため, それらと組み合わせた利用も可能である.

一方, 表 3.7 の SVM の精度を見ると, マウス操作タスクの推定精度はキーボードタスクより低い. この原因として, キーボードの足の押圧に比べ, マウスの押圧は LL Sensor で測定が困難である可能性や, マウス操作による押圧の変化が少ないことなどが考えられる. マウス操作タスクについて推定精度を向上することは, タスク全体の推定精度の向上と共に, 今後の重要な課題である.

3.4.2 キーボード, マウスの非操作時における状態推定

提案手法では, ユーザがタイピングやマウス操作を行っていない間は机上の圧力に変化がないため, 割り込み可能性は推定できない. そこで本研究では, ユーザがキー

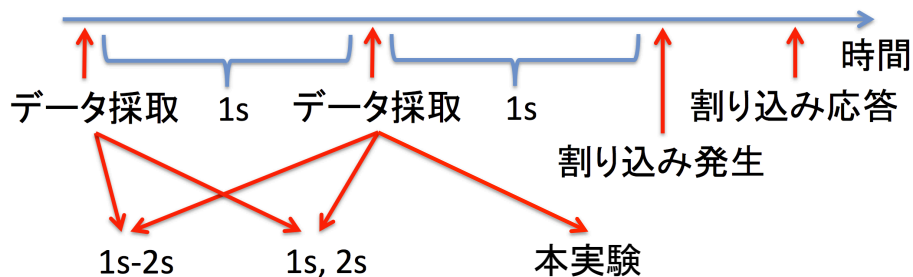


図 3.20: 圧力測定のタイムライン

ボードやマウスで作業をしていない間は割り込み可能であると仮定している。

しかし実環境では必ずしもそのような仮定は成立しない。例えば、考え事、ウェブページ閲覧、動画鑑賞などに集中している時は、キーボードやマウスを操作していない間でも割り込み不可能な状態が存在する。この問題に対する対して提案手法による解決策として、特徴を拡張することが考えられる。例えばマグカップの重量や、腕を机の上に置いた際の形状などの特徴を追加することが簡単にできる。これらは圧力センサで取得可能な情報である。ユーザが集中状態にあれば、マグカップを頻繁に取り上げなくなる、腕の位置を変えなくなるなどの現象が予想されるため、これらの特徴により机上の圧力による解決が期待できる。

3.4.3 時系列特徴の効果

前節の評価実験においては、ある時刻における 24 特徴のスナップショットをデータとして利用したが、異なる時刻のデータ間の差分が有益な情報を持っている可能性がある。そこで、そのような圧力の時間変化にユーザ状態推定に有用な情報が含まれているか検証するために、図 3.20 のように、時系列上のデータを複数、及び差分を与えて推定を行った。その結果の一部を表 3.8 に示す。ここで、各数字は情報通知と使用したデータとの時間差を表し、“-”はその差分をデータとして使用していることを表す。すなわち、“1s”は割り込み発生の 1 秒前のデータを用いており、これは表 3.7 のタイピングタスクの結果と同一である。従って用いる特徴量は 24 である。“1s, 2s”は割り込み発生の 1 秒前と 2 秒前のデータを用いており、各データ採取ポイントから 24 特徴量

表 3.8: 時系列データを使用した場合の推定精度 (下線が各学習アルゴリズムの最良値, 太字はすべてにおける最良値)

学習アルゴリズム	精度 [%]			
	1s	1s,2s	1s,1s-2s	1s-2s
SVM	<u>76.8</u>	74.3	74.0	57.0
Random Forests	<u>73.2</u>	<u>73.2</u>	66.7	61.2
C4.5	<u>68.5</u>	68.7	61.2	55.5

が得られるため、あわせて 48 特徴量を用いて学習及び分類を行なっている。“1s-2s”は割り込み発生の 1 秒前と 2 秒前のデータの差分を用いており、各特徴量毎に差を取るため、特徴量の数は 24 となる。

図 3.20 に示した他にも [1s,5s] [1s,10s] [1s-3s] [1s-2s, 1s-3s] など 11 通りの組み合わせを調査したが、いずれも当初の精度を上回ることなく、本実験の環境において、採用した 24 特徴量について、圧力の時間変化にはユーザ状態推定に有用な情報は乏しいか存在しないことが実験的に示された。

しかし、直感的には時間的な変化に情報が含まれると予想され、これと矛盾する。本実験において時間変化から情報が得られなかった理由として、センサの応答特性の影響が考えられる。本実験で使用したセンサは押圧の変化に対する応答特性が非常に低く、更にヒステリシスがあるようなセンサであった。これにより、数秒程度の時間幅内の押圧変化は平均化された押圧値が観測されていたと考えられる。

更に、当然ながら本実験で採用しなかった特徴量についてはこの限りではない。例えばマグカップの重量は、内容物の減り方が状態推定に寄与していることが直観的に予測できる。

一方、時系列データを使用せず、情報通知からの時間差を 30ms から 10s の間で変化させた結果、各学習アルゴリズムでの最高精度は、Random Forests は 200ms で、C4.5 は 5s で、いずれも 74.8%となった。これらは SVM には及ばないが、学習アルゴリズム毎に調整を行うことで精度が向上する可能性を示唆している。

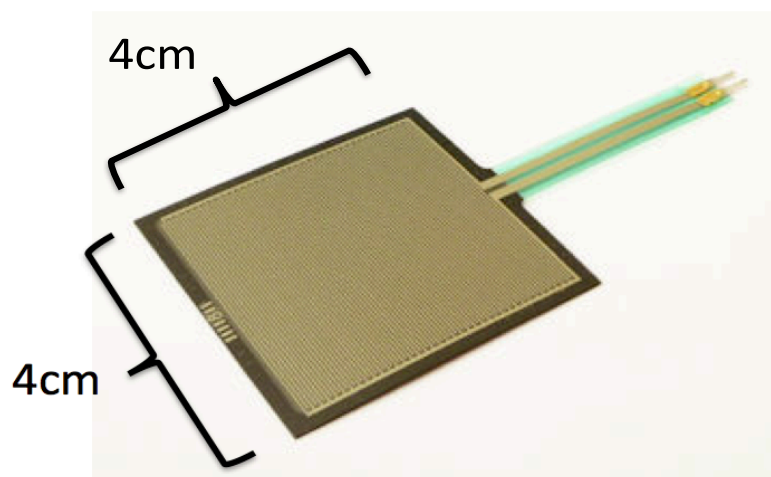


図 3.21: 小型で安価なセンサ

3.4.4 より実用的な特徴選択

これまでの実験により提案手法の有効性を確認できたが、LL Sensor は微小な圧力センサを格子状に持つ特殊なセンサで、容易に入手できる安価なセンサではない。従って実用にはコストが問題となる。この問題の解決法として、図 3.21 に示す、FSR406 (Interlink Electronics Inc.) のような、小さく安価な圧力センサを使用することが考えられる。このような小さなセンサはキーボードの 3 つの足とマウスの底面に簡単に取り付けることができる。この手法で LL Sensor を使用した場合と同等の精度で状態推定が可能ならば、提案手法はより実用的なものとなる。

このような小型のセンサを計測点に貼り付ける手法は、実験で獲得した 24 の特徴量データのうち、キーボードやマウスが机の上に与える圧力の特徴量だけを用いて推定することにより、シミュレートできる。SVM によって上記の特徴量から推定した結果を表 3.9 に示す。なお、この表で、一番下の行の「すべての特徴量」が、LL sensor を用いた場合の表 3.7 の SVM の結果である。タイピングタスク、マウス操作タスクのそれぞれについて、キーボードとマウスの特徴量だけを用いた場合と、すべての特徴量を用いた場合で、ほぼ同等の精度が得られたことがわかる。よって、キーボードの足とマウスの裏に図 3.21 のような小型の圧力センサと装着できる環境においては、提案方法をより実用的に利用できることがわかった。

表 3.9: 局所的にセンサを配置した場合の推定精度

選択された特徴	TP Rate [%]		
	タイピング	マウス操作	全体
キーボードの圧力	74.7	—	—
マウスの圧力	—	64.3	—
キーボード圧+マウス圧	75.8	70.0	69.8
すべての特徴量	76.8	72.3	73.6

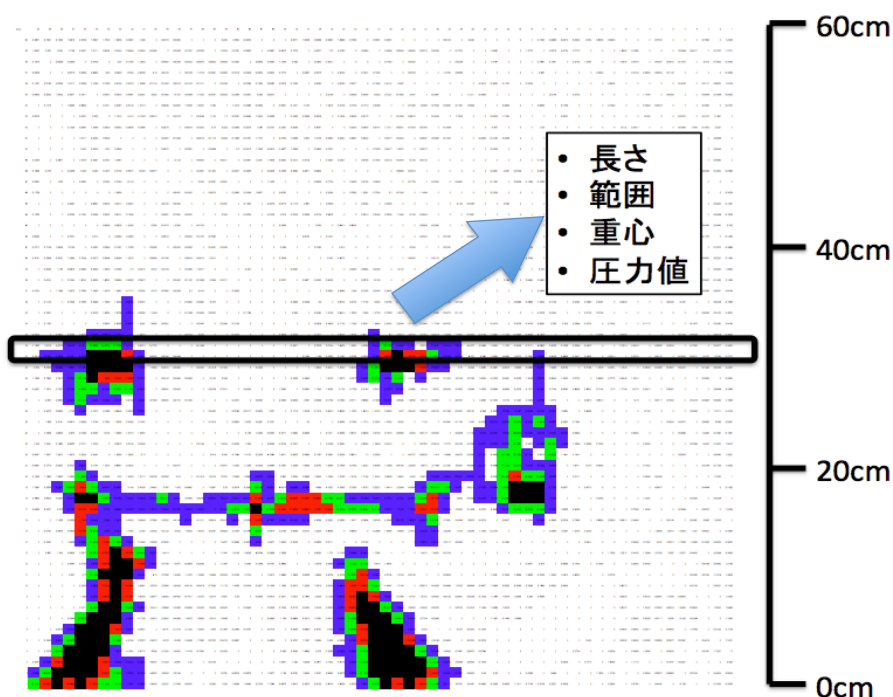


図 3.22: 直線状にセンサを配置した場合のシミュレート

また、他の特徴選択の方法として、圧力センサを平面全体に敷かず、机の手前から5cmや10cmなどの位置に横一列の直線上に配置する手法が考えられる。そのような手法の有効性は、図3.22のように、実験で得られた平面のデータから、直線上にあるセンサを手前から奥に走査することでシミュレートできる。ここで、抽出する特徴量は次の4つとした。

- 圧力が掛かっている範囲(長さ)
- 圧力が掛かっている個数(腕が2本あれば2個, など)

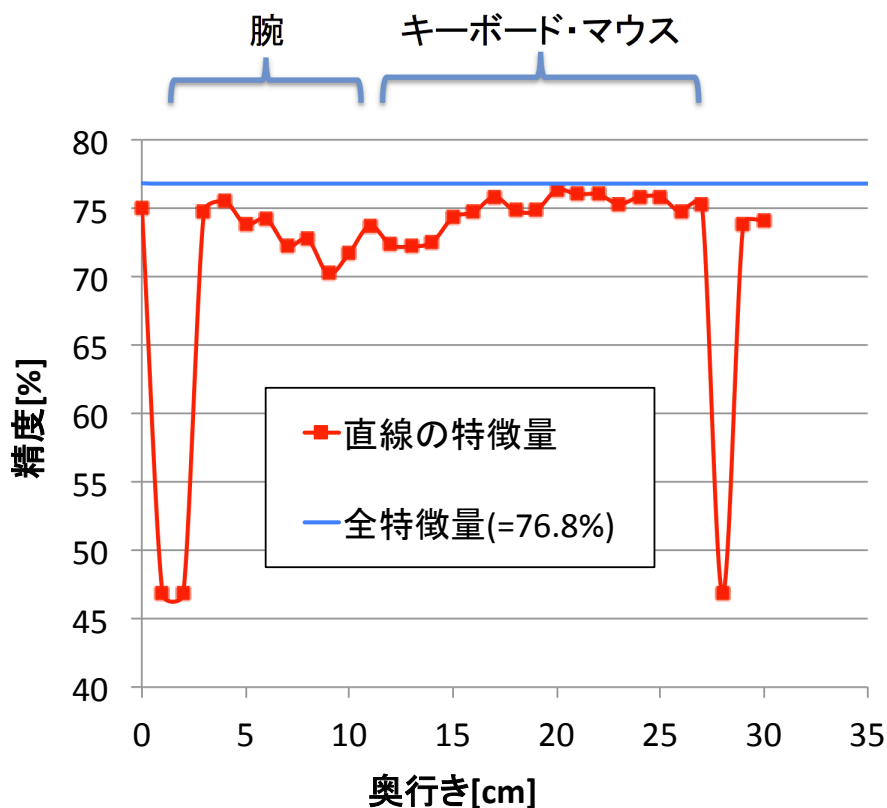


図 3.23: 直線状にセンサを配置した場合の推定精度

- 重心
- 圧力の平均値

この手法を用いてタイピングタスクについてSVMを用いて状態推定した場合の推定精度を図3.23に示す。ここで、青線は24特徴量すべてを使用した場合で、表3.7のタイピングタスクをSVMを用いて推定した場合と同一である。すべての特徴量を用いた場合と比較して若干劣るが、机の手前からの距離が3cm～4cm及び20cm～26cmの範囲において、LL sensorを使った場合とほぼ同等の精度が得られたことがわかる。つまり、この実験結果により、ある範囲に横一列に小型圧力センサーを設置することで、LL sensorと同程度の割り込み可能性の推定が可能であることが示唆されたと言える。

また、腕が置かれている領域(奥行き0cm～10cm)とキーボードやマウスが置かれている領域(10cm～30cm)の両方から同程度の精度で推定できていることから、これらの圧力にも、推定に有用な情報が含まれていることがわかった。一方、机の手前から

の距離が 1cm ~ 2cm 及び 28cm の位置で極端に精度が落ちているが、これはセンサの劣化により一部のセンサセルの精度が悪化し、圧力を正確に測定できていなかった可能性があり、該当センサを使用する推定が正常に行われなかったことが原因と考えられる。

以上のことから、前述の安価なセンサの利用と共に、設置条件が許されれば、直線上のセンサ配置も実用的な特徴として有効であることがわかった。

3.4.5 本研究の適用範囲

本研究の実験はいくつかの条件の元でのみ適用されうるため、その条件について考察する。

まず、本実験では、割り込みの可否を問うダイアログを画面中央、メインタスクを隠すように表示している。これは表示位置としては最も妨害的であり、参加者に対しストレスを与える位置であると言える。しかし、表示位置を画面端などの妨害的でない位置とした場合、周辺視の効果によりダイアログに気づかない可能性が高くなり、任意の時点での割り込み可能性と押圧のデータが得られなくなることが予想される。つまり、位置がどの程度妨害的か（参加者のストレスの高さ）と、十分なデータが得られることがトレードオフの関係になっている。本研究では十分なデータを得ることを優先し、画面中央にダイアログを表示することとした。一方、実験におけるダイアログは、あくまでも割り込みの可否を問うだけのものであり、割り込みを許可しても実際に何らかの作業を求めるものではない。更に、可否はフットスイッチを用いて答えるものとしているため、回答にかかるコストは最小限のものとしている。これは、実験で表示したダイアログが実際の割り込みとは必ずしも同じ性質のものでないことを意味する。ただし、ここで本質的かつ重要な要素は割り込みの可否であり、そこに直接的な影響を与える要素は参加者にかかるストレスであり、それが十分に取り除けたか否かについては不明である。

また、本実験では、割り込みの重要度の統制は、あくまでも教示によりイメージを与えるに留まっている。週末の食事の予定、という教示を与えているが、これにより

参加者がどの程度の重要度を想定したか検証は行っていない。また、これ以外の重要度の割り込みについては実験を行っていない。更に、メインタスクの負荷を変えることで繁閑の変化を誘発したが、参加者がタスクにどの程度没入、あるいは集中したかなどといったことは解析していない。

本実験では学習に十分な量のデータが得られていないため個人適応を行っていないが、机上にかかる圧力の性質は個人差、あるいは傾向のグループ化ができると予想される。これにより推定精度の向上が見込まれる。

更に、本実験では、あくまでも参加者1人の環境におけるデスクワークを想定しており、他者と会話しながらデスクワークするような環境は想定していない。しかし、仮にそのような環境で提案法を使用した場合、例えば参加者の体が会話対象の方向を向くことで、机上にかかる圧力の左右の配分が変化する可能性がある。あるいは、会話していても会話対象を向いていなければ机上にかかる圧力に変化がなく、割り込み可能性が低下していないと推定されうる。

第4章 タッチパネルにおけるUIデザインを考慮した操作特性モデル

4.1 タッチパネル入力の問題点

近年，スマートフォンやタブレットPCなど，インターフェイスとしてタッチパネルを利用したデバイスが普及し，仮想エージェントを実装するデバイスとしても多く利用されている．タッチパネルによる操作は従来の機械的なボタンによる操作に比べ直感的に行えること，UIの自由度が高いことなど多くの利点がある [Pickering 86] ．

一方，タッチパネルはタップ対象(ボタン，キー，アイコン)が物理的に区切られていないため曖昧であり，意図と異なる入力が行われやすい．また，入力時に機械的な変化を伴わない，すなわち機械的フィードバックがないため，ユーザ自身が与えた入力を確認する手段が，従来の機械的ボタン入力に比べ乏しい．これらの理由により，入力精度が低下することが知られている．更に，機械的な抵抗を伴わず入力されるため，気づかないうちに触れてしまうなど，意図せず入力される場合もある．特にスマートフォンは画面領域が狭いため必然的にUIも小さくなり，隣のボタンが押されたと認識される fat finger 問題 [Siek 05] も発生しやすくなるため，入力精度の低下が顕著である．

この問題はポインティング操作の精度低下の一種と言える．今後デバイスの進歩によりタッチパネル以外の入力デバイスが普及することが予想されるが，直接操作 (direct manuputation) を用いた電子機器の普及が進むことで，ポインティング操作はより重要性を増すと考えられ，その精度向上は重要な課題である．

この問題を解決し入力精度を向上する研究は，図4.1のような，スマートフォンのソフトウェアキーボードについて盛んに行われている．ソフトウェアキーボードは，狭い領域に数多くのキーを配置する必要があり，必然的に個々のキーが小さくなるため，

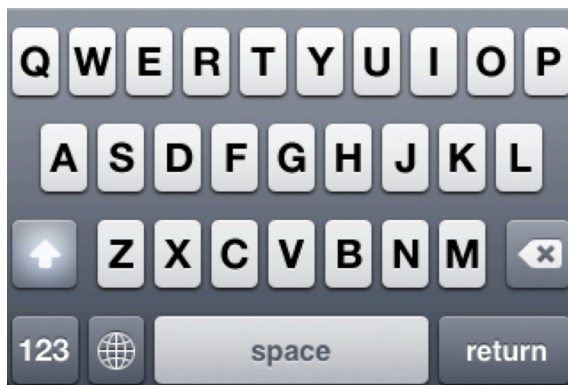


図 4.1: スマートフォンのソフトウェアキーボードの例

fat finger 問題が発生する代表的な事例である。

これらの研究は、主に2種類の手法を用いている。第一は言語モデルを用いる手法 [Al Faraj 09, Goodman 02] で、利用言語の情報を用いて、入力された文字列パターンから次に入力される文字を確率的に予測するものである。第二はタップモデルを用いる手法 [Findlater 12, Himberg 03] で、画面上に表示したタップ対象に対し、ユーザが実際にタップする座標の差を統計的に算出し、平面的に補正するものである。また、上記2種類の手法を組み合わせる研究も行われている [Goel 13, Gunawardana 10, Rudchenko 11]。

また、アプリケーションを限定せず入力精度の向上を目指す研究も行われている [Holz 11, Weir 12]。これらの研究は、タップにおける指の角度や、ユーザが意図したタップ位置とデバイスに入力されるタップ位置の関係など、人間のタップ特性をモデル化し、タップ位置の補正を行う。一方、インターフェイス形状の違いがタップ特性に与える影響は考察されていない。

そこで本研究では、インターフェイス形状の違い、すなわちタップ対象の色、サイズ、及び近傍のタップ対象との距離がタップ特性に与える影響を解析、モデル化し、入力精度向上を目指す。

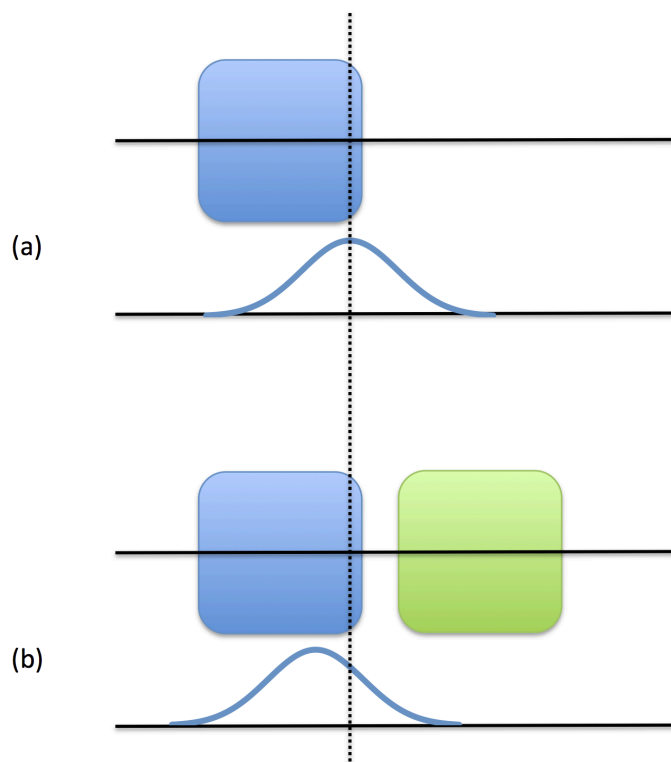


図 4.2: タップ対象とタップ位置

4.2 タップ位置補正による入力精度向上

4.2.1 インターフェイス形状によるタップモデルへの影響

タッチパネルを用いた入力にはタップ（画面上の1点に触れて離す）、ドラッグ（画面上の1点に触れたまま接触点を移動する）、フリック（画面上の1点に触れたまま接触点を移動する動作のうち、素早く1方向にのみ動かす動作）、ピンチ（画面上の2点に触れたまま接触点間の距離を変化させる動作）など様々な操作が用いられる。本研究ではこれらの動作のうち最も基本的で、かつ他の動作の要素にもなるタップのみを考える。

図 4.2 において、青い四角形をタップ対象とする。先行研究 [Goel 13, 萩谷 12] によると、人間のキネマティクス的な要因により、手のホームポジションからの移動距離を小さくする方向にずれることが知られている。そのため、右手で操作する場合、青線のような分布を持つ（図 4.2(a)）。

しかし，実環境でのタップにおいては，インターフェイスの形状によりタップ位置が変化することが予想される．すなわち，タップ対象の近傍に別のオブジェクト(緑の四角形)が存在する場合，ユーザはそれが誤反応しないよう意識するため，タップ位置は緑の四角形から遠ざかるように移動すると考えられる(図4.2(b))．他にも，タップ対象の色や形状，大きさなどにより，タップモデルが変化する可能性がある．

4.2.2 エラーモデル

タップにおいて人間が意図したタップ座標 $p_c = (x_c, y_c)$ とシステムが認識するタップ座標 $p_s = (x_s, y_s)$ の間にはしばしばエラー $e = (x_e, y_e)$ が生じる．すなわち

$$p_c = p_s + e^T \quad (4.1)$$

となる．

先行研究で指摘されているキネマティックエラー e_k は，ユーザの手の形状 h やタップ位置 p_s の影響を受ける．

$$e_k = f(h, p_s) \quad (4.2)$$

この関数 f は p_c, p_s, h から回帰により得る．

本研究では更に，インターフェイスの形状によりエラー e_c が生じると考える．すなわち

$$e = \alpha e_k + (1 - \alpha) e_c^T \quad (4.3)$$

ここで α は重み付けの係数である．

e_c はタップ位置周辺のタップ対象の配置により決定される．また，タップ位置 p_s の影響を受けることが予想される．タップ対象のインデックスを n ，タップ位置とタップ対象の距離を l ，タップ対象の大きさを s ，タップ対象の色を c とすると，

$$e_c = \sum_n f_n(l_n, s_n, c_n, p_s) \quad (4.4)$$

となる．

本研究では，各エラーのモデル獲得に，ガウス過程回帰 (GPR) [Rasmussen 05] を用いる．ガウス過程回帰の詳細は後述するが，ノンパラメトリック回帰であるため，求めようとする関数の様子が事前にわかっている必要がない．一方，学習にかかるコストが高い，すなわち，学習に時間がかかる，メモリが多量に必要であるという欠点を持つ．タッグモデルの獲得においては，関数の様子は事前にわかっておらず，学習には十分な時間とメモリを使用することができるため，ガウス過程回帰を採用した．

4.3 ガウス過程回帰

本節では，本研究に用いるガウス過程回帰について説明し，更に，本研究で必要となる，多変数出力の回帰に拡張することを考える．

4.3.1 多変量ガウス分布

ガウス分布 \mathcal{N} を考える．

このガウス分布は多変量であるとする．すなわち

$$\mathcal{N}(\mathbf{x}) \tag{4.5}$$

$$\mathbf{x} = (x_1, x_2, \dots, x_n)^t \tag{4.6}$$

である．平均 $\boldsymbol{\mu}$ と分散 (共分散行列) $\boldsymbol{\Sigma}$ も多次元であり，

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{4.7}$$

$$\boldsymbol{\mu} = (\mu_1, \mu_2, \dots, \mu_n) \tag{4.8}$$

$$\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2n} \\ \vdots & & \ddots & \vdots \\ \sigma_{n1} & \sigma_{n2} & \cdots & \sigma_{nn} \end{pmatrix} \tag{4.9}$$

となる．

このとき，条件付き分布 $\mathcal{N}(a|b)$ の平均と分散は

$$\boldsymbol{\mu}_{a|b} = \boldsymbol{\mu}_a + \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}(\mathbf{x}_b - \boldsymbol{\mu}_b) \quad (4.10)$$

$$\boldsymbol{\Sigma}_{a|b} = \boldsymbol{\Sigma}_{aa} - \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}\boldsymbol{\Sigma}_{ba} \quad (4.11)$$

となる．

4.3.2 ガウス過程

\mathcal{N} が，より次元の大きな多変量ガウス分布 $\mathcal{N}_p(\boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p)$ の条件付き分布である場合を考える． \mathcal{N}_p の条件付き分布 $\mathcal{N}_*(\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*)$ を仮定する． \mathcal{N} に対し， \mathcal{N}_* が加わった $\mathcal{N}_+(\boldsymbol{\mu}_+, \boldsymbol{\Sigma}_+)$ も，当然 \mathcal{N}_p の条件付き分布（あるいは \mathcal{N}_p 自体）となる．

このような \mathcal{N}_p をガウス過程と呼ぶ．

ガウス過程は多変量ガウス分布と近い概念であると考えられる．ただしガウス過程では，変数の個数が無限大になる．

4.3.3 ガウス過程の回帰への応用

ある入出力系 $y = f(t)$ が，ガウス過程 $\mathcal{N}_p(\boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p)$ に従う状態をとるものと仮定する．つまり

$$\mathcal{N}_p(y, t) \quad (4.12)$$

である．

これは \mathcal{N}_p の様子がわかり t が決まれば $\mathcal{N}_p(y|t)$ がわかることを意味する．そこで，既知の入出力データ（訓練データ）からガウス過程の様子を推定することで，未知入力 t_* が与えられた時の y_* の平均及び分散を得る．これがガウス過程を利用した回帰，すなわちガウス過程回帰である．

上記の通り，ガウス過程回帰は，関数のパラメータを同定するのではなく，ある入力値とその時の出力値が，他の入力値とその時の出力値に対して，どの程度類似しているかだけを元に出力値を推定する．そのため，ノンパラメトリック回帰に分類される．

4.3.4 ガウス過程回帰の多次元出力への拡張

ガウス過程回帰に限らず，一般的に回帰は1つのスカラー値を出力する関数を回帰する．一方，本研究では平面に対するエラー，すなわち2次元ベクトルを推定する必要がある．そこでガウス過程回帰を拡張し，多次元出力を推定する関数を回帰することを考える．

ガウス過程回帰により得ようとする関数の出力を

$$\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_d(\mathbf{x}))^t \quad (4.13)$$

のように d 次元ベクトルに拡張することを考える．

任意の2点および2成分 $(f_i(\mathbf{x}), f_j(\mathbf{x}'))$ がガウス分布に従うものと拡張すると，ガウス過程の定義

$$\text{平均値 } m(\mathbf{x}) = E[f(\mathbf{x})] \quad (4.14)$$

$$\text{共分散 } k(\mathbf{x}, \mathbf{x}') = E[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))^t] \quad (4.15)$$

は，

$$\mathbf{m}(\mathbf{x}) = E[\mathbf{f}(\mathbf{x})] \quad (4.16)$$

$$\equiv (m_1(\mathbf{x}), \dots, m_d(\mathbf{x}))^t$$

$$k_{ij}(\mathbf{x}, \mathbf{x}') = \text{cov}[f_i(\mathbf{x}), f_j(\mathbf{x}')] \quad (4.17)$$

のように拡張される．なお共分散行列 \mathbf{K} は

$$\mathbf{K} = \begin{pmatrix} k_{11}(\mathbf{x}_1, \mathbf{x}_1) & k_{12}(\mathbf{x}_1, \mathbf{x}_1) & \cdots & k_{1d}(\mathbf{x}_1, \mathbf{x}_1) & k_{11}(\mathbf{x}_1, \mathbf{x}_2) & \cdots & k_{1d}(\mathbf{x}_1, \mathbf{x}_n) \\ k_{21}(\mathbf{x}_1, \mathbf{x}_1) & k_{22}(\mathbf{x}_1, \mathbf{x}_1) & \cdots & k_{2d}(\mathbf{x}_1, \mathbf{x}_1) & k_{21}(\mathbf{x}_1, \mathbf{x}_2) & \cdots & k_{2d}(\mathbf{x}_1, \mathbf{x}_n) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ k_{d1}(\mathbf{x}_1, \mathbf{x}_1) & k_{d2}(\mathbf{x}_1, \mathbf{x}_1) & \cdots & k_{dd}(\mathbf{x}_1, \mathbf{x}_1) & k_{d1}(\mathbf{x}_1, \mathbf{x}_2) & \cdots & k_{dd}(\mathbf{x}_1, \mathbf{x}_n) \\ k_{11}(\mathbf{x}_2, \mathbf{x}_1) & k_{12}(\mathbf{x}_2, \mathbf{x}_1) & \cdots & k_{1d}(\mathbf{x}_2, \mathbf{x}_1) & k_{11}(\mathbf{x}_2, \mathbf{x}_2) & \cdots & k_{1d}(\mathbf{x}_2, \mathbf{x}_n) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ k_{d1}(\mathbf{x}_n, \mathbf{x}_1) & k_{d2}(\mathbf{x}_n, \mathbf{x}_1) & \cdots & k_{dd}(\mathbf{x}_n, \mathbf{x}_1) & k_{d1}(\mathbf{x}_n, \mathbf{x}_2) & \cdots & k_{dd}(\mathbf{x}_n, \mathbf{x}_n) \end{pmatrix} \quad (4.18)$$

という, $(d \times n) \times (d \times n)$ の正方行列とする.

同時に, 確率変数ベクトル

$$\mathbf{f} = (f(x_1), \dots, f(x_n))^t \quad (4.19)$$

は,

$$\begin{aligned} \mathbf{f} = & (f_1(\mathbf{x}_1), f_2(\mathbf{x}_1), \dots, f_d(\mathbf{x}_1), \\ & f_1(\mathbf{x}_2), f_2(\mathbf{x}_2), \dots, f_d(\mathbf{x}_2), \\ & f_1(\mathbf{x}_n), f_2(\mathbf{x}_n), \dots, f_d(\mathbf{x}_n))^t \end{aligned} \quad (4.20)$$

のように $d \times n$ 次元ベクトルに拡張し, \mathbf{f}_+ についても同様に

$$\begin{aligned} \mathbf{f}_+ = & (f_1(\mathbf{x}_1), f_2(\mathbf{x}_1), \dots, f_d(\mathbf{x}_1) \\ & f_1(\mathbf{x}_2), f_2(\mathbf{x}_2), \dots, f_d(\mathbf{x}_2) \\ & f_1(\mathbf{x}_n), f_2(\mathbf{x}_n), \dots, f_d(\mathbf{x}_n) \\ & f_1(\mathbf{x}_*), f_2(\mathbf{x}_*), \dots, f_d(\mathbf{x}_*))^t \end{aligned} \quad (4.21)$$

のように $d \times (n + 1)$ 次元ベクトルに拡張する.

ここで, f_+ は結合ガウス分布 $\mathcal{N}(m_+, K_+)$ に従う. ただし

$$\begin{aligned} m_+ &= (m_1(\mathbf{x}_1), m_2(\mathbf{x}_1), \dots, m_d(\mathbf{x}_1), \\ &\quad m_1(\mathbf{x}_2), m_2(\mathbf{x}_2), \dots, m_d(\mathbf{x}_2), \\ &\quad m_1(\mathbf{x}_*), m_2(\mathbf{x}_*), \dots, m_d(\mathbf{x}_*))^t \end{aligned} \quad (4.22)$$

$$K_+ = \begin{pmatrix} K & \mathbf{k}_* \\ \mathbf{k}_*^t & \sigma_*^2 \end{pmatrix} \quad (4.23)$$

$$\mathbf{k}_* = \begin{pmatrix} k_{11}(\mathbf{x}_*, \mathbf{x}_1) & k_{12}(\mathbf{x}_*, \mathbf{x}_1) & \cdots & k_{1d}(\mathbf{x}_*, \mathbf{x}_1) \\ k_{21}(\mathbf{x}_*, \mathbf{x}_1) & k_{22}(\mathbf{x}_*, \mathbf{x}_1) & \cdots & k_{2d}(\mathbf{x}_*, \mathbf{x}_1) \\ \vdots & \vdots & \ddots & \vdots \\ k_{d1}(\mathbf{x}_*, \mathbf{x}_1) & k_{d2}(\mathbf{x}_*, \mathbf{x}_1) & \cdots & k_{dd}(\mathbf{x}_*, \mathbf{x}_1) \\ k_{11}(\mathbf{x}_*, \mathbf{x}_2) & k_{12}(\mathbf{x}_*, \mathbf{x}_2) & \cdots & k_{1d}(\mathbf{x}_*, \mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ k_{d1}(\mathbf{x}_*, \mathbf{x}_2) & k_{d2}(\mathbf{x}_*, \mathbf{x}_2) & \cdots & k_{dd}(\mathbf{x}_*, \mathbf{x}_2) \\ k_{11}(\mathbf{x}_*, \mathbf{x}_n) & k_{12}(\mathbf{x}_*, \mathbf{x}_n) & \cdots & k_{1d}(\mathbf{x}_*, \mathbf{x}_n) \\ \vdots & \vdots & \ddots & \vdots \\ k_{d1}(\mathbf{x}_*, \mathbf{x}_n) & k_{d2}(\mathbf{x}_*, \mathbf{x}_n) & \cdots & k_{dd}(\mathbf{x}_*, \mathbf{x}_n) \end{pmatrix} \quad (4.24)$$

$$\sigma_*^2 = \begin{pmatrix} k_{11}(\mathbf{x}_*, \mathbf{x}_*) & k_{12}(\mathbf{x}_*, \mathbf{x}_*) & \cdots & k_{1d}(\mathbf{x}_*, \mathbf{x}_*) \\ k_{21}(\mathbf{x}_*, \mathbf{x}_*) & k_{22}(\mathbf{x}_*, \mathbf{x}_*) & \cdots & k_{2d}(\mathbf{x}_*, \mathbf{x}_*) \\ \vdots & \vdots & \ddots & \vdots \\ k_{d1}(\mathbf{x}_*, \mathbf{x}_*) & k_{d2}(\mathbf{x}_*, \mathbf{x}_*) & \cdots & k_{dd}(\mathbf{x}_*, \mathbf{x}_*) \end{pmatrix} \quad (4.25)$$

である.

これを条件付き分布の平均と共分散の式

$$\boldsymbol{\mu}_{a|b} = \boldsymbol{\mu}_a + \Sigma_{ab} \Sigma_{bb}^{-1} (\mathbf{x}_b - \boldsymbol{\mu}_b) \quad (4.26)$$

$$\Sigma_{a|b} = \Sigma_{aa} - \Sigma_{ab} \Sigma_{bb}^{-1} \Sigma_{ba} \quad (4.27)$$

$$\boldsymbol{\mu} = (\boldsymbol{\mu}_a, \boldsymbol{\mu}_b)^t \quad (4.28)$$

$$\Sigma = \begin{pmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{pmatrix} \quad (4.29)$$

に当てはめると,

$$\bar{\mathbf{f}}_* = \mathbf{k}_*^t (K + \sigma_n^2 I)^{-1} \mathbf{f} \quad (4.30)$$

$$\text{var}[\mathbf{f}_*] = \sigma_*^2 - \mathbf{k}_*^t (K + \sigma_n^2 I)^{-1} \mathbf{k}_* \quad (4.31)$$

が得られる．ただし $m = 0$ とした．

4.4 評価実験

4.4.1 実験方法

エラーモデル e_c の同定，及びそのモデルを用いて補正した場合の入力精度評価を行う．

参加者にタッチパネル上に表示されるマーカをタップするタスクを与える．図 4.3 に実験の様子を，図 4.4 にタスク画面を示す．ランダムで，図 4.4(a) のように目的のマーカのみを表示したり，図 4.4(b) ~ 図 4.4(d) のように，周辺にマーカを 1 個 ~ 8 個表示する．ここで，中心に黒い印のある灰色の四角形が目的のマーカ，無印の灰色の四角形が追加のマーカである．目的のマーカをタップすると，そのマーカは消え，別の位置に次のマーカが表示される．参加者には追加のマーカをタップしないよう教示する．マーカのサイズは $3\text{mm} \times 3\text{mm}$ ，マーカ間の距離は 1mm である．

利き手の人差し指でタップし，もう一方の手でデバイスを保持する．マーカは計 1000 回表示する．タップのパラメータを記録し，考察する．デバイスは Nexus 4((株) Google, 筐体サイズ $133.9 \times 68.7 \times 9.1\text{ mm}$ ，画面サイズ 4.7 inch ，画面解像度 $1280\text{ pixel} \times 768\text{ pixel}$ 図 4.5) を使用する．

4.4.2 評価方法

インターフェイス形状に対するタップ位置のエラー

マーカの中心を目標とするタップ座標とする．目的のマーカのみを表示させた時の e を e_k とする．追加のマーカを表示させた時の e から e_k を除いたものを e_c とする．

入力補正精度の評価

提案手法によって得られたタップモデルを用いて入力データを補正し，入力精度の評価を行う．比較対象として，無補正，及びインターフェイス形状を考慮しないすな

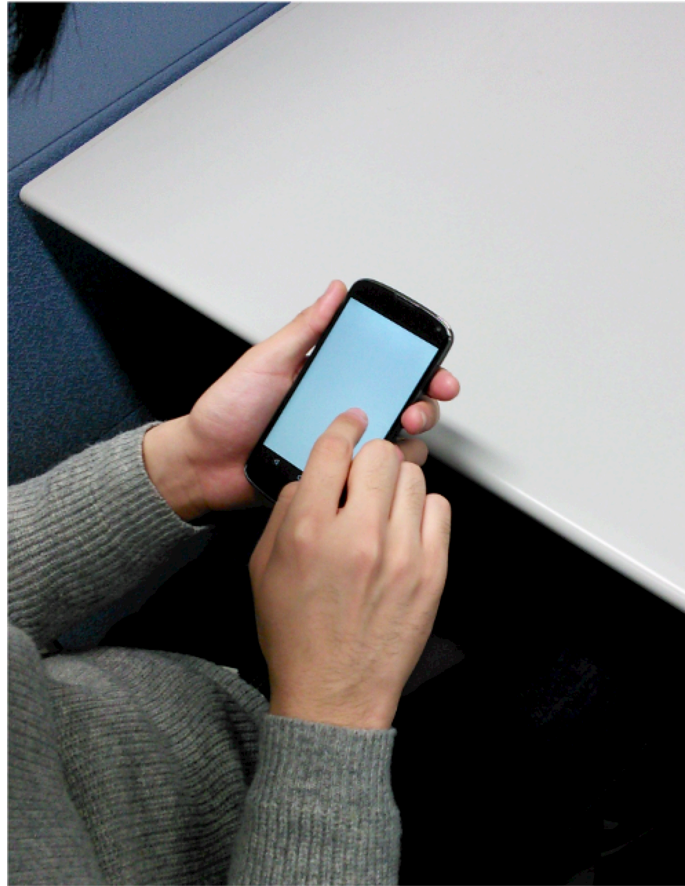


図 4.3: 実験環境の概観

わちキネマティックエラーだけのモデルを検討する。

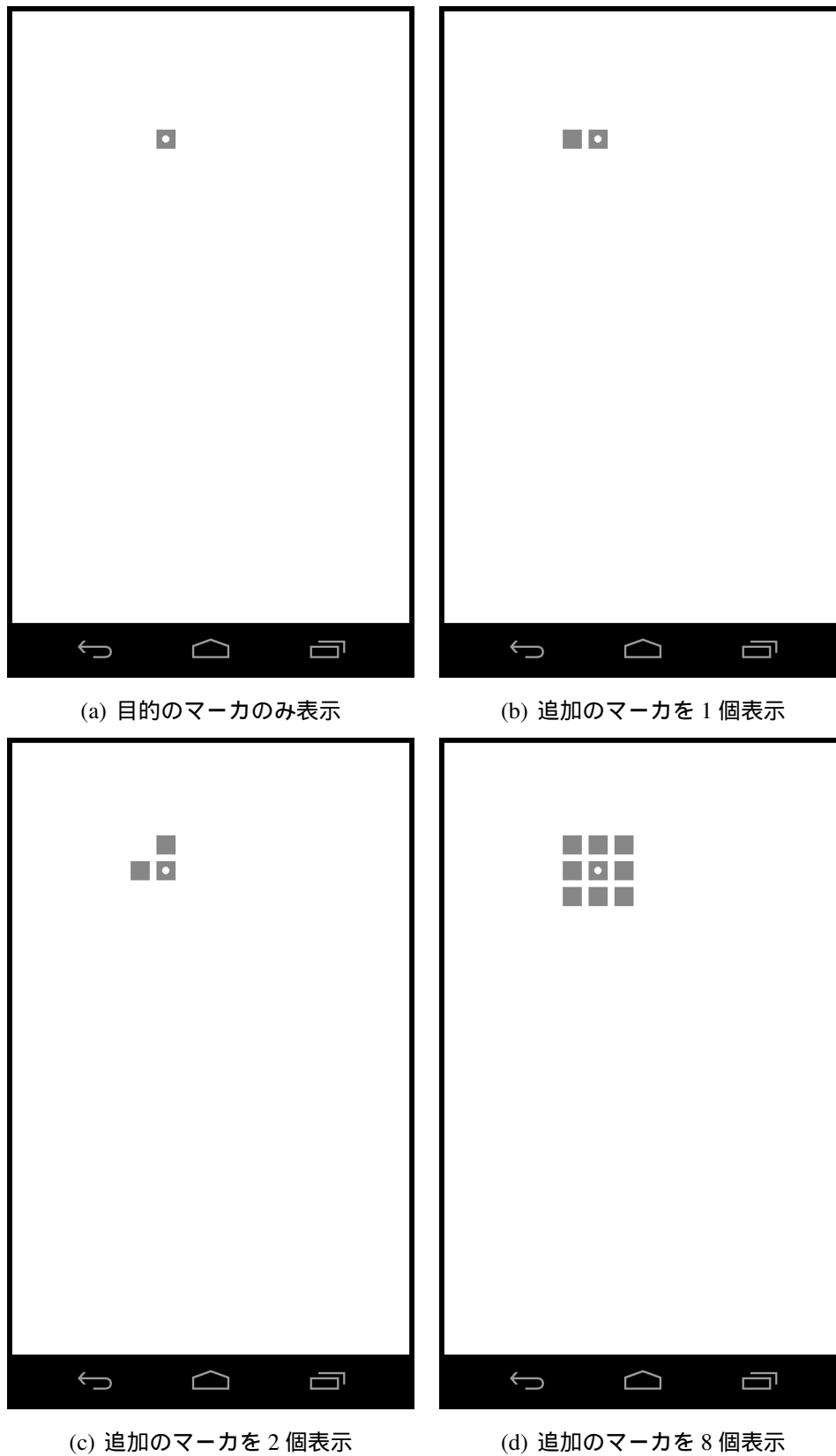


図 4.4: タスク画面

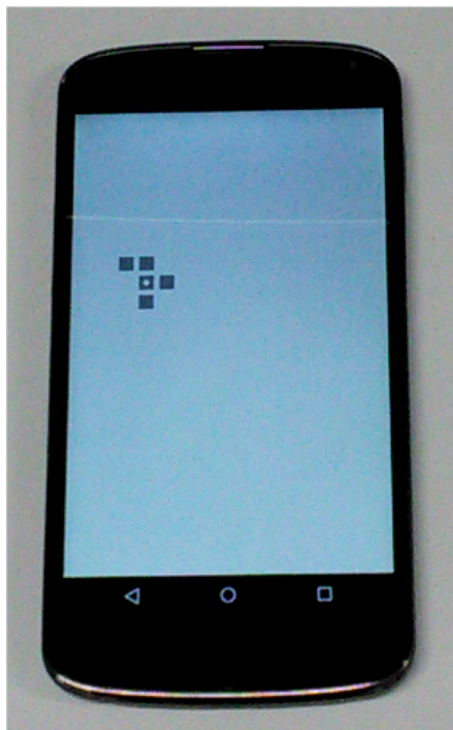


図 4.5: 実験に使用したスマートフォン : Nexus 4

7	6	5
4	×	3
2	1	0

図 4.6: 周辺マーカ番号

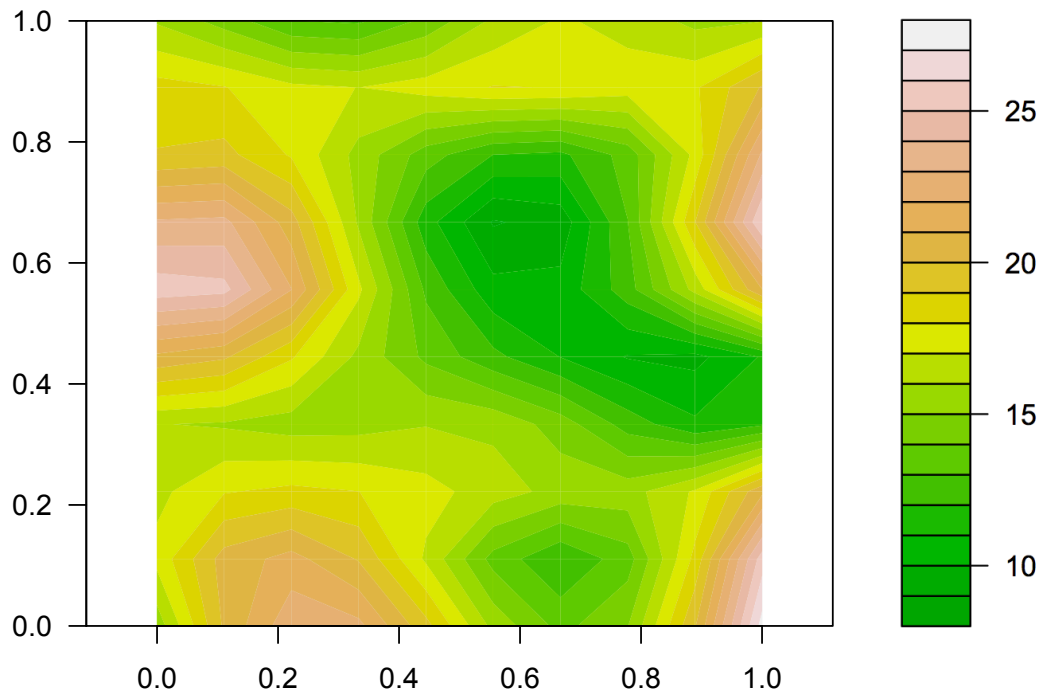
4.4.3 実験結果

図 4.7 に、目的のマーカのみを表示した時のタップ位置とエラー、及びその回帰結果を示す。横軸はタップ位置（図 4.7(a) では画面の左が 0，右が正，図 4.7(b) では画面の上が 0，下が正），縦軸はエラーを表す。

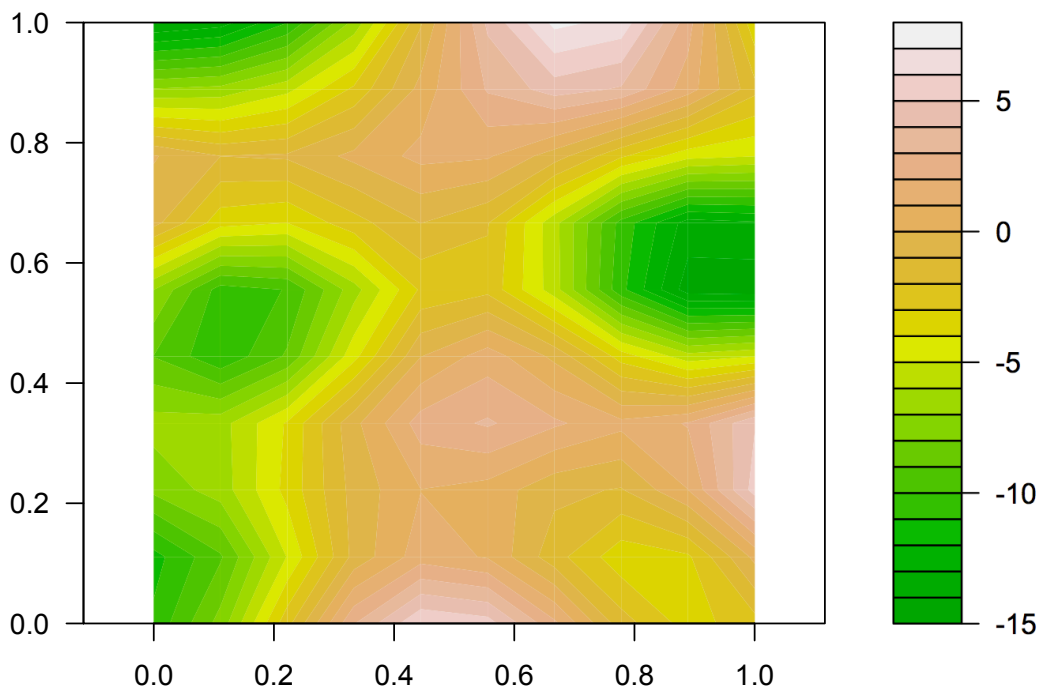
図 4.8 に、周辺マーカを 1 つ表示した時のエラー、及びその回帰結果を示す。横軸は追加のマーカを表示した位置（角度）であり，0 は右に，90 は上に，225 は左下に表示していることを意味する。図 4.9 に、周辺マーカの全表示パターンに対するタップ座標の誤差の平均値を示す。図 4.10，および表 4.1 に、周辺マーカの全表示パターンに対する回帰結果を示す。ただし，パターン値は以下のように定義する。周辺マーカに対し，表 4.4.3 の通り番号を付与する。更に，各番号を桁数とし，各位置にマーカを表示した場合を 1，表示しない場合を 0 とする，8 桁の 2 進数を定義する。これを 10 進数に直したものを，パターン値とした。すなわち，図 4.4 の各パターンに対するパターン値は，4.4(a)=0，4.4(b)=16，4.4(c)=80，4.4(d)=255 となる。

表 4.2 に、各条件におけるタップの成功率を示す。成功率は，タップ座標，あるいは推定されたタップ座標が，目標マーカの領域に含まれている場合成功，含まれていない場合失敗として，タップ回数に対する成功の割合とした。

目標マーカのみデータセット（上段）に対して e_k を適用する場合（2 列目），10-fold cross validation を適用し成功率を算出した。周辺マーカを表示したデータセット（中段及び下段）に対して e_k を適用する場合（2 列目及び 4，5 列目），目標マーカのみ表示した時のタップデータを訓練データとしてガウス過程回帰を用いてモデルを構築し，周辺マーカを表示した時のタップデータに適用した。周辺マーカを表示したデータセット（中段及び下段）に対して e_c を適用する場合（3，4，5 列目），10-fold cross validation を適用し，成功率を算出した。また， e_c のモデル化はガウス過程回帰を用いて行った



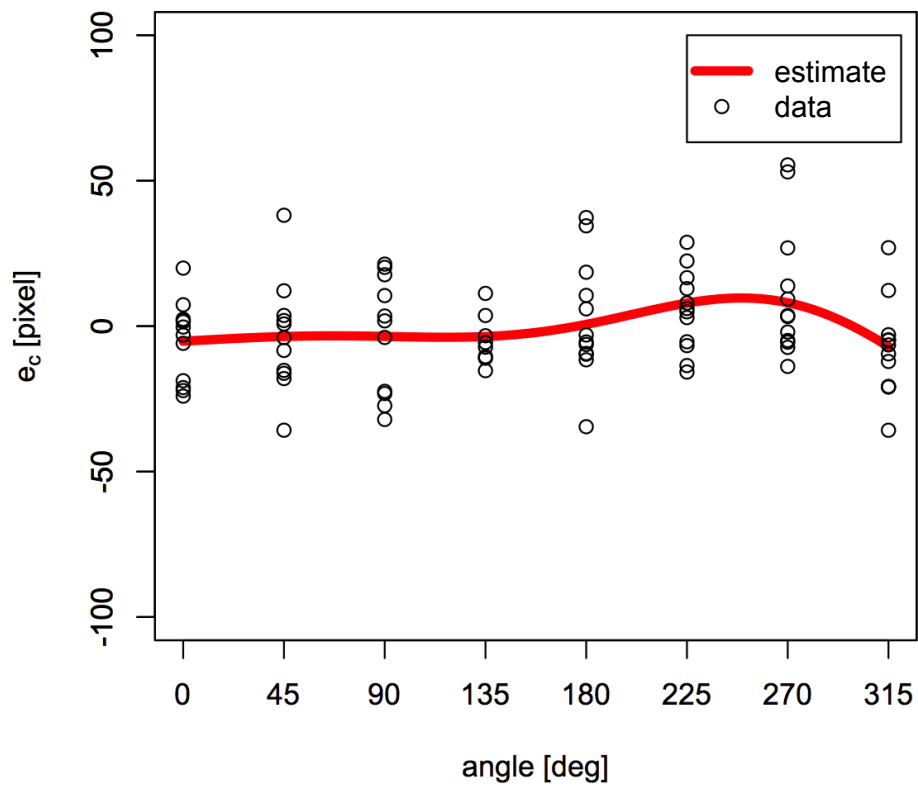
(a) 水平方向



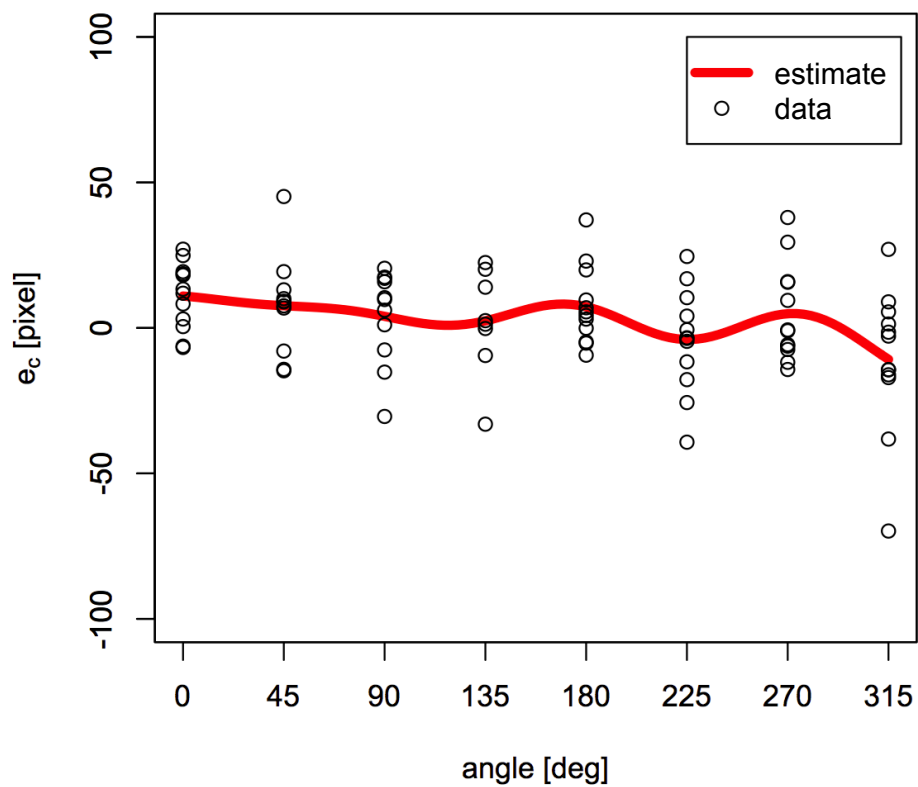
(b) 垂直方向

図 4.7: タップ位置とキネマティックエラーの関係

が、回帰せず、各パターンに対する座標誤差の平均値を減算により取り除く方法も行った(5列目)。



(a) 水平方向



(b) 垂直方向

図 4.8: マーカの表示位置と認知的なエラーの関係 (周辺マーカが1つのみの場合)

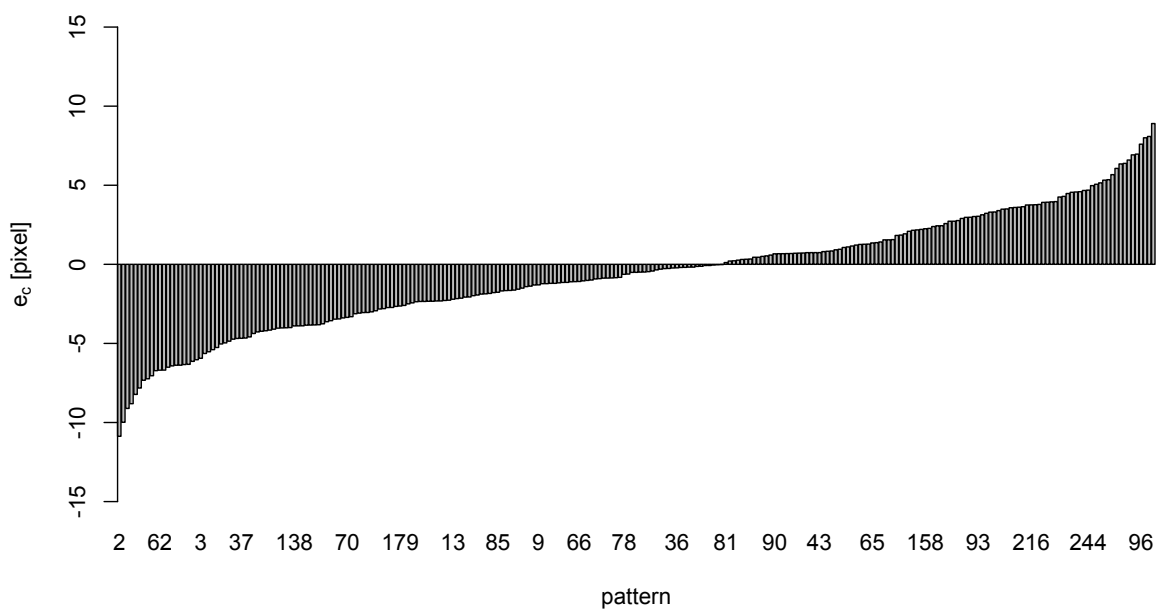
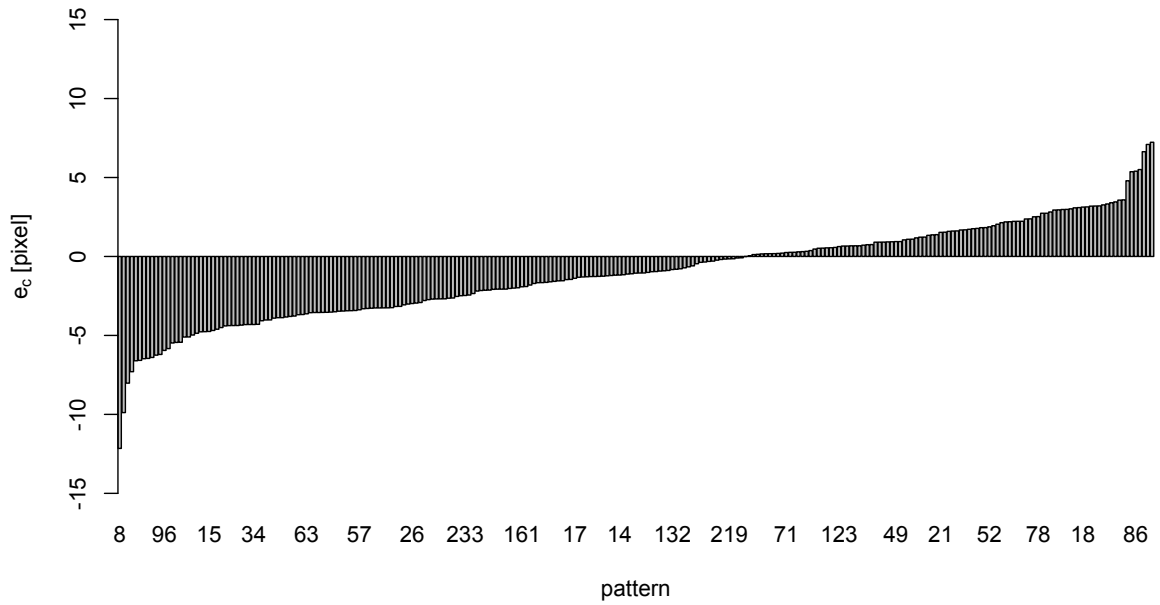
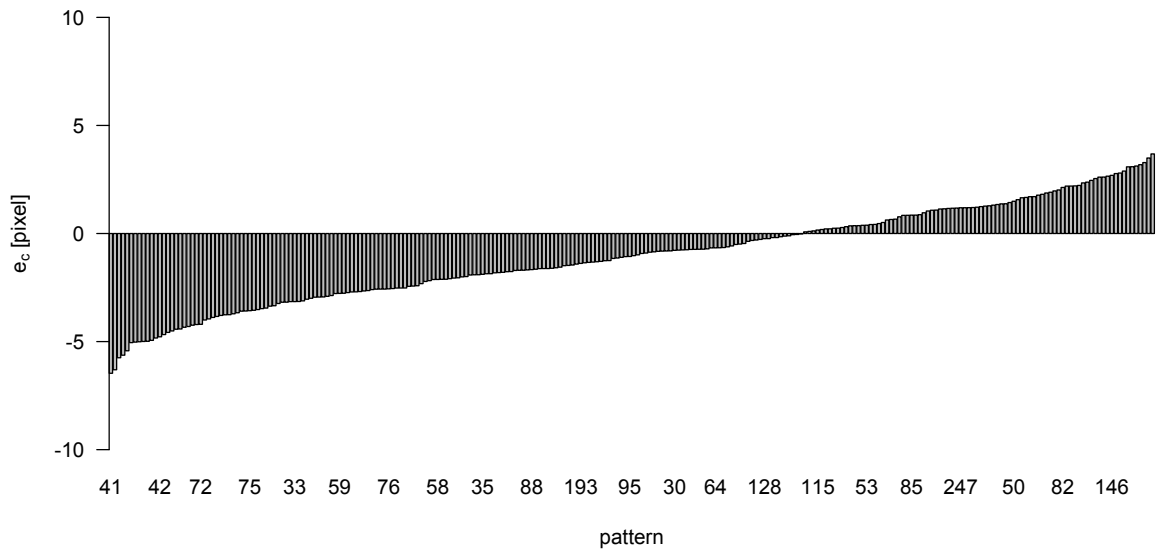
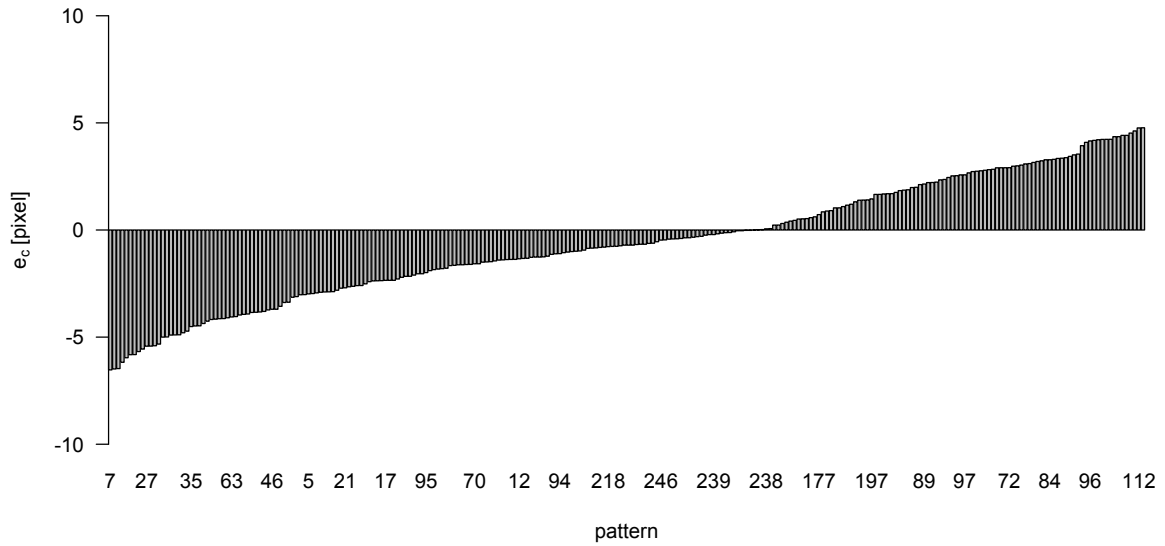


図 4.9: 周辺マーカパターンと認知的なエラーの関係 (全ての周辺マーカパターンに対するタップ座標誤差の平均)



(a) 水平方向



(b) 垂直方向

図 4.10: 周辺マーカパターンと認知的なエラーの関係 (全ての周辺マーカパターンに対する回帰結果)

表 4.1: 周辺マーカパターンと認知的なエラーの関係

パターン	エラー		パターン	エラー		パターン	エラー	
	横方向	縦方向		横方向	縦方向		横方向	縦方向
1	-3.04	-2.89	44	-3.84	0.53	87	1.67	-2.94
2	-0.76	-5.67	45	-5.03	-0.84	88	-1.67	3.54
3	-1.62	-6.49	46	-3.15	-3.7	89	-2.95	2.15
4	0.21	-1.06	47	-3.71	-4.26	90	-1.1	-0.99
5	-1.62	-2.98	48	1.14	1.84	91	-1.76	-1.58
6	0.85	-5.42	49	-0.73	-0.15	92	-0.46	3.38
7	-0.36	-6.53	50	1.49	-2.97	93	-2.09	1.68
8	-5.43	-1.44	51	0.24	-4.14	94	-0.05	-1.1
9	-6.3	-2.51	52	2.61	1.86	95	-1.06	-1.99
10	-4.42	-5.55	53	0.39	-0.43	96	-1.36	4.16
11	-4.68	-5.82	54	2.8	-2.89	97	-2.76	2.57
12	-3.76	-1.34	55	1.2	-4.36	98	-0.86	-0.63
13	-4.99	-2.71	56	-2.69	1.4	99	-1.65	-1.4
14	-2.91	-5.4	57	-3.95	-0.03	100	0.51	4.42
15	-3.52	-5.96	58	-2.13	-3.11	101	-1.25	2.54
16	1.2	-0.36	59	-2.77	-3.73	102	0.84	-0.31
17	-0.71	-2.35	60	-1.32	1.31	103	-0.3	-1.38
18	1.71	-5	61	-2.94	-0.42	104	-4.94	3.94
19	0.42	-6.17	62	-0.92	-3.14	105	-5.75	2.91
20	2.54	-0.41	63	-1.92	-4.06	106	-4.24	-0.55
21	0.28	-2.7	64	-0.66	2.79	107	-4.43	-0.77
22	2.89	-4.99	65	-2.12	1.2	108	-3.18	4.09
23	1.24	-6.47	66	-0.02	-1.82	109	-4.34	2.76
24	-2.58	-0.46	67	-0.85	-2.6	110	-2.64	-0.34
25	-3.88	-1.9	68	1.08	2.98	111	-3.18	-0.86
26	-1.86	-4.8	69	-0.73	1.09	112	1.18	4.76
27	-2.55	-5.42	70	1.57	-1.58	113	-0.66	2.82
28	-1.34	-0.62	71	0.38	-2.65	114	1.38	-0.24
29	-3	-2.35	72	-4.2	2.91	115	0.16	-1.37
30	-0.78	-4.9	73	-5.05	1.88	116	2.61	4.77
31	-1.82	-5.82	74	-3.34	-1.4	117	0.42	2.53
32	-1.71	0.88	75	-3.57	-1.62	118	2.65	-0.18
33	-3.15	-0.74	76	-2.56	2.99	119	1.08	-1.61
34	-1.07	-3.7	77	-3.76	1.66	120	-2.32	4.62
35	-1.89	-4.51	78	-1.86	-1.26	121	-3.55	3.23
36	0.19	1.16	79	-2.44	-1.78	122	-1.91	-0.08
37	-1.61	-0.76	80	1.77	3.34	123	-2.52	-0.67
38	0.67	-3.37	81	-0.11	1.39	124	-0.99	4.52
39	-0.51	-4.48	82	2.13	-1.49	125	-2.57	2.83
40	-5.63	0.36	83	0.87	-2.63	126	-0.74	-0.13
41	-6.47	-0.71	84	3.09	3.28	127	-1.7	-1.01
42	-4.78	-3.92	85	0.85	1.03	128	-0.24	0
43	-4.99	-4.18	86	3.28	-1.5	129	-1.42	-1.37

パターン	エラー		パターン	エラー		パターン	エラー	
	横方向	縦方向		横方向	縦方向		横方向	縦方向
130	0.77	-4.16	172	-2.7	1.69	214	3.19	-1.63
131	0.22	-4.72	173	-3.59	0.58	215	1.88	-2.81
132	1.34	0.07	174	-1.8	-2.29	216	-1.68	3.5
133	-0.19	-1.61	175	-2.06	-2.59	217	-2.66	2.36
134	2.19	-4.04	176	1.81	2.45	218	-0.9	-0.78
135	1.28	-4.9	177	0.25	0.72	219	-1.26	-1.11
136	-4	0.02	178	2.38	-2.1	220	-0.67	3.2
137	-4.58	-0.8	179	1.43	-3.02	221	-1.99	1.76
138	-2.78	-3.84	180	3.09	2.34	222	-0.04	-1.03
139	-2.73	-3.85	181	1.17	0.3	223	-0.75	-1.66
140	-2.52	-0.02	182	3.49	-2.16	224	-1.02	4.23
141	-3.45	-1.14	183	2.2	-3.38	225	-2.13	2.91
142	-1.46	-3.83	184	-1.91	2.21	226	-0.32	-0.3
143	-1.77	-4.14	185	-2.87	1.03	227	-0.8	-0.81
144	1.97	0.42	186	-1.13	-2.04	228	0.65	4.36
145	0.36	-1.32	187	-1.48	-2.41	229	-0.81	2.73
146	2.69	-3.97	188	-0.73	1.99	230	1.19	-0.12
147	1.7	-4.89	189	-2.05	0.51	231	0.36	-0.94
148	3.12	0.23	190	-0.12	-2.21	232	-4.51	4.21
149	1.16	-1.81	191	-0.82	-2.88	233	-5.01	3.44
150	3.68	-4.1	192	-0.24	3.03	234	-3.59	-0.02
151	2.34	-5.32	193	-1.38	1.7	235	-3.48	0.01
152	-1.7	0.52	194	0.63	-1.33	236	-2.93	4.23
153	-2.7	-0.66	195	0.1	-1.85	237	-3.79	3.15
154	-0.77	-3.56	196	1.31	3.08	238	-2.18	0.05
155	-1.15	-3.94	197	-0.19	1.45	239	-2.42	-0.21
156	-0.65	0.23	198	2.02	-1.22	240	0.96	4.36
157	-2.01	-1.25	199	1.13	-2.04	241	-0.57	2.66
158	0.12	-3.8	200	-3.67	3.35	242	1.37	-0.4
159	-0.62	-4.47	201	-4.21	2.57	243	0.46	-1.28
160	-0.49	1.98	202	-2.59	-0.71	244	2.2	4.23
161	-1.62	0.61	203	-2.52	-0.68	245	0.32	2.24
162	0.37	-2.35	204	-2.22	3.3	246	2.46	-0.47
163	-0.15	-2.91	205	-3.12	2.22	247	1.19	-1.65
164	1.22	2.12	206	-1.31	-0.71	248	-2.43	4.42
165	-0.27	0.45	207	-1.59	-0.98	249	-3.36	3.28
166	1.91	-2.16	208	1.65	3.1	250	-1.81	-0.04
167	1.04	-3.02	209	0.08	1.4	251	-2.12	-0.37
168	-4.3	1.66	210	2.22	-1.48	252	-1.29	4.18
169	-4.84	0.84	211	1.27	-2.37	253	-2.57	2.74
170	-3.24	-2.37	212	2.77	2.9	254	-0.83	-0.22
171	-3.15	-2.38	213	0.84	0.9	255	-1.49	-0.85

表 4.2: タップの成功率

表示するマーカ	無補正	e_k のみ	e_c のみ	$e_k + e_c$	$e_k + e_c$ (平均値)
目標マーカのみ	39.3%	70.8%	-	-	-
周辺マーカ1つ	42.9%	68.3%	62.5%	69.7%	-
周辺マーカ全パターン	-	73.6%	-	74.7%	75.5%

4.5 考察

4.5.1 得られたエラーモデル

キネマティックエラーについて，操作する手の方向にタップ位置が移動し，操作する手から遠い位置ではエラーが大きいことがわかる．これは先行研究と同じ傾向である．また，エラーの大きさは最大で約 28[pixel] であった．これは画面上で約 2.5[mm] であり，マーカ 1 つ分程度のエラーが生じていることがわかった．

認知的なエラーについて，周辺マーカを 1 つのみ表示した時の結果から，周辺マーカが左下にあるとタップ位置は右上に移動し，同じくマーカが右下にあると上に，右にあると左下に移動することがわかる．これらの結果は，追加のマーカを避ける方向にタップ位置が移動するという予想を裏付けている．

周辺マーカ全パターンに対するエラーの結果から，水平方向のエラーについて，パターン値が 8 の倍数のときタップ位置は左方向に移動し，16 の倍数のときタップ位置は右方向に移動していることがわかる．すなわち，周辺マーカが目標マーカの左右にあるとき，周辺マーカから離れる方向にタップ位置が移動している．また，パターン値が 20 や 56 など，周辺マーカが目標マーカの左側に複数表示されている場合，タップ位置はより大きく右方向に移動している．垂直方向のエラーについても同様に，パターン値が 116 や 120 など，周辺マーカが目標マーカの上に多い時，タップ位置は大きく下方向に移動している．更に，各パターンに対するエラーの平均値を見ると，エラーは最大で約 11[pixel] であった．これは画面上で約 1[mm] であり，マーカ間距離とほぼ同等，かつキネマティックエラーの約 1/3 程度のエラーが生じていることがわかった．

4.5.2 タップ精度

無補正の場合，周辺マーカが 1 つ表示された場合の方が，目標マーカのみ表示された場合より，成功率が高いことがわかった．実験参加者に対するヒアリングでは，周辺マーカがある方が上手くタップ出来たという回答が複数あり，これと一致したと言える．これは，周辺マーカを表示することで慎重にタップしようとする効果の他，目

標マーカだけではマーカが指で隠れてしまうのに対し、周辺マーカを表示することにより目印となる効果があると考えられる。

目標マーカのみ表示した場合において、キネマティックエラーモデルにより入力座標の補正を行った結果、入力精度は80.2%(31.5ポイント)向上した。なお、精度の向上率は以下の式により算出した。

$$\frac{(\text{提案法の精度}) - (\text{従来法の精度})}{(\text{従来法の精度})} \times 100\% \quad (4.32)$$

これは先行研究と似た傾向であるが、より大幅に精度を向上することに成功しており、キネマティックエラーモデルによる入力補正は十分に機能していると言える。先行研究と精度が異なる理由として、操作姿勢の違いが考えられる。先行研究ではデバイスを両手で横長に保持し、両親指で操作しているのに対し、本研究では片手で縦長に保持し、もう一方の手の人差し指で操作している。よって精度について単純に先行研究と比較することは出来ない。

一方、周辺マーカを1つ表示した場合、同じモデルを用いて入力座標の補正を行った結果、入力精度の向上率は59.2%(25.4ポイント)に留まり、入力精度自体も目標マーカのみ表示した場合と比べて2.5ポイント低下した。これは補正に使用したモデルが人間の行動特性を十分に表せていないことが原因と考えられる。

周辺マーカを1つ表示した場合において、認知的なエラーモデルにより入力座標の補正を行った結果、入力精度は45.7%(19.6ポイント)向上した。キネマティックエラーモデルによる補正に比べ精度の向上率は低いが、認知的なエラーモデルがタップ座標のエラーを表していることが示唆された。

さらに、周辺マーカを1つ表示した場合について、両エラーモデルを組み合わせた場合、キネマティックエラーモデルのみ用いた場合に比べて2.0%(1.4ポイント)向上した。全周辺マーカパターンを表示した場合についても、両エラーモデルを組み合わせた場合、キネマティックエラーモデルのみを用いた場合に比べて1.5%(1.1ポイント)向上し、1標本の t 検定を行ったところ、有意差が認められた($t = 26.8923, p < 0.05$)。いずれも2%程度と小さな向上だが、認知的なエラーモデルを用いることが精度を向上

する効果があることが示唆されたと言える。

また、ガウス過程回帰を用いず、座標誤差の平均値を減算により取り除いた場合、キネマティックエラーモデルのみを用いた場合に比べて2.6% (1.9ポイント)向上し、1標本の t 検定を行ったところ、有意差が認められた($t = 5.51315, p < 0.01$)。ガウス過程回帰を用いた場合に比べて精度が上がった理由として、平均値を用いることは、回帰を行った場合を考えると、実験を行ったパターンに対してオーバーフィッティングしていると思えることができるため、精度が上がったことが考えられる。

第5章 結論

本研究では、ヒューマンコンピュータインタラクションにおける新たな非明示的情報として、デスクワーク中のユーザが机の上に与える圧力の繁閑による変化とタッチパネル操作時のUI形状が人の操作特性に与える影響をモデル化した。

机にかかる圧力について、タイピングタスクとマウス操作タスクの両タスクについて、ユーザの割り込み可能性推定を行った。タイピングおよびマウス操作タスク中のユーザに対し、机にかかる圧力から割り込み可能性を推定する参加者実験を行った結果、タイピングタスクにおいて77%の精度で推定に成功し、机にかかる圧力がユーザ状態推定に有用な情報であることを示した。今後、タイピングタスクなどを行っていない場合における状態推定の可能性を示すことにより、従来法に対する提案手法の優位性が示されると考えられる。

タッチパネル操作時のUI形状が人の操作特性に与える影響について、ボタン配置によるタップ位置の差をモデル化し、そのモデルを用いてタップ位置の補正を行った。様々なボタン配置の中から指定されたボタンをタップする参加者実験を行ったところ、ボタン配置によってタップ位置が変化することが確認された。また、適切なモデルを用いて補正した場合、入力精度は30%程度改善するが、異なるボタン配置に対するモデルを適用した場合改善率が低下することが確認された。これにより、提案手法によって改善率が向上することが確認された。一方、改善率は2%程度にとどまった。他の回帰を用いる、あるいは他の入力変数を用いるなど、モデル構築について検討することにより、改善率向上が期待できる。

以上により、提案した2件の非明示的情報が、コンピュータがユーザの意図を把握するために有用であることを示した。

謝辞

本論文は、様々な方のお力添えのもとに完成いたしました。はじめに、本論文審査委員会の主査であり、主任指導教官でもありました山田誠二教授におかれましては、本研究全般に関して多大なご指導とご鞭撻を賜りました。本研究をまとめるにあたり幾度となく激励していただくとともに我慢強くご支援くださったことなど、先生のご指導無くしては本研究は成し得ないものでありました。心より御礼申し上げます。

本論文をまとめるにあたり、論文審査委員である藤田欣也教授、相原健郎准教授、市瀬龍太郎准教授、稲邑哲也准教授からも多大なご指導とご鞭撻を賜りました。皆様のご指導、ご鞭撻により本論文をまとめ、より質の高いものとすることができました。重ねて御礼申し上げます。

電力中央研究所の鶴見剛也様には、本研究第三章で使用したガウス過程回帰について解説、および多次元拡張についてご助言いただきました。心より御礼申し上げます。

山田誠二研究室のみなさまにも大変お世話になりました。研究室ミーティングで有益なコメントを頂きましたことを感謝いたします。

最後に、本研究の遂行と本論文の作成に対して、いつも心の支えとなり元気づけてくれた両親に感謝します。

参考文献

- [Al Faraj 09] Al Faraj, K., Mojahid, M., and Vigouroux, N.: BigKey: A Virtual Keyboard for Mobile Devices, in *Proceedings of the 13th International Conference on Human-Computer Interaction. Part III: Ubiquitous and Intelligent Interaction*, pp. 3–10 (2009)
- [Aravena 12] Aravena, P., Delevoye-Turrell, Y., Deprez, V., Cheylus, A., Paulignan, Y., Frak, V., and Nazir, T.: Grip Force Reveals the Context Sensitivity of Language-Induced Motor Activity during “ Action Words ” Processing: Evidence from Sentential Negation, *PLoS ONE*, Vol. 7, No. 12, p. e50287 (2012)
- [Bailey 01] Bailey, B. P., Konstan, J. A., and Carlis, J. V.: The Effects of Interruptions on Task Performance, Annoyance, and Anxiety in the User Interface, in *Proceedings of INTERACT’01*, pp. 593–601 (2001)
- [Bouckaert 10] Bouckaert, R. R., Frank, E., Hall, M., Kirkby, R., Reutemann, P., See-wald, A., and Scuse, D.: WEKA Manual for Version 3-6-4 (2010)
- [Breiman 01] Breiman, L.: Random Forests, *Machine Learning*, Vol. 45, No. 1, pp. 5–32 (2001)
- [Chen 07] Chen, D., Hart, J., and Vertegaal, R.: Towards a physiological model of user interruptibility, in *Proceedings of the 11th IFIP TC 13 international conference on Human-computer interaction - Volume Part II*, INTERACT’07, pp. 439–451 (2007)

- [Drummond 03] Drummond, C. and Holte, R.: C4.5 and class imbalance and cost sensitivity: why under-sampling beats over-sampling, in *Workshop on Learning from Imbalanced Datasets*, pp. 1–8 (2003)
- [Epp 11] Epp, C., Lippold, M., and Mandryk, R.: Identifying Emotional States Using Keystroke Dynamics, in *Proceedings of the 2011 Annual Conference on Human Factors in Computing Systems (CHI 2011)*, pp. 715–724 (2011)
- [Findlater 12] Findlater, L. and Wobbrock, J.: Personalized input: improving ten-finger touchscreen typing through automatic adaptation, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '12*, pp. 815–824, ACM (2012)
- [Fogarty 05] Fogarty, J., Hudson, S. E., Atkeson, C. G., Avrahami, D., Forlizzi, J., Kiesler, S., Lee, J. C., and Yang, J.: Predicting human interruptibility with sensors, *ACM Transactions on Computer-Human Interaction*, Vol. 12, No. 1, pp. 119–146 (2005)
- [Goel 13] Goel, M., Jansen, A., Mandel, T., Patel, S. N., and Wobbrock, J. O.: ContextType: using hand posture information to improve mobile touch screen text entry, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '13*, pp. 2795–2798, ACM (2013)
- [Goodman 02] Goodman, J., Venolia, G., Steury, K., and Parker, C.: Language modeling for soft keyboards, in *Proceedings of the 7th international conference on Intelligent user interfaces, IUI '02*, pp. 194–195, ACM (2002)
- [Gunawardana 10] Gunawardana, A., Paek, T., and Meek, C.: Usability guided key-target resizing for soft keyboards, in *Proceedings of the 15th international conference on Intelligent user interfaces, IUI '10*, pp. 111–118, ACM (2010)
- [萩谷 12] 萩谷 俊幸, 上向 俊晃, 加藤 恒夫: 確率モデルに基づくキーボード入力方式, 第 74 回全国大会講演論文集, 第 2012 巻, pp. 13–15 (2012)

- [Himberg 03] Himberg, J., Häkkinen, J., Kangas, P., and Mäntyjärvi, J.: On-line personalization of a touch screen based keyboard, in *Proceedings of the 8th international conference on Intelligent user interfaces*, IUI '03, pp. 77–84, ACM (2003)
- [Holz 11] Holz, C. and Baudisch, P.: Understanding touch, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pp. 2501–2510, ACM (2011)
- [本田 98] 本田 新九郎, 富岡 展也, 木村 尚亮, 大澤 隆治, 岡田 謙一, 松下 温: 作業者の集中度に応じた在宅勤務環境の提供: 仮想オフィスシステム, 情報処理学会論文誌, Vol. 39, No. 5, pp. 1472–1483 (1998)
- [Iqbal 08] Iqbal, S. T. and Bailey, B. P.: Effects of intelligent notification management on users and their tasks, in *Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pp. 93–102 (2008)
- [Iqbal 10] Iqbal, S. T. and Bailey, B. P.: Oasis: A framework for linking notification delivery to the perceptual structure of goal-directed tasks, *ACM Transactions on Computer-Human Interaction*, Vol. 17, pp. 1–28 (2010)
- [Jaimes 06] Jaimes, A.: Posture and activity silhouettes for self-reporting, interruption management, and attentive interfaces, in *Proceedings of the 11th international conference on Intelligent user interfaces*, IUI '06, pp. 24–31 (2006)
- [Kim 10] Kim, T., Hong, H., and Magerko, B.: Design requirements for ambient display that supports sustainable lifestyle, in *Proceedings of the 8th ACM Conference on Designing Interactive Systems*, DIS '10, pp. 103–112, New York, NY, USA (2010)
- [木村 11] 木村 和行, 田中 貴紘, 藤田 欣也: 頭部の前後・回転運動に着目したデスクワーク中の割り込み拒否度と頭部運動の関係の分析, 情報処理学会論文誌, Vol. 52, No. 4, pp. 1485–1494 (2011)

- [Klami 08] Klami, A., Saunders, C., Campos, de T. E., and Kaski, S.: Can Relevance of Images Be Inferred from Eye Movements?, in *Proceedings of the 1st ACM International Conference on Multimedia Information Retrieval, MIR '08*, pp. 134–140, New York, NY, USA (2008), ACM
- [McCrickard 01] McCrickard, D. S., Catrambone, R., and Stasko, J. T.: Evaluating Animation in the Periphery as a Mechanism for Maintaining Awareness, in *Proceedings of the IFIP TC.13 Conference on Human-Computer Interaction (INTERACT'01)*, pp. 148–156 (2001)
- [水口 04] 水口 充, 竹内 友則, 倉本 到, 渋谷 雄, 辻野 嘉宏 : デスクワークにおける忙しさの自動推定, ヒューマンインタフェース学会論文誌, Vol. 6, No. 1, pp. 69–74 (2004)
- [Pickering 86] Pickering, J. A.: Touch-sensitive screens: the technologies and their application, *Int. J. Man-Mach. Stud.*, Vol. 25, No. 3, pp. 249–269 (1986)
- [Quinlan 86] Quinlan, J. R.: Induction of Decision Trees, *Mach. Learn.*, pp. 81–106 (1986)
- [Quinlan 93] Quinlan, J. R.: *C4.5: programs for machine learning*, Morgan Kaufmann Publishers Inc. (1993)
- [Rasmussen 05] Rasmussen, C. E. and Williams, C. K. I.: *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*, The MIT Press (2005)
- [Rudchenko 11] Rudchenko, D., Paek, T., and Badger, E.: Text text revolution: a game that improves text entry on mobile touchscreen keyboards, in *Proceedings of the 9th international conference on Pervasive computing, Pervasive'11*, pp. 206–213, Springer-Verlag (2011)
- [Siek 05] Siek, K. A., Rogers, Y., and Connelly, K. H.: Fat finger worries: how older and younger users physically interact with PDAs, in *Proceedings of the 2005 IFIP TC13*

- international conference on Human-Computer Interaction*, INTERACT'05, pp. 267–280, Springer-Verlag (2005)
- [田中 10] 田中 貴紘, 松村 京平, 藤田 欣也 : 利用アプリケーション切り替え時に着目したユーザの割り込み拒否度推定法の検討, *人工知能学会論文誌*, Vol. 25, No. 6, pp. 683–693 (2010)
- [Tripathi 08] Tripathi, P. and Panchanathan, S.: Implication of Multimodality in Ambient Interfaces, in *Proceedings of the 2008 Ambi-Sys Workshop on Haptic User Interfaces in Ambient Media Systems*, HAS '08, pp. 10:1–10:10, ICST, Brussels, Belgium, Belgium (2008), ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering)
- [Vapnik 95] Vapnik, V. N.: *The nature of statistical learning theory*, Springer-Verlag New York, Inc. (1995)
- [Weir 12] Weir, D., Rogers, S., Murray-Smith, R., and Löchtfeld, M.: A user-specific machine learning approach for improving touch accuracy on mobile devices, in *Proceedings of the 25th annual ACM symposium on User interface software and technology*, UIST '12, pp. 465–476, ACM (2012)

研究業績

学術論文

1. 谷 堯尚, 山田 誠二 : 机上にかかる圧力を用いたユーザの割り込み可能性推定, 人工知能学会論文誌, Vol.29, No.1 (2013)

国際会議 (査読有り)

1. Takahisa Tani, and Seiji Yamada : Estimating user interruptibility by measuring tabletop pressure, in *Proceedings of on 1st international conference on human-agent interaction*, II-1-1, 4pages (2013)
2. Takahisa Tani, and Seiji Yamada : Estimating user interruptibility by measuring tabletop pressure, in *Proceedings of on Human Factors in Computing Systems, CHI'13, Extended Abstracts*, pp.1707-1712 (2013)
3. Takahisa Tani, and Seiji Yamada : User state identification through desktop interaction, in *Proceedings of The 6th International Conference on Soft Computing and Intelligent Systems*, pp.395-398 (2012)
4. Takahisa Tani, and Seiji Yamada : Applying key typing pressure to estimate a user's state of activity, in *Proceedings of The 21nd IEEE International Symposium on Robot and Human Interactive Communication*, pp.185-190 (2012)
5. Takahisa Tani, and Seiji Yamada : Tap Model to Improve Input Accuracy of Touch Panels, in *Proceedings of 2013 Conference on Technologies and Applications of Artificial Intelligence*, pp. 265-267 (2013)

国内会議 (査読無し)

1. 谷 堯尚, 山田 誠二 : 机上にかかる圧力を利用したユーザ状態推定, 第 27 回 人工知能学会 全国大会, 2F4-OS-04-7, 4 pages (2013)

2. 谷 堯尚, 山田 誠二: 机上にかかる圧力のユーザ状態推定への利用, HAI シンポジウム 2012, 3B-2, 6 pages (2012)
3. 谷 堯尚, 山田 誠二: 机上にかかる圧力を用いたユーザ状態の推定, 第 26 回 人工知能学会 全国大会, 3O1-OS-3a-4, 4 pages (2012)
4. 谷 堯尚, 山田 誠二: 机上にかかる圧力を用いたユーザ状態の推定, HAI シンポジウム 2011, 2B-4-S, 2 pages (2011)
5. 谷 堯尚, 山田 誠二: タッチパネルにおける UI デザインを考慮した操作特性モデル, HAI シンポジウム 2013, P20, 4 pages (2013)