

平成 26 年度 修士学位論文

題目 モバイル自己抑制のための
 階段コンテキスト認識に関する研究

指導教員 服部 峻

提出者 室蘭工業大学大学院 情報電子工学系専攻

氏 名 吉田 京平

学籍番号 13054053

提出年月日 平成 27 年 1 月 30 日

目次

第 1 章	序論	1
第 2 章	関連研究	4
2.1	コンテキスト・アウェアネスに関する研究	4
2.2	階段コンテキストに関する研究	5
2.3	屋内測位システムに関する研究	7
2.4	関連研究まとめ	8
第 3 章	提案手法	11
3.1	加速度センサを用いたパラメータ変動手法	11
3.2	加速度センサ +Wi-Fi を用いた手法	17
第 4 章	評価実験	26
4.1	加速度センサを用いたパラメータ変動手法に関する実験	26
4.2	加速度センサ +Wi-Fi を用いた手法に関する実験	47
第 5 章	結論	63
	謝辞	66
	参考文献	67

目次

1.1	東京消防庁管内における年別の救急搬送人員	2
1.2	事故が起きた建物の場所と事故の種類の関係	2
1.3	階段における歩きスマホの自粛を促す様子 (イメージ)	3
2.1	モバイル自己抑制アプリケーションに関する要素技術の樹形図	9
3.1	加速度センサを用いたパラメータ変動手法におけるシステムの全体像	12
3.2	加速度センサから得られる Android 端末上の座標系	13
3.3	X-Z 軸における加速度と鉛直方向の加速度	14
3.4	Y-Z 軸における加速度と鉛直方向の加速度	14
3.5	各コンテキストの認識手法と閾値について	16
3.6	各コンテキストの状態遷移図	16
3.7	加速度センサ + Wi-Fi を用いた手法におけるシステムの全体像	18
3.8	座標推定機構における主なプロセス	18
3.9	3次元空間における3つの球の様子	20
3.10	各アクセスポイントの設置例	21
3.11	様々なアクセスポイントの組み合わせから出力される推定座標	22
3.12	複数の推定座標における各2点間距離の計測	22
3.13	複数の推定座標における最小の2点間距離に選択された座標	22
3.14	コンテキスト推定機構における移動コンテキスト分類の流れ	24
4.1	加速度センサを用いたパラメータ変動手法における歩行ルート	27
4.2	accel.txt の例	28
4.3	timing.txt の例	28
4.4	Change_Context_log.txt の例	29
4.5	tougou.txt の例	29
4.6	WS=40,max-in=1.0,min-in=1.0 における「階段×昇り」の再現率	32
4.7	WS=40,max-in=1.0,min-in=1.0 における「階段×昇り」の適合率	32
4.8	WS=40,max-in=1.0,min-in=1.0 における「階段×降り」の再現率	32
4.9	WS=40,max-in=1.0,min-in=1.0 における「階段×降り」の適合率	32

4.10	WS=40,max-in=1.0,min-in=1.0 における「平坦×歩き」の再現率	32
4.11	WS=40,max-in=1.0,min-in=1.0 における「平坦×歩き」の適合率	32
4.12	WS のみ変動させたときの再現率	33
4.13	WS のみ変動させたときの適合率	33
4.14	WS のみ変動させたときの F 値	33
4.15	加速度センサ +Wi-Fi を用いた手法における歩行ルート	48
4.16	accel-wifi.txt の例	49
4.17	timing-wifi-CP.txt の例	50
4.18	昇りルート時のチェックポイント (CP) 位置	50
4.19	降りルート時のチェックポイント (CP) 位置	50
4.20	Change_Context_wifi_log.txt の例	52
4.21	timing-wifi.txt の例	52
4.22	tougou_wifi.txt の例	53
4.23	accelideal.txt の例	53

表目次

2.1	階段コンテキスト認識を行っている関連研究の精度	7
2.2	コンテキスト対象の比較	10
3.1	サポートしているセンサの一例	12
3.2	センサイベントの取得間隔に設定可能な定数	13
3.3	サポートしているセンサの一例	17
3.4	2点間の最小距離において選択された座標の回数	23
4.1	初期設定における再現率	30
4.2	初期設定における適合率	30
4.3	パラメータ最適化後の再現率	31
4.4	パラメータ最適化後の適合率	31
4.5	特定のコンテキストに特化したときの最大値 (F 値)	36
4.6	特定のコンテキストに特化したときの最大値 (再現率・適合率)	36
4.7	昇り + 降りの再現率	37
4.8	昇り + 降りの適合率	37
4.9	歩き + 昇り + 降りの再現率	38
4.10	歩き + 昇り + 降りの適合率	38
4.11	各被験者の最適化パラメータ	39
4.12	各被験者における特定のコンテキストに特化したときの最大値 (F 値)	41
4.13	各被験者における特定のコンテキストに特化したときの最大値 (再現率)	42
4.14	各被験者における特定のコンテキストに特化したときの最大値 (適合率)	42
4.15	昇り + 降りの再現率 (No.1)	43
4.16	昇り + 降りの適合率 (No.1)	43
4.17	歩き + 昇り + 降りの再現率 (No.1)	43
4.18	歩き + 昇り + 降りの適合率 (No.1)	43
4.19	昇り + 降りの再現率 (No.2)	43
4.20	昇り + 降りの適合率 (No.2)	43
4.21	歩き + 昇り + 降りの再現率 (No.2)	44
4.22	歩き + 昇り + 降りの適合率 (No.2)	44

4.23	昇り + 降りの再現率 (No.3)	44
4.24	昇り + 降りの適合率 (No.3)	44
4.25	歩き + 昇り + 降りの再現率 (No.3)	44
4.26	歩き + 昇り + 降りの適合率 (No.3)	44
4.27	昇り + 降りの再現率 (No.4)	44
4.28	昇り + 降りの適合率 (No.4)	44
4.29	歩き + 昇り + 降りの再現率 (No.4)	44
4.30	歩き + 昇り + 降りの適合率 (No.4)	44
4.31	昇り + 降りの再現率 (No.5)	45
4.32	昇り + 降りの適合率 (No.5)	45
4.33	歩き + 昇り + 降りの再現率 (No.5)	45
4.34	歩き + 昇り + 降りの適合率 (No.5)	45
4.35	昇り + 降りの再現率 (No.6)	45
4.36	昇り + 降りの適合率 (No.6)	45
4.37	歩き + 昇り + 降りの再現率 (No.6)	45
4.38	歩き + 昇り + 降りの適合率 (No.6)	45
4.39	昇り + 降りの再現率 (No.7)	45
4.40	昇り + 降りの適合率 (No.7)	45
4.41	歩き + 昇り + 降りの再現率 (No.7)	46
4.42	歩き + 昇り + 降りの適合率 (No.7)	46
4.43	昇り + 降りの再現率 (No.8)	46
4.44	昇り + 降りの適合率 (No.8)	46
4.45	歩き + 昇り + 降りの再現率 (No.8)	46
4.46	歩き + 昇り + 降りの適合率 (No.8)	46
4.47	昇り + 降りの再現率 (No.9)	46
4.48	昇り + 降りの適合率 (No.9)	46
4.49	歩き + 昇り + 降りの再現率 (No.9)	46
4.50	歩き + 昇り + 降りの適合率 (No.9)	46
4.51	アクセスポイント (AP) の設置座標	49
4.52	チェックポイント (CP) の設定座標	51
4.53	加速度 + Wi-Fi を用いた手法における No.1 の再現率	54
4.54	加速度 + Wi-Fi を用いた手法における No.1 の適合率	54
4.55	加速度 + Wi-Fi を用いた手法における No.2 の再現率	55
4.56	加速度 + Wi-Fi を用いた手法における No.2 の適合率	55
4.57	加速度 + Wi-Fi を用いた手法における No.3 の再現率	55
4.58	加速度 + Wi-Fi を用いた手法における No.3 の適合率	56
4.59	加速度 + Wi-Fi を用いた手法における No.1 の再現率 (理想)	56
4.60	加速度 + Wi-Fi を用いた手法における No.1 の適合率 (理想)	56

4.61	加速度 + Wi-Fi を用いた手法における No.2 の再現率 (理想)	57
4.62	加速度 + Wi-Fi を用いた手法における No.2 の適合率 (理想)	57
4.63	加速度 + Wi-Fi を用いた手法における No.3 の再現率 (理想)	57
4.64	加速度 + Wi-Fi を用いた手法における No.3 の適合率 (理想)	58
4.65	アクセスポイント (AP) とチェックポイント (CP) 間における推定距離の誤差 (CP1~20)	60
4.66	アクセスポイント (AP) とチェックポイント (CP) 間における推定距離の誤差 (CP21~33)	61

第 1 章

序論

昨今のインターネットの発展により、様々な Web コンテンツが誕生した。例えば、YouTube に代表される動画共有サービスや Amazon に代表される EC（電子商取引）、Facebook に代表される SNS などが挙げられる。これら娯楽性の強い Web コンテンツの利用割合は増加傾向にあり、今後益々増えていくと考えられる。しかし、これらの Web コンテンツに夢中になるあまり、勉強が疎かになるなど、実生活への影響を懸念する声もある。そこで、これまでの研究では、Web コンテンツの誘惑に負けてしまう人々の悩みを解決するために、既存のフィルタリング技術を応用して、自身のアクセス行為を制御（自己抑制）できる固定端末向けのアプリケーションを作成した [1, 2].

しかし、近年はモバイル環境における利用が広がることで、周囲に人間がいる状況での利用など、人間が暮らす生活空間との関係が密になってきている。そのため、従来のような人間と端末だけの関係性では見られなかった、新たな問題が発生している。例として、近年事故が増加傾向にあり、社会問題となりつつある「歩きスマホ問題」が挙げられるだろう。歩きスマホとは、ユーザが歩きながらスマートフォン（広義的には、携帯電話やタブレット端末等のモバイル端末も含む）を操作する行動であるため、周囲に対する注意が散漫となることから危険な行動とされている。特に、公共機関など周囲に多くの人々が行き交う場所では、通行人にぶつかったり、接触事故を起こしたりトラブルが発生し易い。図 1.1 は、東京消防庁管内で平成 22 年から平成 25 年までの 4 年間で歩きながら、自転車に乗りながら等の携帯電話、スマートフォン等に係る事故により 122 人が救急搬送された年別の人員を表している [3]。図 1.1 から、平成 25 年が 36 人と最も多く、歩きスマホが原因による事故が年々増加傾向にあり、今後も同様の傾向が続くと予想される。

これら増加する歩きスマホ事故に対する国内の動向として、NTT docomo, au, SoftBank の大手携帯キャリア各社が歩きスマホを防止するアプリケーションの提供を開始したことや、公益社団法人の AC ジャパンが歩きスマホを啓発するコマーシャルを制作し、テレビ等で放映を行っている等の動きがある。一方国外では、アメリカのアイダホ州とニュージャージー州で歩きスマホを行った人物に対する罰金を設けた法律が制定されており、歩きスマホを強制的に止めさせる動きとなっている。現在の日本国内における動向は、歩きスマホの危険性に対する判断をユーザに委ねている傾向にあるが、アメリカに倣い、法的措置による強制的な縛りが必

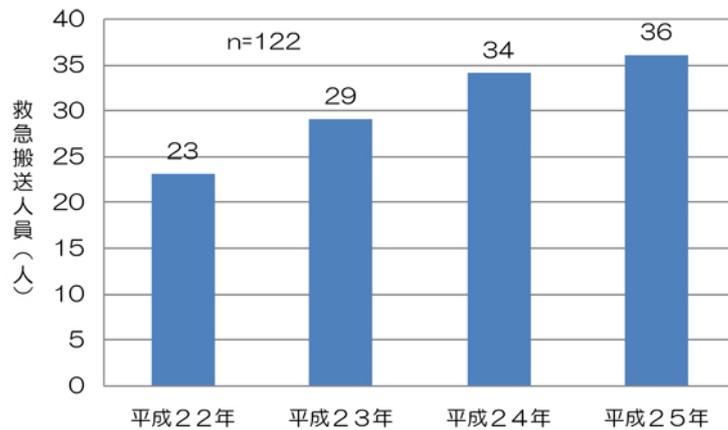


図 1.1 東京消防庁管内における年別の救急搬送人員

要だという議論も多い。しかし法律による禁止は、空間を問わず様々な情報にアクセスできるモバイル端末のメリットを損なってしまう可能性がある。また、ユーザにより危険を感じる状況は異なるため、歩きスマホを一律に禁止すると不満を持つユーザは多く残るだろう。

現状、歩きスマホ問題における事故は、駅のホームから転落する事故が多いが、柵を設置することで物理的な対策を行うことができる。しかし、階段ではユーザ自身で歩きスマホを自己抑制しないと転倒事故を起こす危険性が高く、巻き込み事故など周囲に影響を与える大きな事故も起こし易い。図 1.2 は歩きスマホに関する統計ではないが、国土技術政策総合研究所がマンションの共用部分、駅前ビル、商業ビル、オフィスビル、病院、役所、ホール等の公共性が高い建物内での事故について、事故が起きた建物の場所と事故の種類の関係性を調査したアンケート結果である [4]。

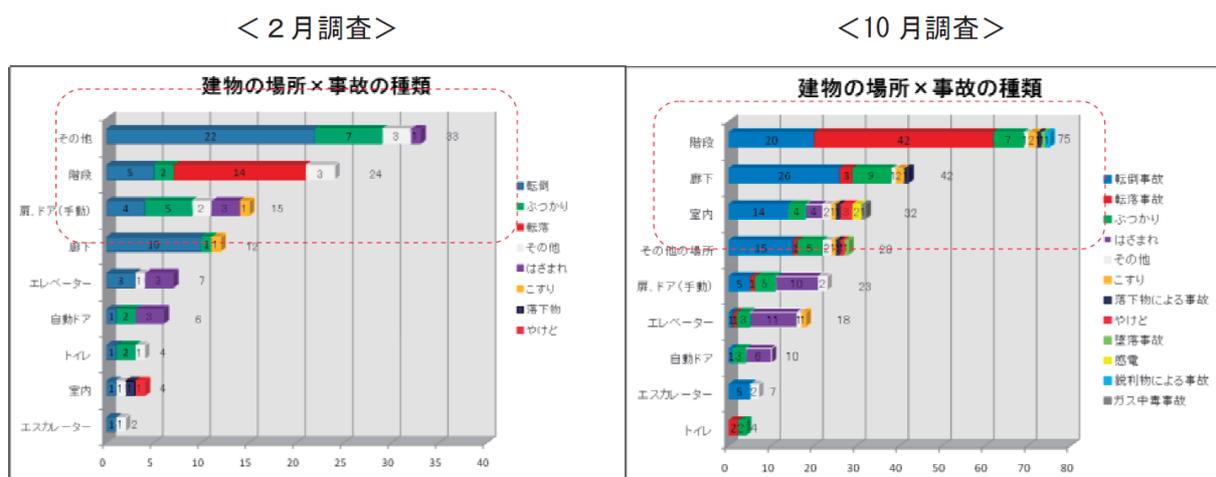


図 1.2 事故が起きた建物の場所と事故の種類の関係

図 1.2 から、歩きスマホを行っていない状態であっても階段における事故が多いことがわかり、歩きスマホを行った場合はより危険性が増すだろう。実際、階段から転落して重傷した事例も確認されている [3]。

以上のことから、歩きスマホ問題のような、従来の固定端末では見られない新しい問題に対して、モバイル環境向けの新しい自己抑制が必要であると考えられる。そこで、階段における歩きスマホ事故を未然に防ぐため、端末が階段におけるユーザの状況（階段コンテキスト）を認識するコンテキスト・アウェアネスの技術を用い、ユーザが歩きスマホをしながら階段を移動したことを検知し、画面にアラートを発することで自粛を促すモバイルアプリケーションを提案する。図 1.3 に提案するモバイルアプリケーションのイメージを示す。

また、提案するアプリケーションの実現にはいくつかの要素技術が考えられ、その 1 つに「階段におけるユーザの状況を認識できるコンテキスト・アウェアなシステム」がある。他にも、「ユーザにとってわかり易い操作性を追求するインタフェース」や「メールやゲーム等そもそも歩きスマホをしているのかどうかを判断するシステム」があるだろう。中でも本研究は、1 つ目の「階段におけるユーザの状況を認識できるコンテキスト・アウェアなシステム」の開発を目指して研究を行っていく。認識方法としては、端末にあらかじめ内蔵されている加速度センサから得られた加速度を利用する [5, 6]。また、その他のセンサとして屋内におけるユーザの位置推定技術に用いられている Wi-Fi を試験的に導入し、階段コンテキスト認識に利用できるかどうかの有用性を模索、検証する。



図 1.3 階段における歩きスマホの自粛を促す様子（イメージ）

第2章

関連研究

第2章では、関連研究との比較を行うことで、本研究の立ち位置を明確にしていく。まずはコンテキスト・アウェアネス（状況認識）の説明を行い、どのようなセンサや手法が用いられているのかを述べる。次に、階段におけるコンテキスト（階段コンテキスト）に関する研究の説明を行う。ここでは、主に加速度センサのみを用いた研究と、加速度センサと気圧センサを組み合わせた研究について述べる。その後、ネットワーク技術を用いてユーザの位置推定を行う、屋内測位システムの研究について説明する。最後に、本研究で目標とするコンテキスト対象を示していく。

2.1 コンテキスト・アウェアネスに関する研究

コンテキスト・アウェアネス（状況認識）とは、コンピュータがコンテキスト（ユーザの位置や行動状況）を認識することで、ユーザの利便性向上を目指す概念である。人間は目や耳といった感覚器が外界からの刺激を感受し、受け取った刺激を脳が処理することで、周囲の状況を認識できるが、コンピュータが周囲の状況を認識するには、感覚器となるセンサが必要となる。そこで、コンピュータにセンサを搭載し、ユーザの周囲における物理量の変化を解析することで状況認識を行う方法がコンテキスト・アウェアネスに関する研究では一般的となっている。

例えば、加速度センサを複数取り付けられたベルトを腰に巻き、得られた加速度をウェーブレット変換することで、ユーザの歩行状況を認識しようとする研究がある [7]。この他にも、人間にセンサを装着して、コンテキスト・アウェア・コンピューティングを実現しようという試みが行われている [8, 9]。ウェアラブルデバイスを用いた研究は、特殊な端末の利用をユーザに強いるため、導入コストが高くなり、普及には時間が掛かる。しかし、最近では、Apple Watch や Google Glass といったウェアラブル端末が普及しつつあるので、以前ほど普及に時間は掛からないと考えられる [10, 11]。それでも、携帯電話やスマートフォン、タブレット端末といったモバイル端末の普及に比べると大きな隔たりがあるため、多くのユーザを想定するならば、現状ではウェアラブル端末は現実的ではない。

モバイル端末の中でも、近年特に普及しているスマートフォンやタブレット端末には多様な

センサが内蔵されており、各ユーザが常に高機能なコンピュータを保持する状態となったため、コンテキスト・アウェアなシステムの導入コストが低くなった。そうした昨今の情勢を受けて、モバイル端末にあらかじめ内蔵されたセンサを用いた関連研究もいくつか行われるようになった。例えば、スマートフォンに内蔵されている加速度センサとマイクを用い、ユーザの飲酒状況を認識することで、酩酊状態または平常状態かを分類している研究が挙げられる [12]。同様に加速度センサとマイクを用い、宅内における皿洗いやアイロンがけ、掃除機がけ、歯磨きなどの生活行動の認識を実現している研究もある [13]。本研究においても、モバイル端末にあらかじめ内蔵されているセンサを利用したシステムを想定している。

その他にもコンテキスト・アウェアネスに関する研究は多数あるが、コンテキストがどういった物事を指し示すのかは、研究によって異なっている。多くの研究ではユーザ自身及び周囲に関する状況をコンテキストとして扱っているが、以降の説明を簡潔にするため、本研究ではコンテキストの対象をユーザのアクション、空間、周囲オブジェクトの3つに分類して話を進めていく。

2.2 階段コンテキストに関する研究

階段におけるコンテキスト（階段コンテキスト）を分類すると、アクションなら「昇り」、「降り」、「静止」などが挙げられる。より細かく分類するならば、「1段飛ばし」や「早昇り」などにも分類できるが、本研究では歩きスマホを行っているユーザを想定しているため、動きが遅いことはあってもせつちちなアクションを行うことは少ないと考えられる。そこで、本研究における階段コンテキストのアクションは、歩きスマホを行っているユーザが頻繁に行うと考えられる「昇り」、「降り」を扱う。

空間に関しては、「階段」、「階段前」を扱う。「階段前」とは、階段に入る前の空間であり、本研究においては廊下と階段の連結部分や、階段の踊り場を示す。例えば、従来研究の階段コンテキスト認識システムを用いて既に階段を歩いているユーザの状況を検知し、ユーザに対し警告を発したとしても、歩きスマホを行っているユーザにとっては、気付かないうちに階段に入っていて、階段でつまづいたり、階段から転げ落ちたりする危険性があり、手遅れになることが考えられる。そこで、階段前を認識して、階段に入る前にユーザに警告を発することができれば、安全性が高まると考えられるので、本研究では「階段前」も扱う。

階段コンテキストにおける周囲オブジェクトは「人」、「手すり」などが考えられる。歩きスマホは周囲の人々に対する危険性が高いことから、迷惑行為としても考えられているため、「人」が存在する場合に歩きスマホを自己抑制できることは、有用性が高いと考えられる。しかし、「人」や「人混み」などの周囲オブジェクトの認識は、先行研究の技術を応用することで実現できると考えられるので、本研究では扱わない。

2.2.1 加速度センサのみを利用した研究

階段コンテキストに関する研究では、加速度センサが多く利用されている。加速度センサは現在広く普及しているスマートフォンやタブレット端末にも搭載されているため、加速度センサで階段コンテキストを認識できれば、多くの人々に利用してもらえるだろう。そこで、本研究では普及率が高い加速度センサを用い、階段コンテキスト認識を行っていく。

まず、関連研究の1つに、加速度センサから得られた値を Naive Bayes や SVM といった 18 種類の機械学習によって状況認識を行っている研究がある [14]。各コンテキストにおける具体的な精度は掲載されていないものの、分類しているコンテキストは、「階段×昇り」や「階段×降り」、「平坦×歩き」、「平坦×走り」、「any（どこでも）×静止」などがある。しかし、「階段×昇り」を「階段×降り」と認識し、逆に「階段×降り」は「階段×昇り」と認識する間違いが多くなっている。2つ目は、携帯電話に加速度センサを取り付け、ユーザのアクションによって得られた値からウェブページ変換や自己組織化マップを行うことで状況認識を行っている研究である [15]。この研究では、「階段×昇り」や「階段×降り」の他にも「平坦×歩き」、「平坦×走り」、「平坦×早歩き」を扱っている。精度は、「階段×昇り」が約 74%、「階段×降り」が約 84%、「平坦×歩き」が約 84% となっており、比較的高い結果が得られている。

上述した関連研究では、機械学習を行うことで、誰が使用しても各コンテキストの認識精度を一定に保つことができる。また、これらの研究目的は、精度を高めることで何らかの支援に役立てるという漠然としたものである。しかし、本研究の目的である「階段における歩きスマホを自己抑制する」という観点から考えると、ユーザによって自己抑制したい状況は異なると考えられるため、認識精度を重視したいコンテキストも異なってくるだろう。全てのコンテキストにおいて 100% の精度を得られるのであれば問題はないが、加速度センサだけでの実現は難しい。そこで本研究では、ユーザが自己抑制したい状況に合わせ易くするため、特定のコンテキストに特化できるシステムを提案する。手法の詳細は後述する第 3 章にて述べる。

2.2.2 加速度センサ以外も利用した研究

複数のセンサを利用することで処理が必要となるデータ数は増えるが、様々な外界の変化を感受できるので、一般的には認識の精度向上が期待できる。階段コンテキスト認識において、加速度センサ以外に頻繁に利用されているセンサとしては、周囲の気圧の変化を観測できる気圧センサが挙げられる。気圧センサから得られる気圧値は、高い場所から低い場所への移動なら正の変化を、低い場所から高い場所への移動なら負の変化を示すため、階段昇降やエレベータなどの高低差が変化するコンテキストが観測し易くなる。実際に気圧センサを利用している関連研究もいくつかある。1つは、事前に作成したフローチャートと、リアルタイムで得られた加速度値、気圧値を比較して、各コンテキストに分類していく研究がある [16]。例えば「階段×昇り」を認識する場合なら、フローチャートの第 1 問「加速度値を変換した 0-7.5Hz 間の最大のパワースペクトル値が 0.005 以上か」を YES と判定し、さらに気圧値が負の変化をし

ているならば、「階段×昇り」と認識される．同様にフローチャートで設けられた質問に YES か NO で答えていくことで、「階段×昇り」や「階段×降り」、「平坦×歩き」、「平坦×走り」、「電車及び any ×静止」などのコンテキストを扱っている．システムの精度は、「階段×昇り」、「階段×降り」、「平坦×歩き」、「平坦×走り」が 100%、「電車及び any ×静止」が 60% となっていて、非常に高い結果が得られている．

また、その他に気圧センサを利用している関連研究に、加速度センサからは加速度の分散値を、気圧センサからは気圧の分散値と気圧差を特徴量として抽出し、構築した決定木による推定アルゴリズムを基に、段階的にコンテキストを推定していく研究がある [17]．この研究では、「階段×昇り」が 99.5%、「階段×降り」が 88.5%、「平坦×歩き」が 100%、「any ×静止」が 100% となっており、非常に高い精度となっている．

このように、加速度センサと気圧センサを組み合わせることで、精度が大きく改善されることがわかる．しかし、現在普及しているモバイル端末は、気圧センサの搭載率が低いため、大半のユーザは高い精度を誇る階段コンテキスト認識の恩恵を受けることができない．そうしたユーザは、加速度センサのみによる認識精度となってしまう、気圧センサと組み合わせた場合に比べ、大きく認識精度が下がってしまう．しかし、普及率が高い加速度センサのみで、高い精度を実現できれば、多くのユーザにとって有用となるだろう．そこで本研究では、加速度センサのみで、表 2.1 にまとめられた関連研究 [16, 17] の精度に近づけることを目標として研究を行っていく．但し、関連研究で述べられている精度は、いずれも本研究における適合率であると考えられるが、明確な評価方法が記載されていないため、詳細は不明である．

その他のセンサを利用して、階段コンテキスト認識を目指しているものに、生体センサを利用した研究がある [18]．生体センサは、ユーザの体温や心電図を計測できるセンサであり、得られた生体情報を利用することで、階段コンテキスト認識にも利用できる可能性がある．具体的な精度を求めるには至っていないが、「階段×昇り」や「階段×降り」では「平坦×歩き」に比べてユーザの心拍数が高くなることがわかっている．普及が進む腕時計型のウェアラブル端末である Apple Watch には、ユーザの心拍数を計測する機能が備わっているため、今後は心拍数を利用した階段コンテキスト認識に関する研究も増加する可能性があるだろう．

表 2.1 階段コンテキスト認識を行っている関連研究の精度

関連研究	「any ×静止」	「平坦×歩き」	「階段×昇り」	「階段×降り」
[15]	–	84%	74%	84%
[16]	60%	100%	100%	100%
[17]	100%	100%	99.5%	88.5%

2.3 屋内測位システムに関する研究

前節では、加速度センサと気圧センサを組み合わせることにより、高精度な階段コンテキスト認識が実現できると述べた．しかし、階段における歩きスマホは、階段を昇り降りしている

間だけが危険とは限らない。例えば、階段が近づいていることを知らずに歩きスマホを続けていて、気付いたときには階段から転げ落ちてしまうという危険性も考えられる。こうした危険性に対しては、ユーザが階段に近い（「階段前」）空間に足を踏み入れると、警告が発せられるアプリケーションを作成することで防げるだろう。だが、従来研究における実際の階段コンテキスト認識の手順は、まず「昇り」や「降り」といったアクションを認識し、それらのアクションが想定されるのは「階段」であるから現在は「階段」空間にいる、よって現在のコンテキストは「階段×昇り」である、という手順で認識されている。つまり、「階段」で一般的に想定されるアクションは「昇り」、「降り」であるという常識的側面に基づいているから、「階段」であると認識しているだけであって、厳密には空間とアクションを切り分けて認識できているのではない。一方「階段前」は、階段に近いという特徴はあるものの、通常の平坦な空間と変わらないため、様々なアクションが想定される。そのため、アクションを認識することで空間も認識するという方法では「階段前」空間を認識することはできない。そこで、アクションに依存せず、空間を認識できる技術である、屋内測位システムを利用する。

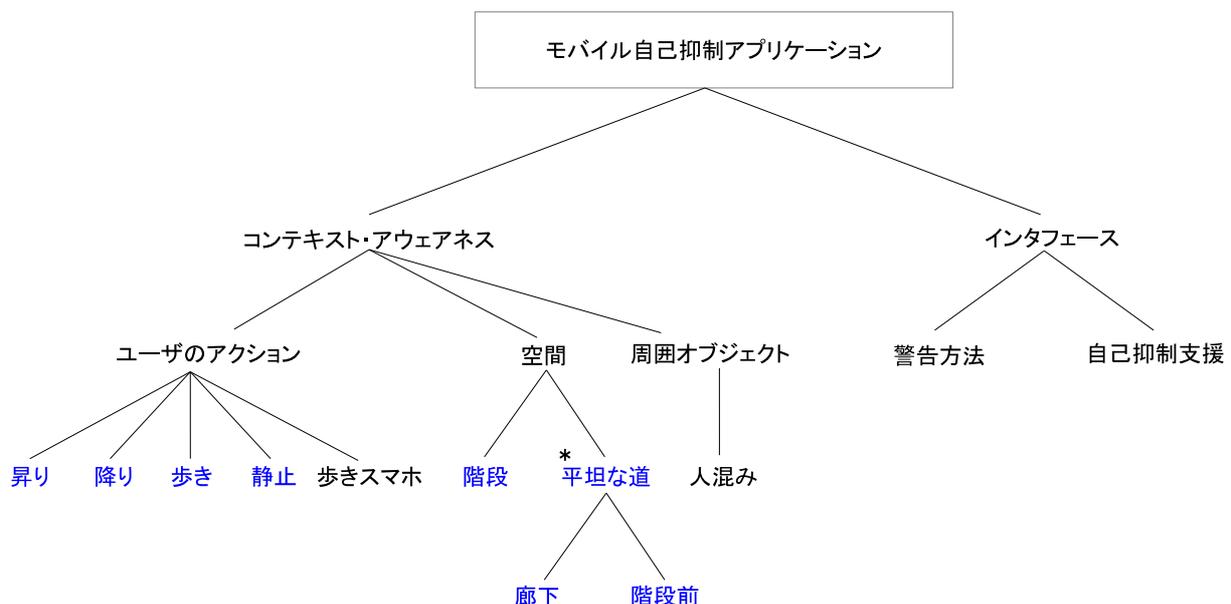
屋内測位システムとは、GPSの届かない屋内や地下でも、建物内に埋め込まれたセンサとユーザが保持するモバイル端末との相互通信を行うことで、ユーザの位置を推定できる技術である。センサには、無線LAN (Wi-Fi) や Bluetooth, RFIDなどが用いられており、空間内にセンサが埋め込まれてさえいれば、ユーザはただモバイル端末を持ち歩くだけで屋内測位システムを利用できる。そのため、ユーザにとっては導入コストが掛からないというメリットもある。

位置推定の主な手法には、センサから発した信号が端末に届くまでの時間を基にセンサと端末の距離を測る TOA, TOA の誤差を修正するために端末が発した信号がセンサに戻る往復時間も考慮した TDOA, RSSI (受信信号強度) によってセンサと端末間の距離を測る RSS-Based の3手法がある [19]。TOA や TDOA は GPS にも利用されている手法であり、高い精度で距離を測定できるが、センサと端末の正確な時刻同期が必要となる。しかし、時刻同期そのものが研究の目的になるほど、正確な時刻の同期は難しい [20]。また、屋内測位システムは GPS に比べて通信距離が短いため、より正確な時刻の同期をしない限り、センサと端末間でやりとりされている通信の時間差を観測することが難しい。そこで本研究では、導入が容易な RSS-Based を用いた手法を扱う。なお、詳細なアルゴリズムは後述の第3章にて述べる。

2.4 関連研究まとめ

第1章では、モバイル自己抑制アプリケーションには想定される要素技術がいくつか挙げられることや、本研究の目的は「階段におけるユーザの状況を認識できるコンテキスト・ウェアなシステム」の開発であると述べた。本研究の目的をより明確にするため、モバイル自己抑制アプリケーションの要素技術を「コンテキスト・ウェアなシステム」と「インタフェース」の2つに分類する。さらに、図2.1の樹形図を用いて細かく分類していくことで、本研究の最終目的を示していく。なお、青字の部分为本研究において扱うものとする。

第2.1節でも述べたが、コンテキストは「ユーザのアクション」、「空間」、「周囲オブジェクト」の3つに分けられる。この3つはさらに様々な要素に細分化できるが、図2.1には本研究に関連した要素のみを掲載している。「ユーザのアクション」に分類される「歩きスマホ」とは、ユーザがモバイル端末を使用しながら何らかのアクションを行っている状態である。「歩き」スマホという言葉を用いているが、本研究においては「歩き」に限定せずに、モバイル端末を使用しながらの全てのアクションとする。本研究では「歩きスマホ」認識を扱ってはいないが、最終的には「歩きスマホ」アクションと「降り」アクション、「階段」空間などを同時に満たす状態、つまり「歩きスマホ×降り×階段」をモバイル自己抑制できるようなアプリケーションの作成を目指し、今回はまず「降り×階段」の認識を目標としている。



* ... 但し、「加速度を用いた手法」では廊下と階段前を区別しないものとする

図 2.1 モバイル自己抑制アプリケーションに関する要素技術の樹形図

「周囲オブジェクト」に分類される「人混み」認識とは、ユーザの周囲に人混みが存在するかどうかを認識するものである。歩きスマホが危険視される原因の1つに、歩きスマホを行っている人が周囲の人々にぶつかることで怪我を負わせている事例があるため、歩きスマホに関連がある要素として図2.1に加えた。「人混み」認識を行っている関連研究に、Kinectを取り付けたスマートフォンで前方に人物やゴミ箱といったオブジェクトの有無を認識し、歩きスマホを行っているユーザのスマートフォン画面に警告を発するCrashAlertがある[21]。

「インタフェース」は「警告方法」と「自己抑制支援」に分けられる。「警告方法」は歩きスマホを行っているユーザに対し、どういった警告方法を提示するのが抑制に効果的であるかを検討するものである。関連研究では、歩きスマホをしながら記事やメールをブラウジングしているユーザに対し、画面上に音声での読み上げ等の代替操作を促す警告方法を提案している[22]。

一方、「自己抑制支援」とは、アプリケーションがユーザのモバイル自己抑制を支援する機

能である。自己抑制は自分で我慢をする行為であるため、我慢強い性格でなければ継続するのが難しく、歩きスマホの自己抑制も消極的になりがちである。そこで、ユーザが自己抑制をし易くなるような機能があれば有用であると考えられる。例えば、時間指定機能により、一度自分で指定した時間を過ぎるまでは設定を解除できないようにして、強制的に我慢を続けさせる方法が考えられる [1, 2]。または、最新の歩きスマホ事故に関する情報をユーザに知らせて、気を引き締めってもらうことで、モバイル自己抑制に対するモチベーションを高く保てるようにする方法も考えられる。

最後に、関連研究と本研究で扱うコンテキスト対象の違いについて、表 2.2 にまとめておく。なお、(階段) 降りという表記は、「階段×降り」と同義であり、「空間×アクション」のコンテキストを特に移動コンテキストとする。

表 2.2 コンテキスト対象の比較

	(any) 静止	(平坦) 歩き	(階段) 昇り	(階段) 降り	(階段前) 静止	(階段前) 歩き
関連研究 [15]	—	✓	✓	✓	—	—
関連研究 [16]	✓	✓	✓	✓	—	—
関連研究 [17]	✓	✓	✓	✓	—	—
本研究	✓	✓	✓	✓	✓	✓

第3章

提案手法

第3章では，端末にあらかじめ内蔵されている加速度センサのみを用いた「加速度センサを用いたパラメータ変動手法」と，屋内測位システムに利用されるネットワークセンサを組み合わせた「加速度センサ+Wi-Fiを用いた手法」の2つの手法について説明する．さらに，センサから得られたデータを利用して，具体的に各コンテキストへ分類する方法を各手法ごとに示していく．

3.1 加速度センサを用いたパラメータ変動手法

3.1.1 システム概要

本研究における階段コンテキスト認識システムは，図 3.1 に示す処理を繰り返すことで，ユーザの移動コンテキスト（空間×アクション）を認識する．なお，「加速度センサを用いたパラメータ変動手法」において認識を行う移動コンテキストは，「any（どこでも）×静止」，「平坦（廊下+階段前）×歩き」，「階段×昇り」，「階段×降り」の4種類である．また，階段での昇り降りを区別するのは，例えば降りは昇りと比べて高所からの転落による大怪我を起こし易いため，降りだけ危険に感じるなど，ユーザによって自己抑制したい状況が異なるからである．

まず処理1では，加速度センサが内蔵されているAndroid端末を持って静止，歩行，昇り，降りなどのアクションを行う．次に処理2で，各アクションによって端末上の座標系における加速度が取得され，これを世界座標系に変換して鉛直方向の加速度を取得する．処理3では，加速度センサから得られた鉛直方向の加速度をコンテキスト推定機構に送ると，コンテキスト推定が実施される．最後に処理4で推定結果を画面に出力し，処理1に戻る．

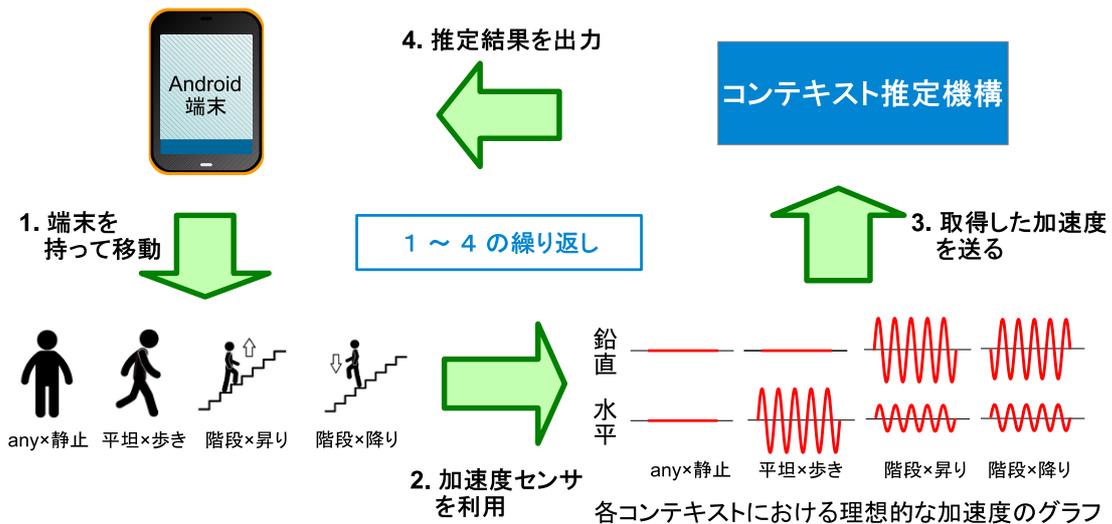


図 3.1 加速度センサを用いたパラメータ変動手法におけるシステムの全体像

3.1.2 加速度センサに関する処理

Android の開発環境には、端末に搭載されたセンサを利用するための `SensorManager` というクラスが用意されている。 `SensorManager` を取得した後、リスナーを登録することで、取得したセンサの値を処理できる。用意されているセンサには表 3.1 のようなものがある。

表 3.1 サポートしているセンサの一例

センサ名	定数名	概要
加速度センサ	<code>Sensor.TYPE_ACCELEROMETER</code>	加速度を検出するセンサ。
直線加速度センサ	<code>Sensor.TYPE_LINEAR_ACCELERATION</code>	重力成分を除いた加速度を検出するセンサ。
地磁気センサ	<code>Sensor.TYPE_MAGNETIC_FIELD</code>	方角を検知するセンサ。端末の向きを計測できる。
温度センサ	<code>Sensor.TYPE_AMBIENT_TEMPERATURE</code>	周囲の温度を検知するセンサ。
照度センサ	<code>Sensor.TYPE_LIGHT</code>	周囲の明るさを検知するセンサ。

例えば加速度センサを利用する場合、 `SensorManager` を用いて `Sensor.TYPE_ACCELEROMETER` をリスナーに登録することで加速度センサの変化を受け取れるようになる。本研究では、重力成分を除いた加速度が得られる直線加速度センサを利用し、以降の加速度センサという記載は直線加速度センサのこととする。

また、Android の開発環境は、表 3.2 のようにセンサのイベント取得間隔も設定することができる。但し、端末に搭載されている加速度センサによっては取得間隔を早い設定にしても想定より大きな遅延が発生する場合がある。本研究で使用した端末の場合は、 `SENSOR_DELAY_FASTEST` に設定しても、 `SENSOR_DELAY_NORMAL` より早い取得間隔にはならなかった。それ故、本研究では `SENSOR_DELAY_NORMAL` の取得間隔を採用した。なお、当然ではあるが端末にセンサが搭載されていない場合は、リスナーを登録しても変化を観測することはできない。

表 3.2 センサイベントの取得間隔に設定可能な定数

定数名	概要
SensorManager.SENSOR_DELAY_FASTEST	変化があり次第, 値を更新する.
SensorManager.SENSOR_DELAY_GAME	イベント変化後, 遅延時間 20ms 以下で値を更新する.
SensorManager.SENSOR_DELAY_NORMAL	イベント変化後, 遅延時間 60ms 以下で値を更新する.
SensorManager.SENSOR_DELAY_UI	イベント変化後, 遅延時間 200ms 以下で値を更新する.

Android 端末では, 加速度センサから得られる値は端末上の座標系であるため, 端末の向きによっては人間が生活する実空間の座標系 (世界座標系) とは異なってくる. そこで, 加速度センサから得られた値を端末上の座標系から世界座標系に変換することで, 端末の向きによらない加速度を取得できるようにする. また, 階段の昇り降りは, 平坦な道での歩行に比べ, 世界座標系における鉛直方向の変化を伴う運動であるため, 鉛直方向の加速度を階段コンテキスト認識に利用する.

図 3.2 に示すように, Android の加速度センサから得られる値の X 軸方向は端末右向きを正, Y 軸方向は端末上向きを正, Z 軸方向は端末の裏側からスクリーンへの向きを正としており, 世界座標系になっていない. そこで, 図 3.3 と図 3.4 に示すように, 端末の傾き θ と端末上の加速度からそれぞれ鉛直方向の加速度に分解し, 合力を計算することで最終的な鉛直方向の加速度を求めていく. なお, 端末の傾きは地磁気センサを利用することで取得できる.

まず, X-Z 平面の場合, 図 3.3 の通りに X 軸と Z 軸に分けて考えていく. X 軸なら加速度センサから得られる X 軸上の加速度 x と, θ を用いて, 鉛直方向の加速度を $x \cos(90^\circ - \theta)$ と表すことができる. 同様に Z 軸上の加速度を z_1 とすると, 鉛直方向の加速度は $z_1 \cos \theta$ と表せる. それぞれ求めた鉛直加速度の合力が, X-Z 軸平面における鉛直方向の加速度となる.

また, Y-Z 軸平面の場合も, 図 3.4 の通りに Y 軸と Z 軸に分解する. Y 軸上の加速度 y と θ' を用いると $y \cos(90^\circ - \theta')$ と表すことができ, Z 軸上の加速度を用いると $z_2 \cos \theta'$ と表せる. これらの合力を求めると Y-Z 軸平面における鉛直方向の加速度となる.

最後に, X-Z 軸平面における鉛直方向の加速度と, Y-Z 軸平面における鉛直方向の加速度を合わせた合力が, 端末の傾きによらない世界座標系の鉛直加速度となる.



図 3.2 加速度センサから得られる Android 端末上の座標系

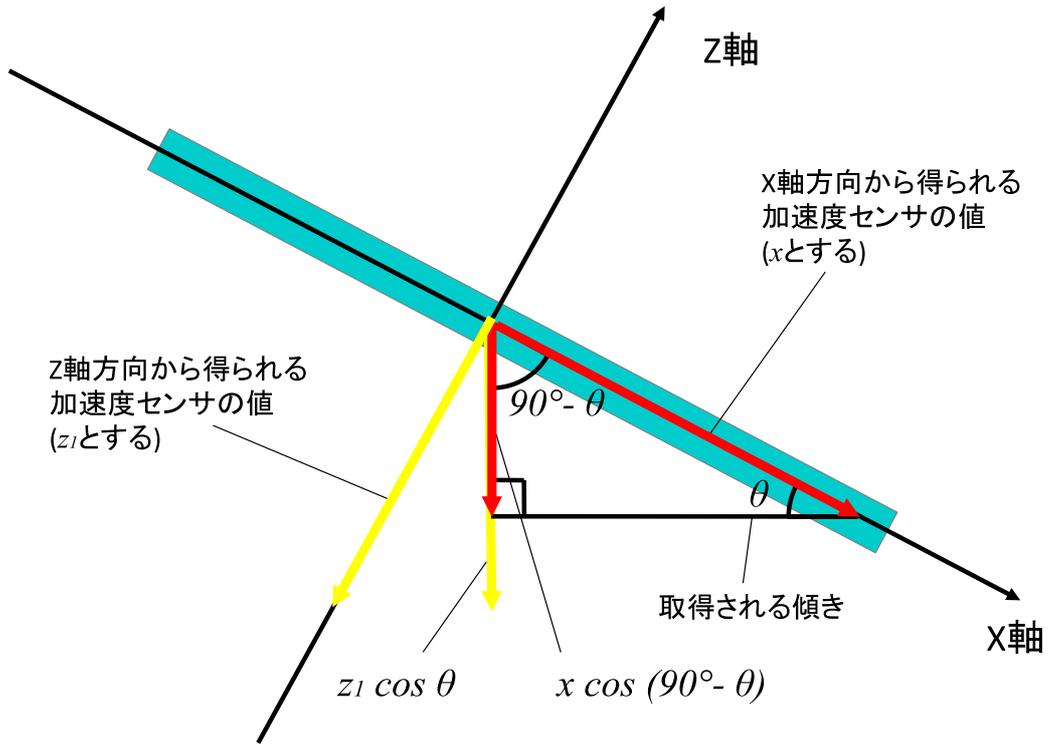


図 3.3 X-Z 軸における加速度と鉛直方向の加速度

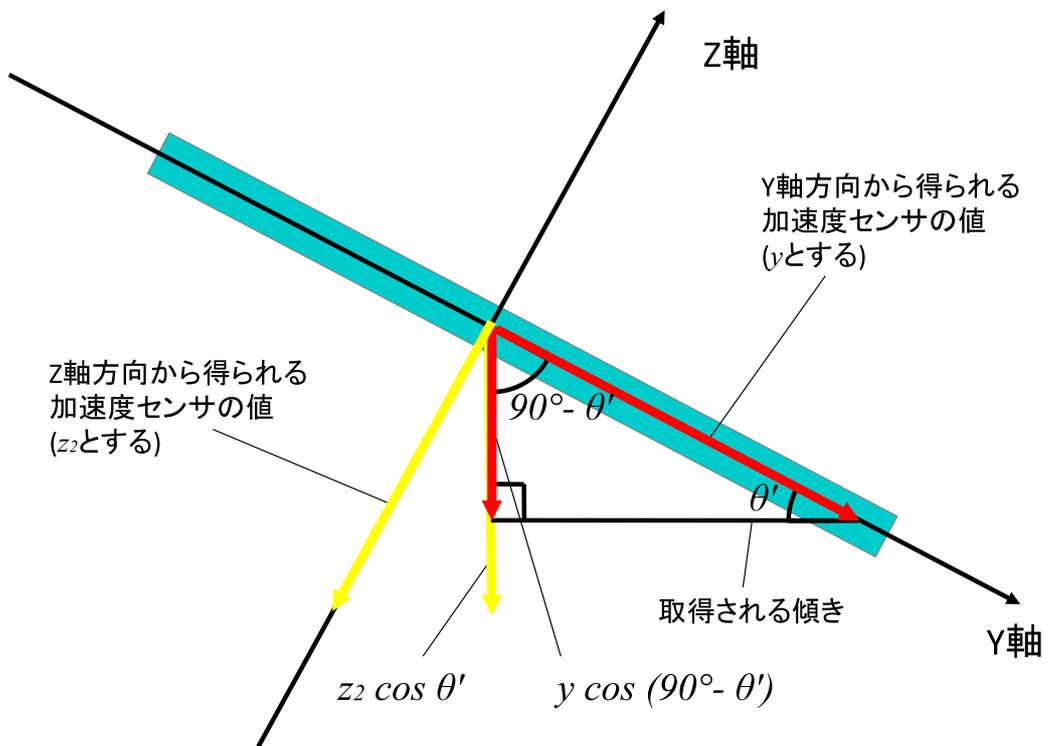


図 3.4 Y-Z 軸における加速度と鉛直方向の加速度

3.1.3 コンテキスト推定機構

本項では、図 3.1 に示したコンテキスト推定機構の詳細について説明する。コンテキスト推定機構では、鉛直方向の加速度とウィンドウを利用して、各コンテキストの認識を行っている。ウィンドウは加速度を格納する配列であり、ウィンドウサイズ（以下、WS）は 30 点である。また、サンプリング間隔はいずれも 60ms 間隔以内ではあるが不定である。図 3.5 に、加速度とウィンドウを利用した各コンテキストの認識手法を示す。

各コンテキストには閾値が設定されており、これをウィンドウに含まれる加速度の最大値・最小値と比較し、コンテキスト認識を行う。まず「any × 静止」の場合、閾値は最大値が 0.5 以下かつ最小値が -0.5 以上のときである。ウィンドウ内の最大値と最小値はそれぞれ 0.4 と 0.0 であり、閾値を超えないため、「any × 静止」と認識される。「平坦 × 歩き」は、最大値の閾値が 0.5 以上かつ 2.0 以下を満たし、さらに最小値の閾値が -2.0 以上かつ -0.5 以下を満たすときである。ウィンドウ内の最大値は 0.6 で、最小値は -0.8 であるので、「平坦 × 歩き」と認識される。「階段 × 昇り」の閾値は、最大値が 2.0 以上を満たすときである。ウィンドウ内の最大値は 2.2 で、最小値は 0.0 であるが、「階段 × 昇り」の認識には最小値は無関係であるため、最大値が 2.0 を超えてさえいれば「階段 × 昇り」と認識される。一方、「階段 × 降り」の閾値は、最小値が -2.0 以下を満たすときである。ウィンドウ内の最大値は 1.6 で、最小値は -3.1 であるが、「階段 × 降り」の認識には最大値は無関係であるため、最小値が -2.0 を下回っていれば階段降りと認識される。

しかし、実際の階段は複数段あるため、図 3.5 のような 1 段分の波形にはならず、振幅の大きな波が連続して観測される。昇りと降りは共に加速度の振幅が大きいため、例えば実際のコンテキストが昇りであったとしても、ウィンドウ内の最大値が 2.0 を超えた直後に最小値が -2.0 を下回り、再び最大値が 2.0 を超える状況や、最大値・最小値が同時に昇り降りの閾値を満たす状況が発生する。そこで状態遷移を導入し、昇りと降りは互いに行き来しないようにすることで、昇りと降りの認識を交互に繰り返すのを防いでいる。つまり、始めに昇りと認識した場合、その後も昇りと認識され続けるようになっている。そのため、始めの認識を間違えた場合は、その後の認識も間違い続けてしまう。また、階段で急に昇りから降りに方向転換した場合なども、その変化を精確に認識することはできない。だが、方向転換の問題に関しては、水平方向の加速度を利用することで解決できると考えられる。図 3.6 に状態遷移図を示す。

なお、本システムでは各コンテキストの閾値を 4 つに分類している。「any × 静止」と「平坦 × 歩き」を区別する最大値側の閾値を max-in、同じく最小値側の閾値を min-in とし、「平坦 × 歩き」と「階段 × 昇り」の閾値を max-out、「平坦 × 歩き」と「階段 × 降り」の閾値を min-out とする。max-in と min-in は「any × 静止」と「平坦 × 歩き」の認識に影響を与えるが、「階段 × 昇り」と「階段 × 降り」の認識には無関係である。一方、max-out は「階段 × 昇り」と「平坦 × 歩き」に影響を与えるが、「any × 静止」と「階段 × 降り」には影響を与えず、min-out は「階段 × 降り」と「平坦 × 歩き」に影響を与えるが、「any × 静止」と「階段 × 昇り」には影響を与えない。

本研究では、4つの閾値とWSをパラメータとし、変動させることで、各ユーザが自己抑制したいコンテキストに調整できるようにしている。そこで、 $WS=30$, $max-out=2.0$, $max-in=0.5$, $min-out=-2.0$, $min-in=-0.5$ を初期設定のパラメータとし、第4章にて各被験者に対するパラメータ最適化を行っていく。

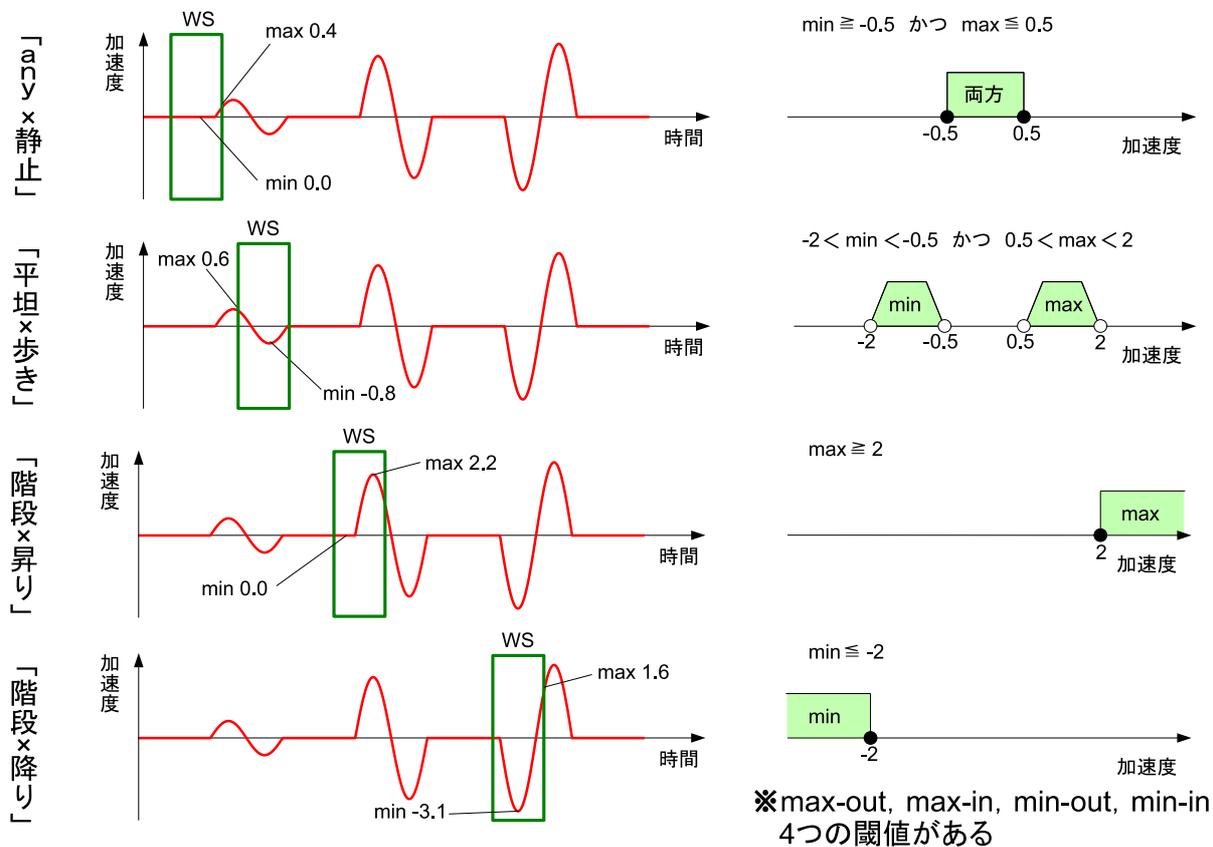


図 3.5 各コンテキストの認識手法と閾値について

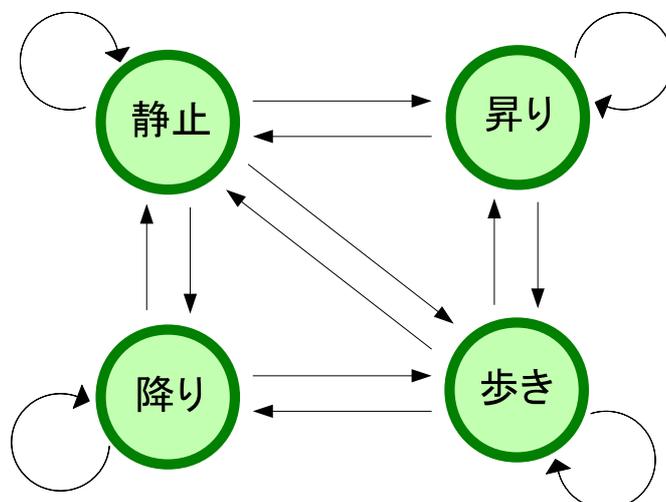


図 3.6 各コンテキストの状態遷移図

3.2 加速度センサ + Wi-Fi を用いた手法

3.2.1 システム概要

加速度センサと Wi-Fi を組み合わせた階段コンテキスト認識システムでは、図 3.7 に示す処理を繰り返すことで認識を行う。なお、「加速度センサ + Wi-Fi を用いた手法」において認識を行う移動コンテキストは、「any (階段前以外) × 静止」、「平坦 (廊下) × 歩き」、「階段 × 昇り」、「階段 × 降り」、「階段前 × 静止」、「階段前 × 歩き」の 6 種類である。

まず処理 1 では、加速度センサ及び Wi-Fi に対応している Android 端末を携帯して静止、歩き、昇り、降りなどのアクションを行う。処理 2 では、各アクションによって端末上の座標系における加速度を取得、これを世界座標系に変換をして、鉛直方向の加速度を取得する。続いて処理 3 で、事前に特定の x , y , z 座標に設置してある周囲の Wi-Fi アクセスポイントから、受信信号強度 (RSSI) を取得する。処理 4 の座標推定機構では、取得した RSSI (負値) が小さければアクセスポイントとの距離が遠く、逆に大きければ距離が近いという性質を利用して、Android 端末とアクセスポイント間の距離を推定する。その後、推定距離とアクセスポイントの座標から、端末の x , y , z 座標を推定する。次に処理 5 で、鉛直方向の加速度と、端末の推定座標をコンテキスト推定機構に送ると、コンテキスト推定が実施される。最後に処理 6 で、推定結果を画面に出力し、処理 1 に戻る。

3.2.2 座標推定機構

座標推定機構では、図 3.8 のような 3 つの処理が行われている。まずは第 1 プロセス「推定距離の算出」について説明していく。Android の開発環境には、加速度センサや地磁気センサと同様に、Wi-Fi の各種情報を取得できる仕組みが用意されている。WifiManager クラスを利用すると、周囲の Wi-Fi アクセスポイントをスキャンして、表 3.3 にある情報を取得できる。

表 3.3 サポートしているセンサの一例

BSSID	アクセスポイントの MAC アドレス
SSID	ネットワーク名
capabilities	暗号化情報
frequency	周波数 (MHz)
level	RSSI (dBm)

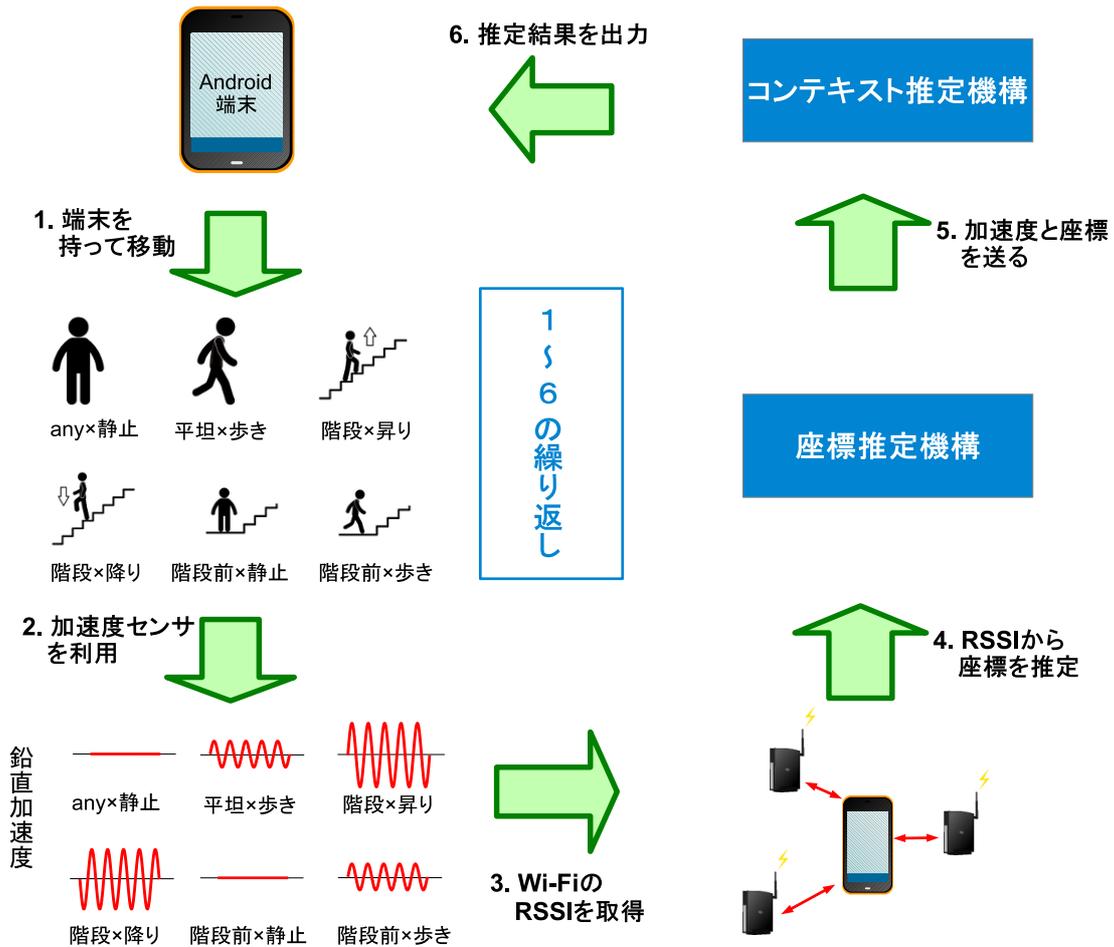


図 3.7 加速度センサ +Wi-Fi を用いた手法におけるシステムの全体像

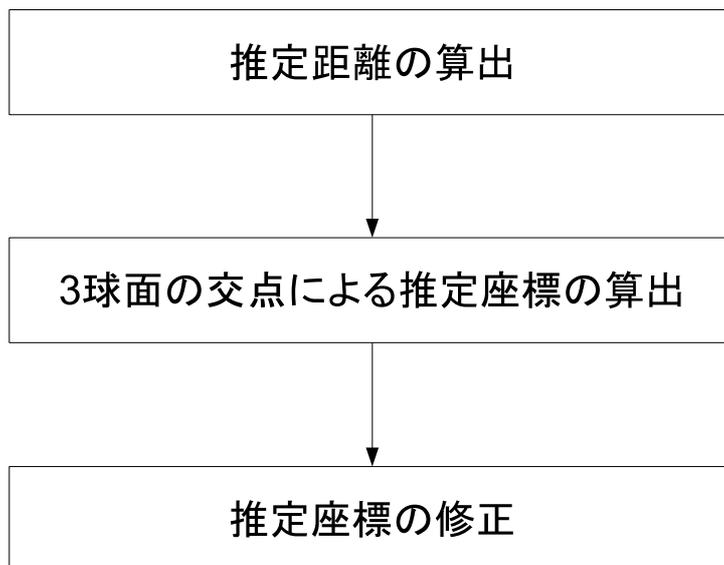


図 3.8 座標推定機構における主なプロセス

アクセスポイントとモバイル端末間の距離を推定するには、電波伝搬の関係式を利用する方法がある。実際の建物は複雑な構造であるため、電波の伝わり方は反射や回折を繰り返し減衰していく。しかし、複雑な電波伝搬の減衰を表した式を立てることは難しいため、本研究では単純な自由空間伝搬損失の関係式を用いる。その中でも特に、送受信アンテナにアイソトロピックアンテナを使用した場合の自由空間基本伝搬損失を利用する。

テレビや衛星に利用されるアンテナはダイポールアンテナと呼ばれ、電波に指向性を保持させるため、入り組んだ骨組みをしている。一方、アイソトロピックアンテナは、指向性を持たず、電波を全ての方向に発射あるいは受信するアンテナである。本研究に使用したアクセスポイントとモバイル端末は、複雑な骨組みのアンテナを使用していないことから、アイソトロピックアンテナであると仮定して、自由空間基本伝搬損失を利用する。自由空間基本伝搬損失の関係式を式 3.1 に示す。

$$P_r = P_t - L_B \quad (3.1)$$

式 3.1 の P_r は受信電力、 P_t は送信電力、 L_B は自由空間基本伝搬損失を表している。なお、受信電力は表 3.3 の level で得られる RSSI、送信電力はアクセスポイントとモバイル端末の距離が 0 であるときの RSSI と言い換えることができる。そのため、アクセスポイントとモバイル端末を密着させて RSSI を計測した値が、 P_t の値となる。事前実験の結果、 $P_t = -25[\text{dBm}]$ という結果が得られた。なお、数値は使用する機器によって異なるので、実験環境に合わせて逐一計測する必要がある。さらに、 L_B は式 3.2 と表すことができる。

$$L_B = 20 \log_{10} \left(\frac{4\pi d}{\lambda} \right) \quad (3.2)$$

式 3.2 の d がアクセスポイントからの距離 [cm] であり、現在求めたい推定距離にあたる。また、 λ は電波の波長を表している。距離 0 における RSSI であることから P_t を P_0 とすると、式 3.1 と式 3.2 から、最終的な自由空間基本伝搬損失の関係式を式 3.3 と表すことができる [23]。

$$P_r = P_0 - 20 \log_{10} \left(\frac{4\pi d}{\lambda} \right) [\text{dBm}] \quad (3.3)$$

また、 λ は、電波の周波数 f と電波の速度を c とすると式 3.4 と表せる。なお、電波は光と同じ速度であるため、 $c = 3 \times 10^8 [\text{km/s}]$ である。

$$\lambda = \frac{c}{f} \quad (3.4)$$

f の周波数は、表 3.3 の frequency から得られるので、 λ を求めることができる。求めた λ を式 3.3 に代入し、さらに式 3.3 の P_r は表 3.3 の level から、 P_0 は事前実験の結果からそれぞれ既知であるので、残りの変数は推定距離の d のみとなり、計算ができる。以上で、図 3.8 における第 1 プロセス「推定距離の算出」は終わる。

続いて、第 2 プロセス「3 球面の交点による推定座標の算出」について説明していく。3 次元空間におけるユーザの位置を推定する方法の 1 つに、GNSS 測量がある。古くは GPS 測量

と呼ばれ、GPS (Global Positioning System) に利用されている方法である。GNSS 測量では、地球を周回する GPS 衛星が自身の位置 (x_i, y_i, z_i) と時刻 t_i を送信し、地上にある GPS 受信機 (x, y, z) で受信した時刻 t と発信時刻 t_i との差から受信機と衛星間の距離 r_i がわかるため、3次元空間における2点間の距離を求める式 3.5 を立てることができる。但し、 (x, y, z) は未知数、 c は光速とする。

$$(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2 = \{c(t_i - t)\}^2 = r_i^2 \quad (3.5)$$

3つの衛星から電波を受信することで式 3.5 を3本立てることができるので、3元連立方程式を解くことにより未知数 (x, y, z) を求めることができる。しかし、屋内測位システムで用いる機器は、GPS 衛星に利用されているような高性能な原子時計が内蔵されていないため、時刻の同期が難しい。そこで、本研究では RSSI を用いた手法 (RSS-Based) に基づいて、式 3.3 から直接 r_i^2 を求める。なお、実際の GNSS 測量では、GPS 受信機に搭載されている時計の誤差 Δt (未知数) を考慮し、4個の衛星から電波を取得して、さらにもう1本多く式を立てて計算を行っている。

ここで、式 3.5 は球の方程式となっているため、図 3.9 に示すようなそれぞれ半径 r_1, r_2, r_3 の3つの球を考える。但し、図中の AP とはアクセスポイントのことである。

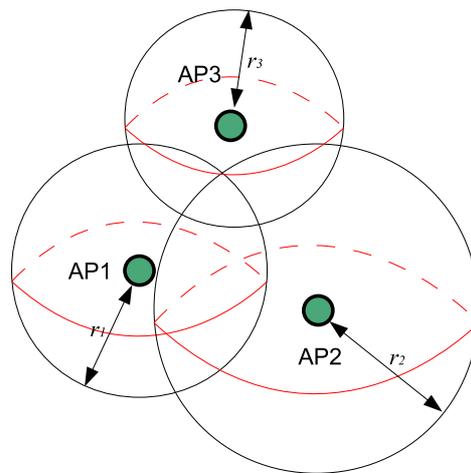


図 3.9 3次元空間における3つの球の様子

各アクセスポイントが形成する球の半径 r_1, r_2, r_3 は、第1プロセス「推定距離の算出」によって求められ、例えば r_1 は AP1 とユーザ間の推定距離となる。各々、 $P_i(x_i, y_i, z_i)[i=1, 2, 3]$ を中心とする半径 $r_i[i=1, 2, 3]$ の球とし、また、ユーザの座標を3球面の交点 $P(x, y, z)$ とすると、式 3.6~式 3.8 が成り立つ。但し、 $P_i(x_i, y_i, z_i)[i=1, 2, 3]$ は既知であり、例えば図 3.10 のように、事前にアクセスポイントを特定の座標に設置しておくこととする。

$$(x_1 - x)^2 + (y_1 - y)^2 + (z_1 - z)^2 = r_1^2 \quad (3.6)$$

$$(x_2 - x)^2 + (y_2 - y)^2 + (z_2 - z)^2 = r_2^2 \quad (3.7)$$

$$(x_3 - x)^2 + (y_3 - y)^2 + (z_3 - z)^2 = r_3^2 \quad (3.8)$$

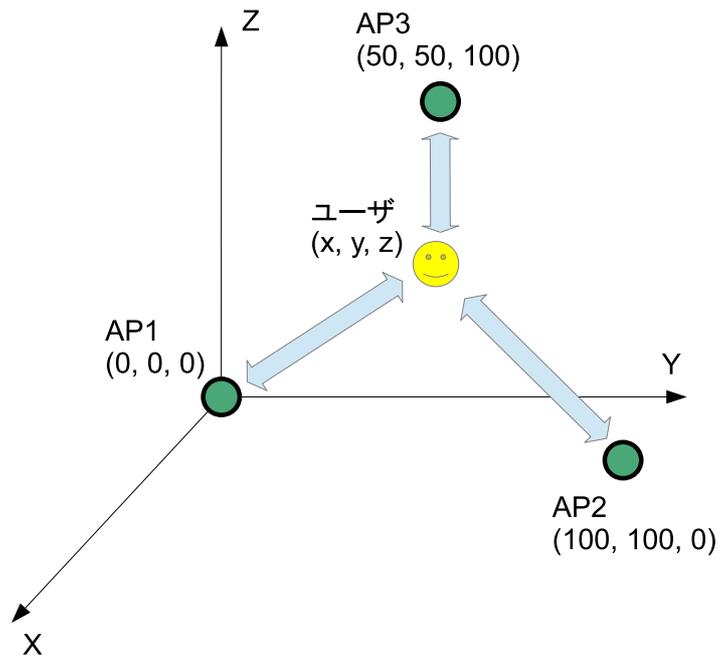


図 3.10 各アクセスポイントの設置例

最終的に式 3.6～式 3.8 は、未知数が x , y , z の 3 つの 3 元連立方程式となるので、これを解くことでユーザの座標を求められる。なお、3 球面の交点がない場合は、ユーザの座標を求めることはできない。

最後に、第 3 プロセス「推定座標の修正」について説明していく。第 2 プロセス「3 球面の交点による推定座標の算出」では、ユーザの推定座標を求められることを説明した。しかし、方程式に交点がある場合は解（推定座標）が 2 つ（重解の場合もあり）得られる。そのため、最終的なユーザの座標を 1 点に定めるには、2 点の内どちらがユーザの座標であるかを推定しなければならない。また、アクセスポイントの数が、図 3.10 のように 3 つだけの場合とは異なり、複数のアクセスポイントが存在する場合もある。この場合、3 つのアクセスポイントを選ぶ組み合わせも複数存在するため、より多くの推定座標が得られる。そこで、第 3 プロセスでは、複数出力された座標から、最終的な 1 点を定めるための処理を説明する。

例えば周囲にアクセスポイントが 7 つあった場合、3 つのアクセスポイントを選ぶ組み合わせは ${}^7C_3 = 35$ 通りとなる。それらの内、実際に交点が存在する組み合わせが 3 つであったとすると、図 3.11 のように A, B, C, D, E, F の全部で 6 つの推定座標が出力される。この時、第 1 プロセスで求めた推定距離が精確な場合は、どのアクセスポイントの組み合わせでも重なる点、例えば A, D, E の 3 つが重なるので、重なった点がユーザの座標と推定できる。しかし、実際は推定距離に誤差が生じるので、図 3.11 のように推定座標も分散してしまう。そこで、互いに座標が近い点の周辺にユーザの座標が存在するものと仮定して、各 2 点間の距離を計測し、距離が最小の 2 点を見つける。

組み合わせ1: AP1, AP2, AP3による座標 → (A) (B)
 組み合わせ2: AP1, AP2, AP4による座標 → (C) (D)
 組み合わせ3: AP3, AP6, AP7による座標 → (E) (F)

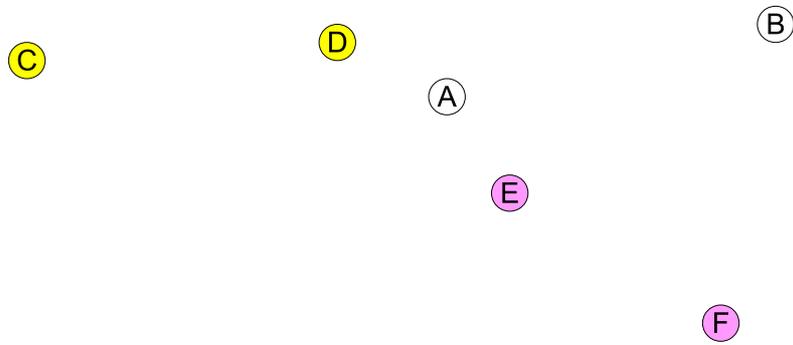


図 3.11 様々なアクセスポイントの組み合わせから出力される推定座標

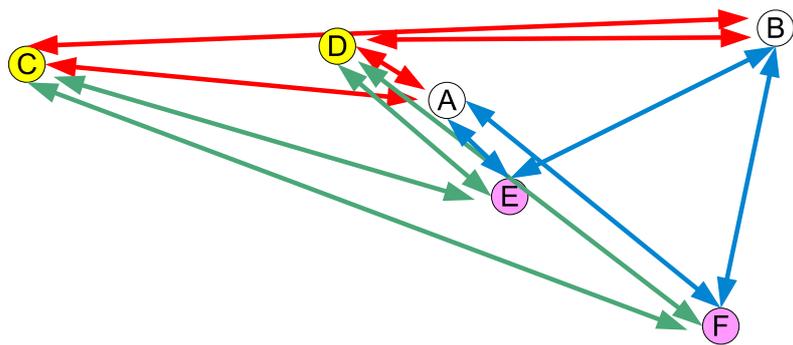


図 3.12 複数の推定座標における各2点間距離の計測

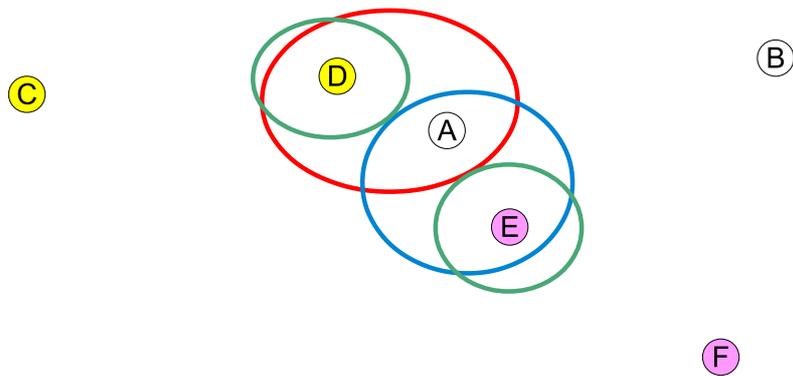


図 3.13 複数の推定座標における最小の2点間距離に選択された座標

例えば図 3.12 の場合では、赤の矢印は組み合わせ 1 の座標と組み合わせ 2 の座標で、青の矢印は組み合わせ 1 の座標と組み合わせ 3 の座標で、緑の矢印は組み合わせ 2 の座標と組み合わせ 3 の座標で、それぞれ 2 点間距離を計測することを表している。その結果、図 3.13 に示すように、赤の矢印では A と D が、青の矢印では A と E が、緑の矢印では D と E がそれぞれ最小の 2 点間距離となる。最小の 2 点間距離に選択された座標の回数をまとめると、表 3.4 のようになり、この結果から、選択された回数が多い方、つまり、組み合わせ 1 では A の方が B に比べてユーザに近い座標（基準座標）とする。同様に、組み合わせ 2 なら基準座標は D、組み合わせ 3 なら F となる。基準座標を決めることで、3 球面の交点で得られた 2 点の推定座標を 1 点に絞ることができる。なお、例えば選択された回数が A は 1 回、B も 1 回の同数の場合は、どちらの点も基準座標として扱わない。

表 3.4 2 点間の最小距離において選択された座標の回数

組み合わせ	座標	選択された回数
組み合わせ 1	A	2
	B	0
組み合わせ 2	C	0
	D	2
組み合わせ 3	E	2
	F	0

続いて、基準座標を最終的な 1 点に絞る方法として、軽重率と最確値という概念を導入する。まずは軽重率から説明していく。軽重率は実際の測量に用いられている測定値の信用度を示す重みであり、測定回数に比例し、測定距離に反比例する。地図の作成などに利用されていた三角測量は、ある基線の両端にある既知の点から測定したい点への角度をそれぞれ測定することによって、未知点の位置を決定する測量方法であり、新たに既知となった点を利用して三角測量を繰り返すことでより広い範囲の測量ができるようになっていく。この時、三角測量に用いられた回数が多い点は信用度が高くなる。また、三角測量に利用した点と未知点との距離が長い場合は、測量に使用した既知点を信用度が低い点として扱う。

電波に関しても同様に、多くの関連論文でアクセスポイントからの距離が遠いほど信用度が低い点として扱っている。これは、アクセスポイントとユーザ間の距離が長いほど、RSSI を減衰させる要因が混入し易いからである。しかし、最終的な 1 点の座標を決める方法は各研究でそれぞれ異なっている。そこで本研究では、軽重率を導入して最終的な 1 点を求める方法を用い、どの点がより多く選択されて基準座標となったかを測定回数として信用度に比例させ、アクセスポイントの合計距離を測定距離として信用度に反比例させる。 i を組み合わせ番号とすると、軽重率 $p_i \{p_i > 0\}$ は、式 3.9 のようになる。但し、 n は測定回数とし、 $d_{i,a}$, $d_{i,b}$, $d_{i,c}$ はそれぞれ組み合わせ i における AP 番号 a , b , c とユーザ間の推定距離とする。

$$p_i = \frac{n}{d_{i,a} + d_{i,b} + d_{i,c}} \quad (3.9)$$

例えば、表 3.4 の組み合わせ 3 を例にとると、測定回数は E が 2 回選択されて基準座標となっているため $n = 2$ となり、合計距離は AP3, AP6, AP7 とユーザ間の推定距離をそれぞれ $d_{3,3}$, $d_{3,6}$, $d_{3,7}$ とすると、 $d_{3,3} + d_{3,6} + d_{3,7}$ となる。

次に最確値について説明する。各アクセスポイントの組み合わせにおける軽重率を $p_i [i = 1, 2, 3 \dots]$ 、基準座標を $P_i (X_i, Y_i, Z_i) [i = 1, 2, 3 \dots]$ とすると、式 3.10~3.12 で定義される。なお、最確値で得られた X, Y, Z が最終的なユーザの推定座標となる。

$$X = \frac{p_1 X_1 + p_2 X_2 + p_3 X_3 \dots}{p_1 + p_2 + p_3 \dots} \tag{3.10}$$

$$Y = \frac{p_1 Y_1 + p_2 Y_2 + p_3 Y_3 \dots}{p_1 + p_2 + p_3 \dots} \tag{3.11}$$

$$Z = \frac{p_1 Z_1 + p_2 Z_2 + p_3 Z_3 \dots}{p_1 + p_2 + p_3 \dots} \tag{3.12}$$

ここで、それぞれ p_2 と p_3 を組み合わせ 2, 組み合わせ 3 の軽重率とし、組み合わせ 1 の基準座標を $P_1 (X_1, Y_1, Z_1)$ 、組み合わせ 2 の基準座標を $P_2 (X_2, Y_2, Z_2)$ 、組み合わせ 3 の基準座標を $P_3 (X_3, Y_3, Z_3)$ とすると、 X は式 3.13 として表される。同様にして、 Y, Z を求めることで、最終的に 1 点の推定座標を求めることができる。

$$X = \frac{p_1 X_1 + p_2 X_2 + p_3 X_3}{p_1 + p_2 + p_3} \tag{3.13}$$

3.2.3 コンテキスト推定機構

本項では、図 3.7 におけるコンテキスト推定機構の詳細について説明する。まず、具体的な処理の流れを図 3.14 に示す。

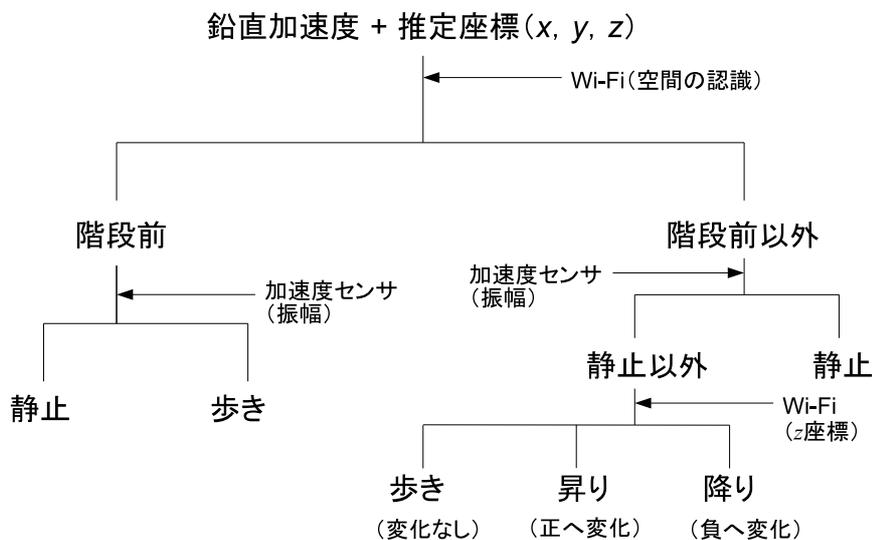


図 3.14 コンテキスト推定機構における移動コンテキスト分類の流れ

コンテキスト推定機構では、鉛直方向の加速度とユーザの推定座標を利用して、各コンテキスト認識を行っている。図 3.14 に示す通り、最初は Wi-Fi を利用して空間の分類を行っている。方法としては、あらかじめ階段前空間の 3 次元座標を定義しておき、Wi-Fi アクセスポイントから推定されたユーザの座標が階段前空間の内側であれば「階段前」空間であるという認識を行っている。一方、外側であれば階段前以外の空間であるという認識を行う。なお、具体的な階段前空間の座標定義については、第 4 章にて説明する。

続いて、「階段前」空間では加速度センサを利用して「階段前×静止」、「階段前×歩き」に分類を行い、階段前以外の空間では「any (階段前以外) ×静止」と静止以外に分類している。ここでは、3.1.3 項における「any (どこでも) ×静止」と「平坦×歩き」を分類する際と同様の処理を行っているため、ウィンドウサイズ (WS) や閾値 (max-in, min-in) の設定を変動させることも可能である。

さらに、静止以外ではユーザの z 座標 (鉛直方向上向きを正) の変動から「平坦 (廊下) ×歩き」、「階段×昇り」、「階段×降り」に分類される。本研究の実験環境では、階段 1 段分の長さが 17cm であったため、前回と現在の推定 z 座標を比較して 17cm 以上増加していれば「階段×昇り」という認識を、17cm 以上減少している場合は「階段×降り」という認識を、17cm 未満の変化であれば「平坦 (廊下) ×歩き」という認識をそれぞれ行う。

第4章

評価実験

第4章では、「加速度センサを用いたパラメータ変動手法」と、「加速度センサ + Wi-Fi を用いた手法」に分けて説明していく。「加速度センサを用いたパラメータ変動手法」では、まずは著者の歩行データを使用して各コンテキスト認識の再現率・適合率を求め、評価を行う。その後、他の被験者から得られた歩行データから、再度システムの評価を行う。次に「加速度センサ + Wi-Fi を用いた手法」では、「加速度センサを用いたパラメータ変動手法」と同様に、被験者の歩行データから再現率・適合率を求め、システムを評価する実験と、推定座標の精度を評価する実験の2つを実施した。

4.1 加速度センサを用いたパラメータ変動手法に関する実験

4.1.1 実験環境

自身の加速度を記録する端末を持ち歩く者と、実際にコンテキストが切り替わるタイミングを計測する者の2名で行い、大学校舎1階から4階を10往復して、階段昇り、降りのデータを10本ずつ得た。なお、役割の交代を行わなかったため、今回取得した歩行データはいずれも同じ人物（著者）のものである。実験時は、加速度センサが内蔵されているAndroid端末（本実験ではASUS社のEee Pad TF201を使用）を持ち歩き、データの取得を行った。計測する場所として利用した歩行ルートを図4.1に示す。但し、「加速度センサを用いたパラメータ変動手法に関する実験」では、平坦におけるコンテキストを廊下と階段前に区別せずに「平坦×歩き」として扱う。昇りの歩行データ取得ルートは、図4.1の1F廊下に示された赤丸から始まり、赤線で引かれたルートを辿って4Fの赤丸が終点となっている。具体的には、1F廊下（静止）→1F廊下（歩き）→1F階段前（歩き）→階段（昇り）→1.5F階段前（歩き）→階段（昇り）→2F階段前（歩き）→・・・→3.5F階段前（歩き）→階段（昇り）→4F階段前（歩き）→4F廊下（歩き）→4F廊下（静止）となっており、途中で立ち止まることはない。一方、降りの歩行ルートは昇りと逆で、4Fの赤丸が起点となり、1Fの赤丸が終点となる。また、階段はいずれも階段（降り）となる。

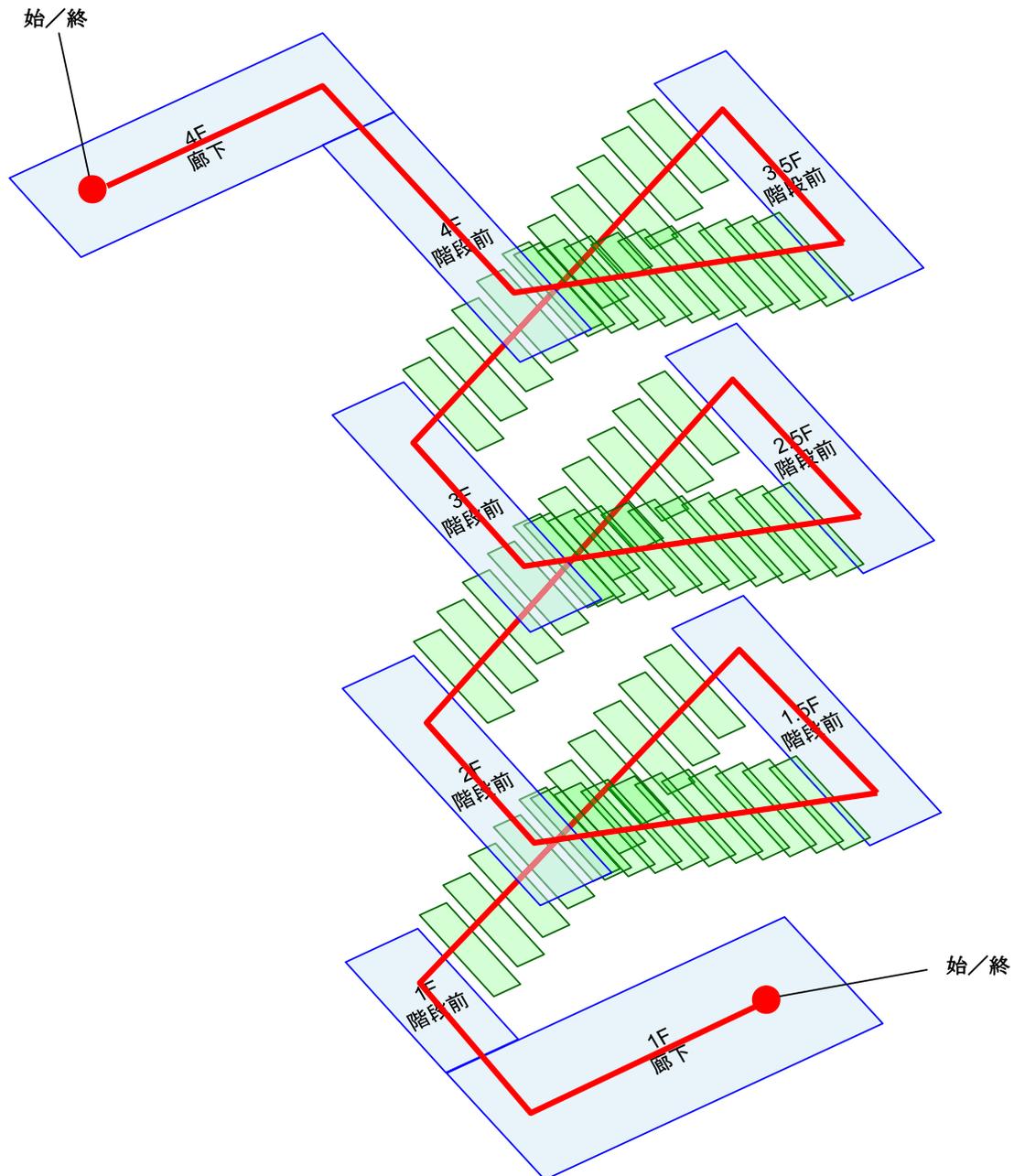


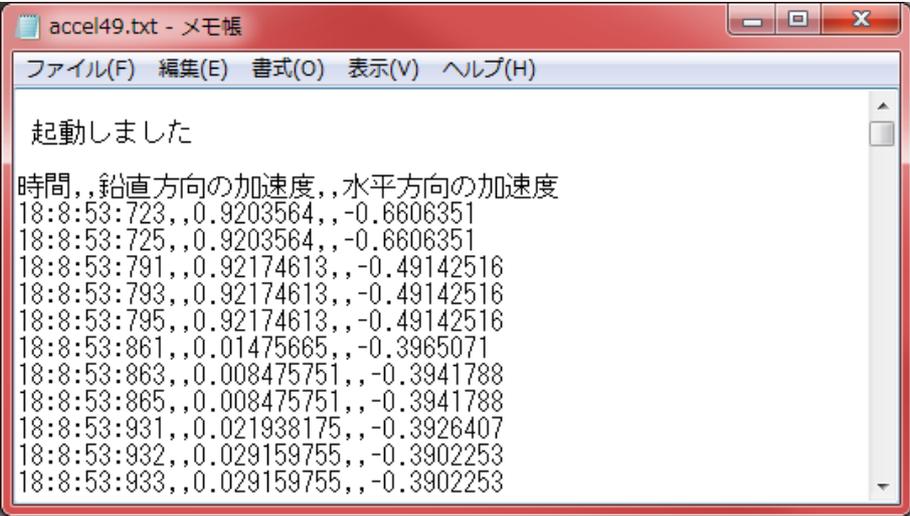
図 4.1 加速度センサを用いたパラメータ変動手法における歩行ルート

取得するデータは、鉛直方向、水平方向の加速度であり、テキストファイルとして出力される。加速度を記録したテキストファイルを図 4.2 に、コンテキストが切り替わるタイミングを記録したテキストファイルを図 4.3 にそれぞれ示す。図 4.2 のファイルは自動的に書き込まれるが、図 4.3 のファイルは、歩行者のコンテキストが切り替わるタイミングを見計らって、観測者が端末の画面上にあるボタンを押した時間が書き込まれるようになっている。図 4.2 のテキストファイルを PC 上に実装した本システムに流すと、5 つのパラメータ (WS, max-out, max-in, min-out, min-in) に応じてコンテキスト認識が行われ、図 4.4 のような認識したコンテキストと時間を記したテキストファイルを出力する。その後、実際のタイミングを計測した図 4.3 に示したテキストファイル (正解セット) と、図 4.4 のテキストファイルを統合し、

時刻順に整理した図 4.5 を出力する。最後に，図 4.5 に示したテキストファイルから，再現率 (Recall) と適合率 (Precision) を出力していく。例えば，「コンテキストが変化しました (歩き開始)」に示された時刻から次の「コンテキストが変化しました (階段降り)」までの正解コンテキストは「平坦×歩き」となり，その間にシステムが正しく認識した割合はどの程度であるかを評価するのが再現率である。一方，適合率は，システムが認識した時間の内，正しく認識した割合はどの程度であるかを評価する。なお，再現率・適合率の計算をし易くするため，UNIX 時刻に直して統合している。再現率・適合率はそれぞれ以下の式で計算される。

$$\text{再現率} := \frac{\text{システムが認識した正解合計時間}}{\text{実際のコンテキストの合計時間}}$$

$$\text{適合率} := \frac{\text{システムが認識した正解合計時間}}{\text{システムが認識した合計時間}}$$



```

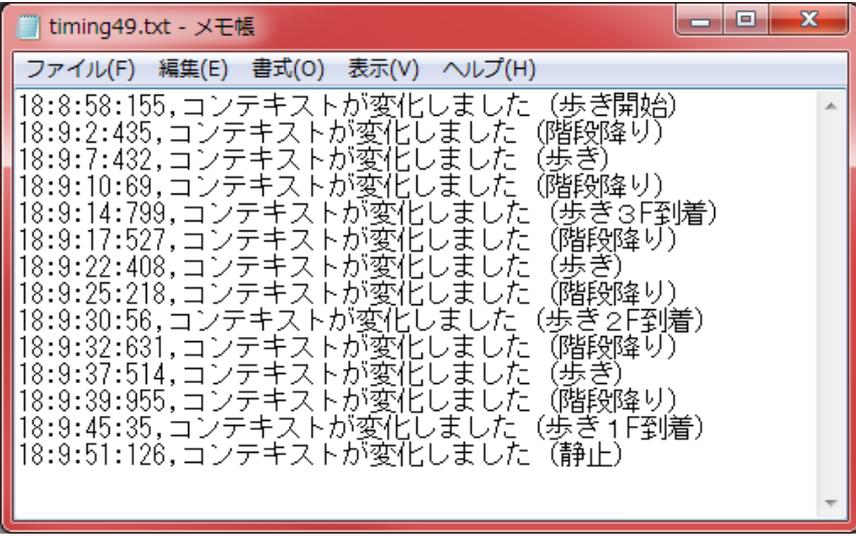
accel49.txt - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)

起動しました

時間,,鉛直方向の加速度,,水平方向の加速度
18:8:53:723,,0.9203564,,-0.6606351
18:8:53:725,,0.9203564,,-0.6606351
18:8:53:791,,0.92174613,,-0.49142516
18:8:53:793,,0.92174613,,-0.49142516
18:8:53:795,,0.92174613,,-0.49142516
18:8:53:861,,0.01475665,,-0.3965071
18:8:53:863,,0.008475751,,-0.3941788
18:8:53:865,,0.008475751,,-0.3941788
18:8:53:931,,0.021938175,,-0.3926407
18:8:53:932,,0.029159755,,-0.3902253
18:8:53:933,,0.029159755,,-0.3902253

```

図 4.2 accel.txt の例



```

timing49.txt - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)

18:8:58:155,コンテキストが変化しました (歩き開始)
18:9:2:435,コンテキストが変化しました (階段降り)
18:9:7:432,コンテキストが変化しました (歩き)
18:9:10:69,コンテキストが変化しました (階段降り)
18:9:14:799,コンテキストが変化しました (歩き3F到着)
18:9:17:527,コンテキストが変化しました (階段降り)
18:9:22:408,コンテキストが変化しました (歩き)
18:9:25:218,コンテキストが変化しました (階段降り)
18:9:30:56,コンテキストが変化しました (歩き2F到着)
18:9:32:631,コンテキストが変化しました (階段降り)
18:9:37:514,コンテキストが変化しました (歩き)
18:9:39:955,コンテキストが変化しました (階段降り)
18:9:45:35,コンテキストが変化しました (歩き1F到着)
18:9:51:126,コンテキストが変化しました (静止)

```

図 4.3 timing.txt の例

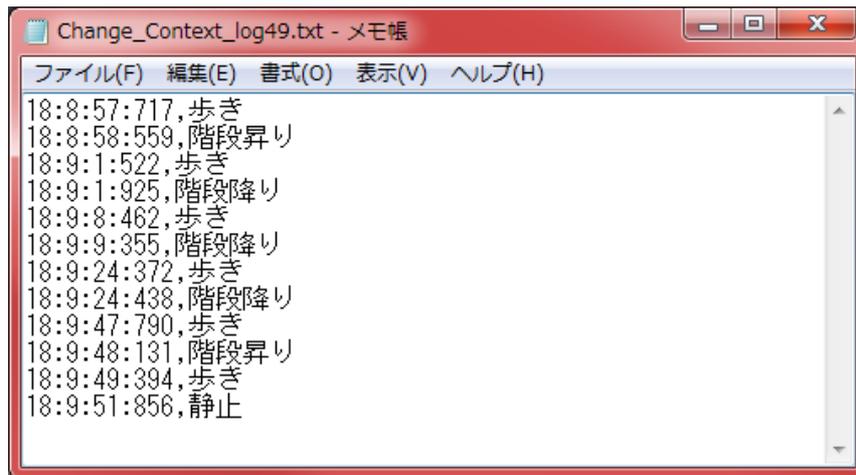


図 4.4 Change.Context.log.txt の例

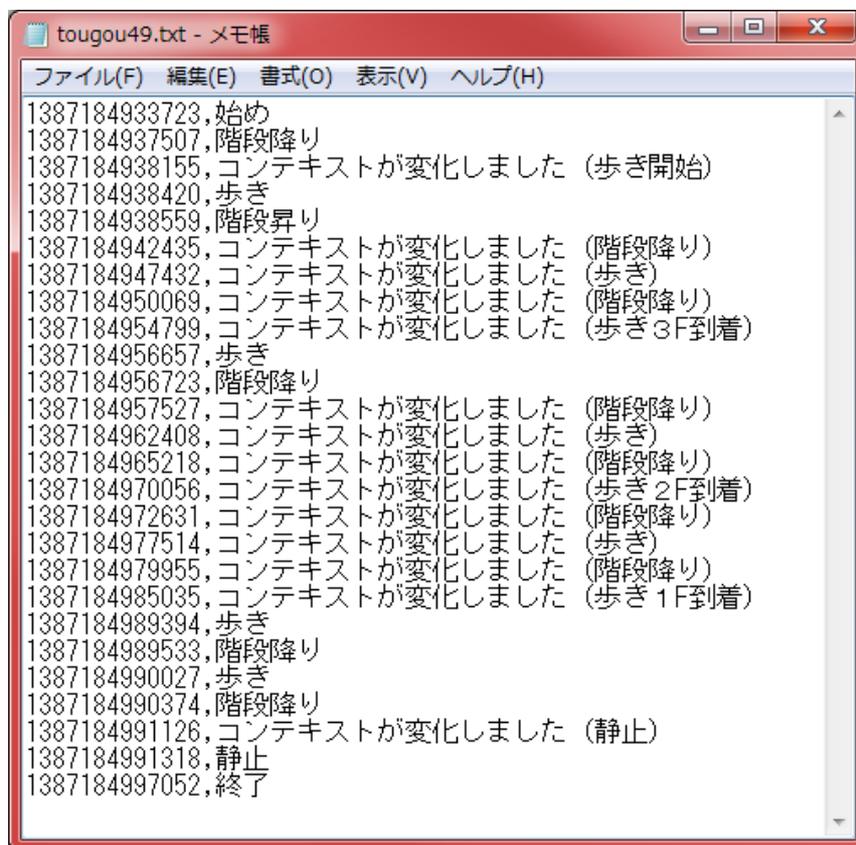


図 4.5 tougou.txt の例

4.1.2 実験（著者の歩行データを使用）

まず、本システムの初期設定パラメータ（WS=30, max-out=2.0, max-in=0.5, min-out=-2.0, min-in=-0.5）による認識割合を表 4.1, 表 4.2 に示す。

表 4.1 初期設定における再現率

システム \ 実際	(any) 静止	(平坦) 歩き	(階段) 昇り	(階段) 降り
(any) 静止	71.28%	26.02%	1.73%	0.47%
(平坦) 歩き	0.01%	12.52%	67.93%	19.54%
(階段) 昇り	0.00%	0.00%	98.28%	1.72%
(階段) 降り	0.00%	1.00%	59.22%	39.78%

表 4.2 初期設定における適合率

システム \ 実際	(any) 静止	(平坦) 歩き	(階段) 昇り	(階段) 降り
(any) 静止	99.92%	0.08%	0.00%	0.00%
(平坦) 歩き	33.82%	64.15%	0.00%	2.03%
(階段) 昇り	0.18%	32.05%	37.78%	29.99%
(階段) 降り	7.90%	52.40%	4.31%	35.38%

パラメータ最適化

続いて、再現率、適合率が最大となるようなパラメータを求める実験を行った。しかし、表 4.1、表 4.2 にあるような複数の数値では比較が難しいことから、1つの数値に集約を行って比較を容易にするための処理を行った。その際、表の太字部分を使って重み付け（静止×0.2 + 歩き×0.2 + 昇り×0.3 + 降り×0.3）を行うことで、再現率、適合率を1つの数値にそれぞれまとめ、さらにそれらを統合するために F 値を利用した。この F 値が最大となるようなパラメータを最適化されている状態とする。パラメータの可動範囲は、

- $20 \leq WS \leq 50$ (5 刻み)
- $1.5 \leq \text{max-out} \leq 4.0$ (0.1 刻み)
- $0.1 \leq \text{max-in} \leq 1.0$ (0.1 刻み)
- $-4.0 \leq \text{min-out} \leq -1.5$ (0.1 刻み)
- $-1.0 \leq \text{min-in} \leq -0.1$ (0.1 刻み)

とする。パラメータを変動させた結果、F 値が最大となったパラメータの組み合わせは、 $WS=40$, $\text{max-out}=3.9$, $\text{max-in}=1.0$, $\text{min-out}=-2.1$, $\text{min-in}=-1.0$ であり、F 値は 0.73 となった。なお、初期設定パラメータにおける F 値は 0.56 であった。最適化パラメータ時の再現率を表 4.3 に、適合率を表 4.4 に示す。

表 4.3 パラメータ最適化後の再現率

システム \ 実際	(any) 静止	(平坦) 歩き	(階段) 昇り	(階段) 降り
(any) 静止	88.83%	9.48%	0.00%	1.69%
(平坦) 歩き	0.98%	37.49%	15.82%	45.71%
(階段) 昇り	0.00%	0.24%	70.49%	29.28%
(階段) 降り	0.00%	3.01%	0.00%	96.99%

表 4.4 パラメータ最適化後の適合率

システム \ 実際	(any) 静止	(平坦) 歩き	(階段) 昇り	(階段) 降り
(any) 静止	95.64%	4.36%	0.00%	0.00%
(平坦) 歩き	5.18%	91.61%	0.47%	2.74%
(階段) 昇り	0.00%	23.93%	76.07%	0.00%
(階段) 降り	0.21%	36.48%	29.34%	33.98%

ウィンドウサイズ (WS) 以外のパラメータ変動

各々のパラメータが再現率・適合率に及ぼす影響を検証するため、パラメータ最適化後のパラメータを使って、一部のパラメータを変動させた。ここでは、階段に影響を及ぼすパラメータである max-out と min-out のみを変動させた様子を図 4.6~4.11 にそれぞれ示す。なお図中の×印は、パラメータ最適化後に関するパラメータの位置を表している。また、max-in, min-in のパラメータは、静止と歩きの認識には関与したが、研究の重点である昇りと降りには影響を及ぼさなかったため、本稿では割愛する。

まず「階段」空間に関しては、図 4.6 から、昇りの再現率が高くなるのは、max-out, min-out の両方が小さいときであるとわかる。これは、昇りに関する閾値の条件が緩くなり、かつ降りに関する閾値の条件が厳しくなることで、全体的に昇りと認識し易くなったものと考えられる。本システムは加速度の振幅が正に大きく振れ、ウィンドウ内の最大値が閾値の max-out を超えることで昇りと認識するため、max-out が小さいほど最大値が超え易くなり、「階段×昇り」と認識をし易くなる。もし、max-out の閾値が高すぎる場合は、ウィンドウ内の最大値が max-out を超えきれないため、昇りと認識され難くなり、再現率が低下する。また、min-out の閾値が大きい場合、max-out の閾値を超えきれずその後の加速度が負に大きく振れて、ウィンドウ内の最小値が min-out を下回り易くなり、降りと誤認識する可能性が高くなる。min-out の閾値を小さくし、降りと認識しづらくすることで、その後、再び加速度が正に振れたときに昇りと認識させ易くなり、結果昇りの再現率が向上する。

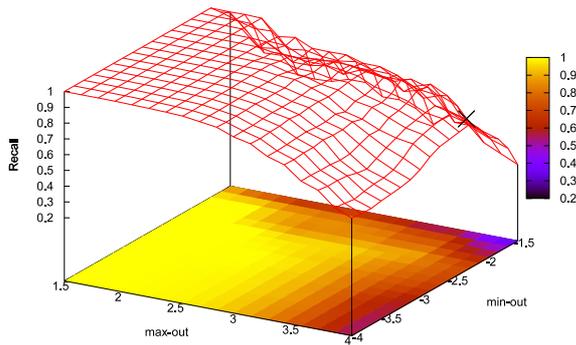


図 4.6 WS=40,max-in=1.0,min-in=1.0
における「階段×昇り」の再現率

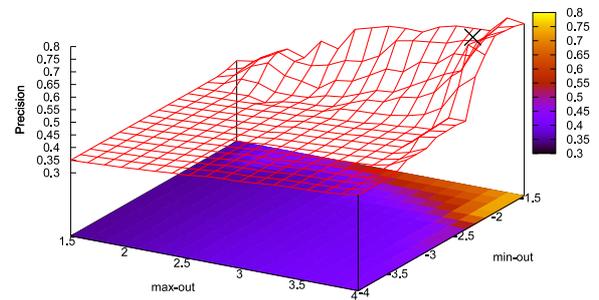


図 4.7 WS=40,max-in=1.0,min-in=1.0
における「階段×昇り」の適合率

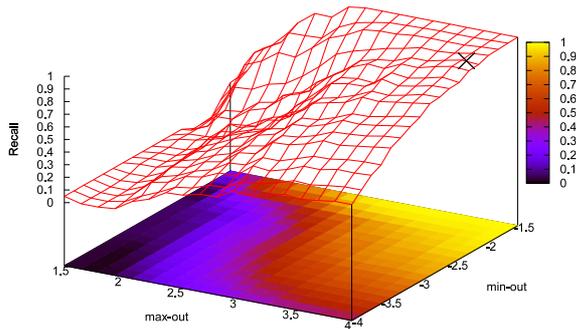


図 4.8 WS=40,max-in=1.0,min-in=1.0
における「階段×降り」の再現率

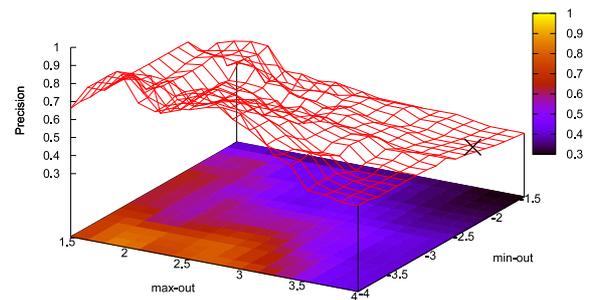


図 4.9 WS=40,max-in=1.0,min-in=1.0
における「階段×降り」の適合率

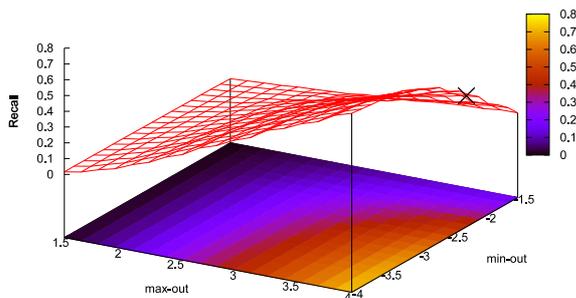


図 4.10 WS=40,max-in=1.0,min-in=1.0
における「平坦×歩き」の再現率

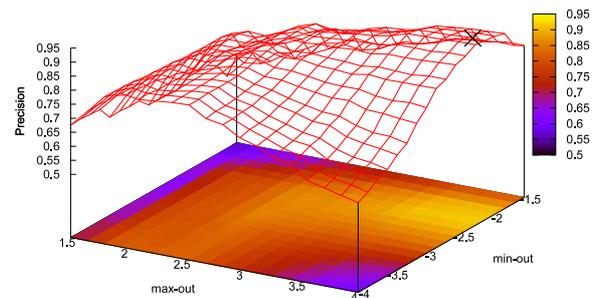


図 4.11 WS=40,max-in=1.0,min-in=1.0
における「平坦×歩き」の適合率

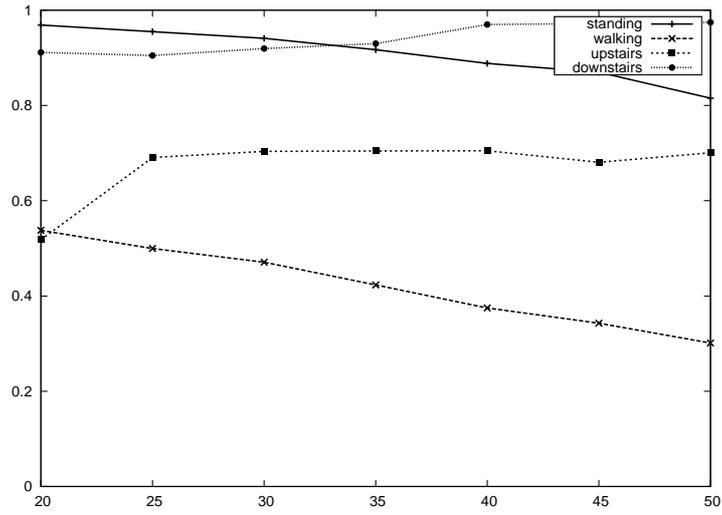


図 4.12 WS のみ変動させたときの再現率

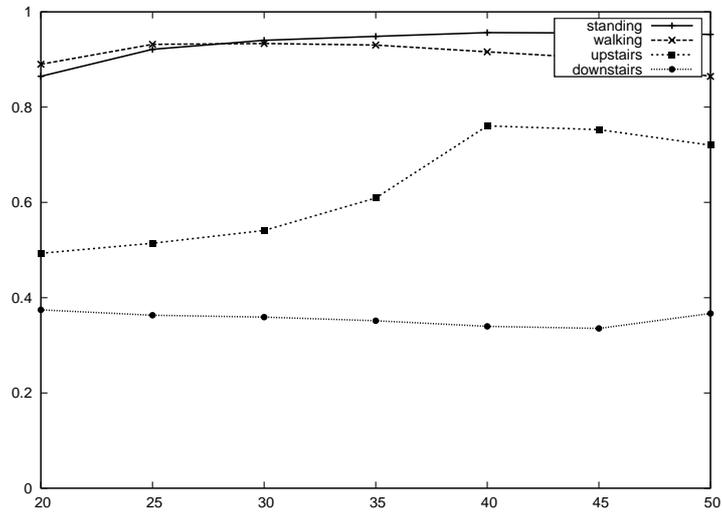


図 4.13 WS のみ変動させたときの適合率

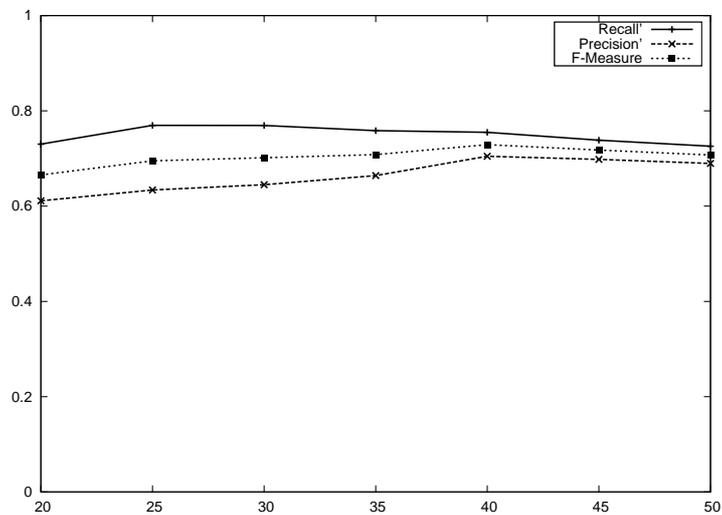


図 4.14 WS のみ変動させたときの F 値

上記は、降りの再現率でも同様のことが考えられる。図 4.8 から、降りの再現率が高くなるのは、max-out, min-out の両方が大きいときであり、昇りの再現率とは真逆となっている。これは min-out が大きいほど、降りの閾値である min-out を下回り易くなるので、降りの再現率が上昇するものと考えられる。また、max-out が大きいほど、昇りと認識しづらくなるので、昇りの誤認識が減ることから、降りの再現率が上昇すると考えられる。一方、適合率の方は、図 4.6 と図 4.7, 図 4.8 と図 4.9 をそれぞれ比較することで、再現率と適合率がトレードオフの関係になっていることがわかる。これは、例えば昇りの場合、閾値を緩くして昇りと認識させ易くすると、取りこぼしが少なくなり再現率が高くなるが、昇り以外のコンテキストも昇りと認識し易くなるため、適合率が下がってしまうものと考えられる。

次に、「平坦×歩き」の再現率に関しては、図 4.10 から、max-out が大きく、min-in が小さい、つまり絶対値が大きくなるほど再現率が高くなるということがわかる。これは、歩きと認識する条件が緩くなるからと考えられる。昇りと降りに関する閾値の条件が厳しくなることで、加速度がある程度大きい振幅となっても、昇り、降りと認識され難くなり、結果「平坦×歩き」と認識し易くなる。一方、適合率の方は、図 4.11 から、再現率が最も高くなるパラメータのとき、再現率が低くなっていることがわかる。また、max-out, min-out の絶対値が 2.5 付近のとき、適合率が高くなる傾向であるとわかる。これは、絶対値が小さすぎると「階段×昇り」や「階段×降り」と認識し易くなるため、「平坦×歩き」の認識がされ難くなり、逆に絶対値が大きすぎると歩きの認識が増えすぎるため、適合率が下がると考えられる。しかし、適合率が最大となるのは、max-out が 4.0 付近のときであるため、昇りと認識し難くすることで、歩きの適合率を高められることがわかる。

以上をまとめると、max-out と min-out を両方大きくすると昇りの適合率と降りの再現率が向上し、両方小さくすると昇りの再現率と降りの適合率が向上する。これらは、それぞれトレードオフの関係であるため、パラメータの変化だけでは昇り、降りの再現率、適合率を両立させることは難しい。特に昇りの適合率と降りの適合率は、トレードオフの傾向が顕著であるため、適合率を重視するユーザが、昇り、降りの両方を自己抑制したい場合、現状のシステムでは実現が難しい。しかし、表 4.3 からわかる通り、F 値が最大となるときの再現率は昇り、降り共に一定の水準を保っているため、再現率を重視するユーザであれば、アプリケーションとして提供できる。

ウィンドウサイズ (WS) のみ変動

次に、WS のみ変動させる実験を行った。ウィンドウは加速度を格納する配列であり、WS が小さいほど古い加速度を格納しなくなるため、最大値、最小値の更新頻度が多くなり、コンテキストの切り替わりが速くなる。しかし、WS が小さすぎると、例えば昇りのアクションをしているときに、加速度が正から負に転ずる部分では一時的に最大値と最小値の絶対値が小さくなるため、昇り→歩き→降り→静止→昇りなどと細切れな認識をしてしまう恐れがある。そこで WS が与える影響を調べるため、WS のみを変動させたときの各コンテキスト毎の再現率を図 4.12 に、適合率を図 4.13 に、重み付けした再現率、適合率及び F 値を図 4.14 にそれぞれ

れ示す。

まず「any × 静止」に関しては、図 4.12 から、WS が小さいほど再現率が高くなることがわかる。これは、歩き→静止への切り替わりが速くなるからだと考えられる。一方、図 4.13 から WS が大きいほど適合率が上がっていることがわかるが、これは再現率とのトレードオフの関係が影響していると考えられる。

次に「平坦 × 歩き」に関しては、図 4.12 から、WS が小さいほど歩きの再現率が上がっていることがわかる。この理由には、階段→階段前のときに切り替わりが遅くなることで、階段前に入ったばかりの部分を昇り、または降りと誤認識することが考えられる。一方、適合率はトレードオフの関係から WS が大きいほど適合率も上がると考えられるが、図 4.13 からわかるように、実際はあまり変化が見られなかった。これは、階段前→階段のときに切り替わりが遅くなることで、階段に入ったばかりの部分を「平坦 × 歩き」と誤認識をして、歩きの適合率が下がることが影響していると考えられる。つまり、トレードオフと切り替わりの遅さの影響が互いに均衡しているため、全体的に適合率の変化が見られなかったと考えられる。

「階段 × 昇り」の再現率に関しては、図 4.12 からわかるように、WS が 20 から 25 に変動すると大きく再現率が増加し、25 以上ではあまり変化しない結果となった。理由として、WS が 20 のときは、静止と歩きの誤認識による細切れが起き易いため再現率が低くなるが、WS が 25 以上では細切れが起き難くなるため、再現率が高くなったと考えられる。一方、適合率に関しても同様に、図 4.13 から、WS が大きい方が適合率が高くなる傾向であることがわかる。これは、WS が大きくなるにつれて階段では、昇り→歩き→降りのような細切れした認識が減ることで、途中で降りと誤認識することも減るからであると考えられる。

「階段 × 降り」に関しては、図 4.12 からわかるように、WS が大きいほど再現率が上がっているが、これは昇りと同様に、細切れが起き難くなるからであると考えられる。また、図 4.13 が示す通り、WS を変動させても、適合率にはあまり影響を与えなかった。

最後に重み付けした再現率、及び、適合率、これらを統合した F 値については、図 4.14 からわかるように、WS による大きな変化は見られなかった。つまり、WS のパラメータ変動は、全体として再現率・適合率を観察した場合、あまり影響を与えないことが窺える。

特定のコンテキストに特化

パラメータはトレードオフの関係にあるため、階段コンテキスト認識における精度の両立は難しいが、ユーザが自己抑制をしたいコンテキストに合わせ逐次パラメータの変動を行うことで特定のコンテキスト認識に特化できる可能性がある。そこで、実際にパラメータを変動させて各コンテキストに特化させた際の F 値を表 4.5 に示す。但し、表中の再現率・適合率は、F 値が最大のときの値であり、再現率・適合率が最大の値とは限らない。

表 4.5 特定のコンテキストに特化したときの最大値 (F 値)

移動コンテキスト	再現率	適合率	F 値
「any × 静止」	91.81%	97.40%	94.52%
「平坦 × 歩き」	70.71%	80.97%	75.49%
「階段 × 昇り」	96.68%	60.18%	74.18%
「階段 × 降り」	66.01%	68.14%	67.06%

表 4.5 から、「any × 静止」の F 値が非常に高い結果となったことがわかる。他のコンテキストに関しても、「階段 × 降り」の F 値が他に比べて悪いものの、全体的には約 70% 以上の結果が得られ、パラメータの変動による特定のコンテキストへの特化は一定の効果があった。

また、表 4.5 の特徴として、「階段 × 昇り」の F 値は 74.18% であるが、再現率は 96.68% という高い精度が得られている点が挙げられる。再現率が高いということは、実際のコンテキストに対し、システムが正しく認識できている割合が高いということであり、言い換えれば必要な場所で必要な警告を出せていると言える。一方、適合率が高い場合は、誤認識が低くなるため unnecessary 警告を見ないで済むようになる。しかし、本研究の目的であるモバイル自己抑制により危険性を減らすという観点から鑑みた場合、たとえ誤認識により unnecessary 警告が多くても、再現率さえ高ければ高い精度でもって必要な場所で必要な警告を出せるアプリケーションとして提供できるため、ユーザにとっては安全性が高く、有用性が高いと考えられる。例えば「階段 × 昇り」が危険であるから自己抑制したいと考えるユーザの場合、誤認識が多くても危険な場所で危険であるという警告を発してもらえらるなら、ユーザの危険防止に役立つことが期待できる。これは、他のコンテキストに関しても同様であり、再現率を高くできれば危険防止に役立つだろう。

そこで、パラメータの変動によって各コンテキストに特化させたとき、F 値の代わりに再現率・適合率を最大としたときの結果を表 4.6 に示す。但し、表中の再現率と適合率は別々のパラメータによって求められた値であり、同時に満たしている数値ではない。

表 4.6 特定のコンテキストに特化したときの最大値 (再現率・適合率)

移動コンテキスト	再現率	適合率
「any × 静止」	97.02%	100.00%
「平坦 × 歩き」	85.50%	98.65%
「階段 × 昇り」	100.00%	77.13%
「階段 × 降り」	99.59%	96.43%

表 4.6 の結果から、再現率に関しては、いずれのコンテキスト認識においても高い精度が得られている。特に危険性が高い「階段 × 昇り」、「階段 × 降り」の認識精度が高いため、階段におけるモバイル自己抑制に貢献できると考えられる。適合率に関しては、「階段 × 昇り」の精

度が他に比べて低いものの、全体的には高い結果が得られた。再現率・適合率共に高い精度が得られたので、どちらかを重視するユーザであれば精度が高いモバイル自己抑制アプリケーションを提供できる。

実用的なケースでの認識精度

上記では、例えば「階段×昇り」のみをモバイル自己抑制する、といった各コンテキスト毎での使用を想定した精度を求めていた。しかし、ユーザによっては「階段×昇り」と「階段×降り」の両方を区別する必要がないユーザも考えられる。そこで、昇りと降りを区別しない場合の認識精度（ケース1）と、歩き、昇り、降りの区別をしない場合の認識精度（ケース2）を考えていく。ケース1の再現率を表4.7に、適合率を表4.8に示し、ケース2の再現率を表4.9に、適合率を表4.10にそれぞれ示す。なお、パラメータにはF値が最大となる $WS=40$, $max-out=3.9$, $max-in=1.0$, $min-out=-2.1$, $min-in=-1.0$ を利用した。また、ケース1では、重み付けを静止×0.2 + 歩き×0.2 + (昇り+降り)×0.6としてF値を計算すると0.83となり、ケース2では重み付けを静止×0.2 + (歩き+昇り+降り)×0.8として計算するとF値は0.97となった。

表 4.7 昇り + 降りの再現率

システム \ 実際	(any) 静止	(平坦) 歩き	(階段) 昇り + 降り
(any) 静止	88.83%	9.48%	1.69%
(平坦) 歩き	0.98%	37.49%	61.53%
(階段) 昇り + 降り	0.00%	1.71%	98.29%

表 4.8 昇り + 降りの適合率

システム \ 実際	(any) 静止	(平坦) 歩き	(階段) 昇り + 降り
(any) 静止	95.64%	4.36%	0.00%
(平坦) 歩き	5.18%	91.61%	3.20%
(階段) 昇り + 降り	0.00%	27.53%	72.36%

表 4.9 歩き + 昇り + 降りの再現率

システム \ 実際	(any) 静止	(平坦) 歩き + (階段) 昇り + (階段) 降り
(any) 静止	88.83%	11.17%
(平坦) 歩き + (階段) 昇り + (階段) 降り	0.49%	99.51%

表 4.10 歩き + 昇り + 降りの適合率

システム \ 実際	(any) 静止	(平坦) 歩き + (階段) 昇り + (階段) 降り
(any) 静止	95.64%	4.36%
(平坦) 歩き + (階段) 昇り + (階段) 降り	2.11%	97.89%

まず、ケース 1 に関しては、「平坦×歩き」の再現率以外は、比較的高い精度が得られている。特に危険性が高いと考えられる「階段×(昇り+降り)」では、再現率が 98.29% という高い結果が得られ、表 4.3 における「階段×昇り」を「階段×降り」と認識してしまう失敗を補った結果となった。適合率に関しても、今度は表 4.4 における「階段×降り」を「階段×昇り」と誤認識するミスを補って、72.36% という結果が得られている。但し、「階段×昇り」に比べると精度が下がってしまっている。

また、各コンテキスト毎での精度とケース 1 を比べると、F 値が 0.72 から 0.83 へと大きく上がっていることから、各コンテキスト毎では「階段×昇り」と「階段×降り」の区別が上手くできていなかったことが窺える。しかし、「平坦×歩き」の再現率に関しては、実際は歩きであるのに階段であるという認識が 61.53% と多くなっている。そのため、「平坦×歩き」では自己抑制をして、階段では自己抑制をしたくないと考えているユーザの場合は、61.53% の確率で歩きの危険防止ができないことになる。このような場合には、今回利用した最適化パラメータの代わりに「平坦×歩き」の再現率を向上させるようなパラメータに変動することで、ユーザの希望に適した精度にできると考えられる。

次にケース 2 に関しては、どのコンテキストにおいても再現率・適合率共に高い精度が得られた。ケース 1 とケース 2 を比較すると、F 値は 0.83 から 0.97 へと大きく上昇していることから、「平坦×歩き」と「階段×(昇り+降り)」の区別が上手くできていなかったことが窺える。「any × 静止」の再現率は 88.83% であり、他と比べると低く感じるが、十分高い精度であ

ると言える。また、「any × 静止」で自己抑制をしたい、つまり、立ち止まった状態でのモバイル端末の操作は危険だからやめよう、というユーザはかなり少数であると考えられるので、仮に再現率が低くても問題ないだろう。結論としては、ケース2のように、静止以外の全てのコンテキストで自己抑制をしたいと考えるユーザにとっては、ほぼ完璧な精度が得られていると言える。

4.1.3 実験（他の被験者から得られた歩行データを使用）

第4.1.2項では、著者個人の歩行データから、逐次パラメータの変動を行うことで特定のコンテキスト認識に特化できることが明らかになった。しかし、著者個人の歩行データから得られた知見であるため、一般的な事実とは言えない。そこで、その他9人の被験者から得られた歩行データを使用し、パラメータの変動によるコンテキスト認識特化の可能性を模索する。なお、パラメータの可動範囲や重み付け、F値といった評価方法も前節と同様の基準で行う。実験方法は、被験者に加速度を記録する端末を持ち歩いてもらい、著者は被験者のコンテキストが切り替わるタイミングを計測した。実験場所は、第4.1節と同様に図4.1のルートを歩いてもらった。但し、被験者の負担を考慮して、10往復から3往復に変更して実験を行い、階段昇り、降りの歩行データを3本ずつ得た。

パラメータ最適化

まず、第4.1.2項における実験と同様に、各種パラメータを変動させてF値が最大となるような各被験者の最適化パラメータを求め、結果を表4.11にまとめた。

表 4.11 各被験者の最適化パラメータ

被験者	WS	max-out	max-in	min-out	min-in	F 値
No.1	35	2.4	1.0	-3.8	-1.0	61.04%
No.2	45	1.5	0.9	-1.6	-0.7	69.25%
No.3	25	4.0	1.0	-4.0	-1.0	62.01%
No.4	45	1.8	1.0	-1.9	-1.0	75.45%
No.5	50	1.7	1.0	-1.9	-0.5	59.65%
No.6	20	3.5	0.8	-3.9	-1.0	62.46%
No.7	25	2.0	0.9	-2.3	-0.9	71.02%
No.8	35	1.5	1.0	-2.0	-0.9	72.68%
No.9	20	3.6	1.0	-4.0	-0.9	66.68%

表4.11から、どのユーザもF値は約60%以上を保っている。しかし、最適化パラメータは人によって異なっていて、全てのユーザに対して最適となるような共通のパラメータは得ら

れなかった。それでも、「any×静止」と「平坦×歩き」を区別するパラメータである max-in と min-in に関しては、いずれも絶対値が大きいため、2つの移動コンテキストを区別するならば、大き目の値を設定しておくのが良いと考えられる。また、F 値が7割程度の高い結果が得られた No.2, 4, 7, 8 の被験者（グループ A）は、WS 以外の最適化パラメータが近い値となっていることが窺える。但し、No.5 の被験者も近いパラメータとなっているが F 値が小さくなっているため、例外もある。だが、No.5 の被験者は min-in の値が -0.5 であり、他の被験者に比べて絶対値が小さいのが特徴的であるとも考えられるので、全体的なパラメータが似ていても、min-in の値によってはグループ A と区別できる可能性がある。

その他には、No.3, 6, 9 の被験者（グループ B）も比較的にお最適化パラメータが近くなっていることが窺える。グループ B の被験者達は、最適化パラメータの可動範囲の最大付近を示しているため、加速度の振幅が大きく現在の範囲では足りていない可能性が考えられる。そのため、可動範囲を拡張することでさらに F 値が向上する可能性があるだろう。No.1 の被験者は、max-out と min-out の絶対値の差に特徴があり、これは2つのパラメータが逆ではあるものの、著者の最適化パラメータ（WS=40, max-out=3.9, max-in=1.0, min-out=-2.1, min-in=-1.0, F 値=0.73）と類似した傾向を持っていると考えられる。今後は、グループ A, B の分類が適切であるかの検討も含め、さらに被験者を増やし、No.1 の被験者と著者の歩行データはグループ C として分類できる見込みがあるか検討していく。

グループ A とグループ B のように分類ができれば、最適化をせずとも高い精度でモバイル自己抑制アプリケーションを利用できるようになる。例えば高い精度でモバイル自己抑制をしたい場合、パラメータの最適化には機械学習が必要となるが、気軽にモバイル自己抑制を試してみたい人や、アプリケーションを今すぐ使いたい人にとっては不満を感じるだろう。その場合、まずはグループ A 向けの最適化パラメータ（例：WS=40, max-out=1.7, max-in=1.0, min-out=-2.0, min-in=-0.9）で使ってもらって、自分に適合していないと感じたらグループ B 向けの最適化パラメータ（例：WS=20, max-out=3.7, max-in=0.9, min-out=-4.0, min-in=-1.0）で使ってもらおうといった提供方法ができるようになり、ユーザにとっては操作性が高まると考えられる。

特定のコンテキストに特化

続いて、パラメータの変動による特定のコンテキスト認識への特化は、他の被験者においても有用になるのかを検証していく。特定のコンテキストに特化させたときの F 値をまとめた結果を表 4.12 に示す。

表 4.12 各被験者における特定のコンテキストに特化したときの最大値 (F 値)

被験者	「any × 静止」	「平坦 × 歩き」	「階段 × 昇り」	「階段 × 降り」
No.1	86.92%	64.46%	61.03%	59.09%
No.2	96.91%	69.70%	63.96%	78.75%
No.3	86.86%	30.10%	54.18%	66.44%
No.4	81.18%	30.81%	82.70%	90.04%
No.5	72.83%	40.28%	64.70%	72.39%
No.6	82.30%	52.34%	71.11%	68.72%
No.7	95.42%	77.05%	62.03%	77.09%
No.8	97.23%	67.46%	68.67%	87.36%
No.9	85.36%	46.62%	55.90%	64.77%

まず「any × 静止」に関しては、No.5 以外は 80% を超える高い精度となった。また、「階段 × 昇り」と「階段 × 降り」に関しては、ユーザによるばらつきはあるものの、ほぼ 60~80% 程度となり、まずまずの精度と言える結果となった。しかし、「平坦 × 歩き」に関しては、一部の被験者の精度が約 30% という低い精度になっている。そのため、「平坦 × 歩き」をモバイル自己抑制したい場合、ユーザによっては不満を感じるだろう。「平坦 × 歩き」の F 値が低い原因としては、加速度の振幅を利用してコンテキスト認識を行っている現在のアルゴリズムに問題があると考えられる。本研究では、各コンテキストに分類する際、振幅が小さければ「any × 静止」に、振幅が大きければ「階段 × 昇り」または「階段 × 降り」へと分類を行っており、「平坦 × 歩き」は中程度の振幅時に分類される。つまり、他のコンテキスト認識に挟まれる形で閾値が設定されているため、振幅が少しでも大小のどちらかに寄ってしまうと誤認識を起こしてしまう。そのため、「平坦 × 歩き」の F 値が低いユーザの場合、振幅の他に有用となりえる新たなパラメータの導入が必要な可能性がある。例えば、関連研究で利用されているパラメータとして分散が挙げられる [14, 17]。今後は、被験者の歩行データを分析して、新たなパラメータの導入を検討していく。

次に、第 4.1.2 項と同様に、パラメータの変動によって各コンテキストに特化させたとき、F 値の代わりに再現率・適合率を最大としたときの結果を表 4.13 と表 4.14 に示す。

表 4.13 各被験者における特定のコンテキストに特化したときの最大値（再現率）

被験者	「any × 静止」	「平坦 × 歩き」	「階段 × 昇り」	「階段 × 降り」
No.1	94.46%	82.94%	100.00%	97.47%
No.2	99.87%	88.18%	100.00%	100.00%
No.3	76.82%	19.76%	100.00%	89.00%
No.4	91.17%	20.55%	100.00%	100.00%
No.5	64.26%	41.16%	100.00%	100.00%
No.6	93.18%	48.48%	100.00%	100.00%
No.7	93.14%	71.88%	100.00%	100.00%
No.8	99.74%	73.45%	100.00%	100.00%
No.9	74.45%	36.44%	100.00%	100.00%

表 4.14 各被験者における特定のコンテキストに特化したときの最大値（適合率）

被験者	「any × 静止」	「平坦 × 歩き」	「階段 × 昇り」	「階段 × 降り」
No.1	100.00%	69.02%	74.28%	100.00%
No.2	97.92%	91.70%	100.00%	86.25%
No.3	100.00%	68.13%	41.22%	75.69%
No.4	93.92%	85.96%	76.64%	95.49%
No.5	88.42%	46.85%	67.46%	58.47%
No.6	85.40%	60.87%	73.79%	99.41%
No.7	99.42%	96.34%	72.15%	92.88%
No.8	100.00%	96.84%	98.84%	99.04%
No.9	100.00%	82.61%	40.17%	80.72%

まず、「any × 静止」に関しては、どの被験者も再現率・適合率共に高い結果となった。「平坦 × 歩き」に関しては、再現率は No.3, 4, 5, 6, 9 の被験者の結果が芳しくなかった。これは、表 4.12 で「平坦 × 歩き」の F 値が低い人物と一致している。また、適合率は全体的に結果が良いことを考えると、5 人の F 値を下げていているのは再現率であると考えられる。一方で、再現率が 70% や 80% 台の被験者もいるため、結果の良し悪しが両極端になっていることが窺える。著者の結果（「平坦 × 歩き」の再現率は 85.50%）も人数に含めるとすると、再現率を重視して「平坦 × 歩き」をモバイル自己抑制する場合、半数は使えて、もう半数は使えないアプリケーションとなる。しかし、「平坦 × 歩き」を「階段 × 昇り」や「階段 × 降り」と誤認識することで再現率が低いなら、3 つの移動コンテキストを区別せずにモバイル自己抑制を行うことで再現率を高めることができる。

続いて「階段×昇り」に関しては、再現率が全員 100% となったため、再現率を重視するユーザであれば完璧な精度を誇るアプリケーションを提供することができる。適合率の方は結果が低い被験者もいるが、安全性の観点から鑑みれば再現率さえ高ければ問題はないと考えられる。

最後に「階段×降り」は、再現率・適合率共に高い結果となっていて、特に再現率の方は顕著である。他のコンテキストに比べ、「階段×降り」中の歩きスマホは、階段から転げ落ちたりして自分や周囲に大きな怪我を負わせる危険性が高いため、危険防止を考慮するならば、再現率が高い今回の結果は、モバイル自己抑制アプリケーションの有用性が示されたと言える。

実用的なケースでの認識精度

次は、昇りと降りを区別しない場合の認識精度（ケース 1）と、歩き、昇り、降りの区別をしない場合の認識精度（ケース 2）を考えていく。各被験者におけるケース 1 及びケース 2 のときの認識割合を No.1 から順に表 4.15~4.50 に示す。なお、各被験者のパラメータには、それぞれ F 値が最大となる最適化パラメータを用いている。

表 4.15 昇り + 降りの再現率 (No.1)

システム \ 実際	(any) 静止	(平坦) 歩き	(階段) 昇り + 降り
(any) 静止	75.79%	23.97%	0.24%
(平坦) 歩き	0.58%	34.94%	64.48%
(階段) 昇り + 降り	0.00%	2.41%	97.59%

表 4.16 昇り + 降りの適合率 (No.1)

システム \ 実際	(any) 静止	(平坦) 歩き	(階段) 昇り + 降り
(any) 静止	96.20%	2.29%	1.51%
(平坦) 歩き	6.94%	60.43%	32.63%
(階段) 昇り + 降り	0.02%	41.28%	58.70%

表 4.17 歩き + 昇り + 降りの再現率 (No.1)

システム \ 実際	(any) 静止	(平坦) 歩き + (階段) 昇り + (階段) 降り
(any) 静止	75.79%	24.21%
(平坦) 歩き + (階段) 昇り + (階段) 降り	0.36%	99.64%

表 4.18 歩き + 昇り + 降りの適合率 (No.1)

システム \ 実際	(any) 静止	(平坦) 歩き + (階段) 昇り + (階段) 降り
(any) 静止	96.20%	3.80%
(平坦) 歩き + (階段) 昇り + (階段) 降り	2.78%	97.22%

表 4.19 昇り + 降りの再現率 (No.2)

システム \ 実際	(any) 静止	(平坦) 歩き	(階段) 昇り + 降り
(any) 静止	92.53%	1.99%	5.48%
(平坦) 歩き	1.24%	5.71%	93.05%
(階段) 昇り + 降り	0.00%	0.21%	99.79%

表 4.20 昇り + 降りの適合率 (No.2)

システム \ 実際	(any) 静止	(平坦) 歩き	(階段) 昇り + 降り
(any) 静止	97.09%	2.91%	0.00%
(平坦) 歩き	8.95%	80.75%	10.30%
(階段) 昇り + 降り	4.28%	55.58%	40.14%

表 4.21 歩き + 昇り + 降りの再現率 (No.2)

システム \ 実際	(any) 静止	(平坦) 歩き + (階段) 昇り + (階段) 降り
(any) 静止	92.53%	7.47%
(平坦) 歩き + (階段) 昇り + (階段) 降り	0.62%	99.38%

表 4.22 歩き + 昇り + 降りの適合率 (No.2)

システム \ 実際	(any) 静止	(平坦) 歩き + (階段) 昇り + (階段) 降り
(any) 静止	97.09%	2.91%
(平坦) 歩き + (階段) 昇り + (階段) 降り	5.54%	94.46%

表 4.23 昇り + 降りの再現率 (No.3)

システム \ 実際	(any) 静止	(平坦) 歩き	(階段) 昇り + 降り
(any) 静止	72.91%	26.18%	0.91%
(平坦) 歩き	0.03%	12.68%	87.29%
(階段) 昇り + 降り	0.00%	0.01%	99.99%

表 4.24 昇り + 降りの適合率 (No.3)

システム \ 実際	(any) 静止	(平坦) 歩き	(階段) 昇り + 降り
(any) 静止	99.93%	0.07%	0.00%
(平坦) 歩き	45.20%	54.14%	0.66%
(階段) 昇り + 降り	0.16%	40.18%	59.66%

表 4.25 歩き + 昇り + 降りの再現率 (No.3)

システム \ 実際	(any) 静止	(平坦) 歩き + (階段) 昇り + (階段) 降り
(any) 静止	72.91%	27.09%
(平坦) 歩き + (階段) 昇り + (階段) 降り	0.01%	99.99%

表 4.26 歩き + 昇り + 降りの適合率 (No.3)

システム \ 実際	(any) 静止	(平坦) 歩き + (階段) 昇り + (階段) 降り
(any) 静止	99.93%	0.07%
(平坦) 歩き + (階段) 昇り + (階段) 降り	18.14%	81.86%

表 4.27 昇り + 降りの再現率 (No.4)

システム \ 実際	(any) 静止	(平坦) 歩き	(階段) 昇り + 降り
(any) 静止	73.66%	10.53%	15.81%
(平坦) 歩き	7.54%	11.04%	81.42%
(階段) 昇り + 降り	0.00%	0.00%	100.00%

表 4.28 昇り + 降りの適合率 (No.4)

システム \ 実際	(any) 静止	(平坦) 歩き	(階段) 昇り + 降り
(any) 静止	87.51%	12.49%	0.00%
(平坦) 歩き	32.29%	67.71%	0.00%
(階段) 昇り + 降り	18.37%	15.74%	65.89%

表 4.29 歩き + 昇り + 降りの再現率 (No.4)

システム \ 実際	(any) 静止	(平坦) 歩き + (階段) 昇り + (階段) 降り
(any) 静止	73.66%	26.34%
(平坦) 歩き + (階段) 昇り + (階段) 降り	3.77%	96.23%

表 4.30 歩き + 昇り + 降りの適合率 (No.4)

システム \ 実際	(any) 静止	(平坦) 歩き + (階段) 昇り + (階段) 降り
(any) 静止	87.51%	12.49%
(平坦) 歩き + (階段) 昇り + (階段) 降り	22.64%	77.36%

表 4.31 昇り + 降りの再現率 (No.5)

システム \ 実際	(any) 静止	(平坦) 歩き	(階段) 昇り + 降り
(any) 静止	33.65%	27.57%	38.79%
(平坦) 歩き	2.51%	1.43%	96.06%
(階段) 昇り + 降り	0.00%	0.00%	100.00%

表 4.32 昇り + 降りの適合率 (No.5)

システム \ 実際	(any) 静止	(平坦) 歩き	(階段) 昇り + 降り
(any) 静止	84.07%	15.93%	0.00%
(平坦) 歩き	68.17%	31.83%	0.00%
(階段) 昇り + 降り	17.43%	33.67%	48.90%

表 4.33 歩き + 昇り + 降りの再現率 (No.5)

システム \ 実際	(any) 静止	(平坦) 歩き + (階段) 昇り + (階段) 降り
(any) 静止	33.65%	66.35%
(平坦) 歩き + (階段) 昇り + (階段) 降り	1.26%	98.74%

表 4.34 歩き + 昇り + 降りの適合率 (No.5)

システム \ 実際	(any) 静止	(平坦) 歩き + (階段) 昇り + (階段) 降り
(any) 静止	84.07%	15.93%
(平坦) 歩き + (階段) 昇り + (階段) 降り	39.51%	60.49%

表 4.35 昇り + 降りの再現率 (No.6)

システム \ 実際	(any) 静止	(平坦) 歩き	(階段) 昇り + 降り
(any) 静止	80.25%	19.75%	0.00%
(平坦) 歩き	4.08%	40.17%	55.75%
(階段) 昇り + 降り	0.00%	12.61%	87.39%

表 4.36 昇り + 降りの適合率 (No.6)

システム \ 実際	(any) 静止	(平坦) 歩き	(階段) 昇り + 降り
(any) 静止	84.47%	15.53%	0.00%
(平坦) 歩き	8.16%	59.45%	32.39%
(階段) 昇り + 降り	0.00%	31.29%	68.71%

表 4.37 歩き + 昇り + 降りの再現率 (No.6)

システム \ 実際	(any) 静止	(平坦) 歩き + (階段) 昇り + (階段) 降り
(any) 静止	80.25%	19.75%
(平坦) 歩き + (階段) 昇り + (階段) 降り	2.04%	97.96%

表 4.38 歩き + 昇り + 降りの適合率 (No.6)

システム \ 実際	(any) 静止	(平坦) 歩き + (階段) 昇り + (階段) 降り
(any) 静止	84.47%	15.53%
(平坦) 歩き + (階段) 昇り + (階段) 降り	3.26%	96.74%

表 4.39 昇り + 降りの再現率 (No.7)

システム \ 実際	(any) 静止	(平坦) 歩き	(階段) 昇り + 降り
(any) 静止	88.26%	3.82%	7.92%
(平坦) 歩き	0.24%	8.71%	91.05%
(階段) 昇り + 降り	0.00%	0.00%	100.00%

表 4.40 昇り + 降りの適合率 (No.7)

システム \ 実際	(any) 静止	(平坦) 歩き	(階段) 昇り + 降り
(any) 静止	99.00%	1.00%	0.00%
(平坦) 歩き	10.71%	89.29%	0.00%
(階段) 昇り + 降り	4.62%	54.74%	40.64%

表 4.41 歩き + 昇り + 降りの再現率 (No.7)

システム \ 実際	(any) 静止	(平坦) 歩き + (階段) 昇り + (階段) 降り
(any) 静止	88.26%	11.74%
(平坦) 歩き + (階段) 昇り + (階段) 降り	0.12%	99.88%

表 4.42 歩き + 昇り + 降りの適合率 (No.7)

システム \ 実際	(any) 静止	(平坦) 歩き + (階段) 昇り + (階段) 降り
(any) 静止	99.00%	1.00%
(平坦) 歩き + (階段) 昇り + (階段) 降り	6.13%	93.87%

表 4.43 昇り + 降りの再現率 (No.8)

実際 \ システム	(any) 静止	(平坦) 歩き	(階段) 昇り + 降り
(any) 静止	91.12%	4.28%	4.60%
(平坦) 歩き	0.41%	2.91%	96.68%
(階段) 昇り + 降り	0.00%	0.00%	100.00%

表 4.44 昇り + 降りの適合率 (No.8)

システム \ 実際	(any) 静止	(平坦) 歩き	(階段) 昇り + 降り
(any) 静止	98.86%	1.14%	0.00%
(平坦) 歩き	25.64%	74.36%	0.00%
(階段) 昇り + 降り	16.87%	35.52%	47.60%

表 4.45 歩き + 昇り + 降りの再現率 (No.8)

システム \ 実際	(any) 静止	(平坦) 歩き + (階段) 昇り + (階段) 降り
(any) 静止	91.12%	8.88%
(平坦) 歩き + (階段) 昇り + (階段) 降り	0.21%	99.79%

表 4.46 歩き + 昇り + 降りの適合率 (No.8)

システム \ 実際	(any) 静止	(平坦) 歩き + (階段) 昇り + (階段) 降り
(any) 静止	98.86%	1.14%
(平坦) 歩き + (階段) 昇り + (階段) 降り	17.00%	83.00%

表 4.47 昇り + 降りの再現率 (No.9)

実際 \ システム	(any) 静止	(平坦) 歩き	(階段) 昇り + 降り
(any) 静止	68.08%	24.87%	7.05%
(平坦) 歩き	0.00%	27.03%	72.97%
(階段) 昇り + 降り	0.00%	0.00%	100.00%

表 4.48 昇り + 降りの適合率 (No.9)

システム \ 実際	(any) 静止	(平坦) 歩き	(階段) 昇り + 降り
(any) 静止	100.00%	0.00%	0.00%
(平坦) 歩き	19.60%	79.46%	0.94%
(階段) 昇り + 降り	0.37%	44.21%	55.42%

表 4.49 歩き + 昇り + 降りの再現率 (No.9)

システム \ 実際	(any) 静止	(平坦) 歩き + (階段) 昇り + (階段) 降り
(any) 静止	68.08%	31.92%
(平坦) 歩き + (階段) 昇り + (階段) 降り	0.00%	100.00%

表 4.50 歩き + 昇り + 降りの適合率 (No.9)

システム \ 実際	(any) 静止	(平坦) 歩き + (階段) 昇り + (階段) 降り
(any) 静止	100.00%	0.00%
(平坦) 歩き + (階段) 昇り + (階段) 降り	7.76%	92.24%

まず、ケース1の再現率に関しては、全ての被験者に共通して「平坦×歩き」の精度が低くなっている。また、低くなっている理由は、いずれの被験者も「階段×(昇り+降り)」に誤認識している点が共通しており、階段昇降との区別が難しいことが窺える。「階段×(昇り+降り)」の再現率は非常に高く、100%となっている被験者もいる。適合率に関しては、再現率と比べて「平坦×歩き」の精度が高くなり、「階段×(昇り+降り)」の精度が低くなっており、正反対の結果となっている。このことから、どのユーザにおいても再現率と適合率はトレードオフの関係にあると推察できる。また、「any×静止」に関しては、再現率・適合率共に高い結果が得られた。

続いて、ケース2の再現率に関しては、「平坦×歩き」+「階段×昇り」+「階段×降り」の結果が良かった。ケース1の際は、「平坦×歩き」の精度が低かったが、ケース2のように区別をしない場合では、精度が大きく改善されることがわかる。再現率程ではないが、適合率も区別をしない場合では精度が改善されている。

以上から、ケース2のように「any×静止」とそれ以外のコンテキストに分類するのはできているが、「平坦×歩き」と「階段×(昇り+降り)」を切り分けて自己抑制したいと考えているユーザに対しては、十分な精度を提供できないことがわかった。

4.2 加速度センサ + Wi-Fi を用いた手法に関する実験

4.2.1 実験環境

第4.1.3項の実験と同様に、被験者に自身の加速度を記録する端末を持ち歩いてもらい、著者が実際にコンテキストが切り替わるタイミングを計測した。但し、設置するアクセスポイント数の都合上、歩行ルートは大学校舎の3階から4階への3往復に変更をして、階段昇り、降りの歩行データを3本ずつ得るようにした。計測場所として実際に使用した歩行ルートを図4.15に示す。

昇りの歩行データ取得ルートは、図4.15の3F廊下に示された赤丸から始まり、赤線で引かれたルートを辿って、3.5Fの階段前で一旦静止した後、再び歩いて終点である4Fの赤丸へと向かう。具体的には、3F廊下(静止)→3F廊下(歩き)→3F階段前(歩き)→階段(昇り)→3.5F階段前(歩き)→3.5F階段前(静止)→3.5F階段前(歩き)→階段(昇り)→4F階段前(歩き)→4F廊下(歩き)→4F廊下(静止)となっている。一方、降りの歩行ルートは昇りと逆で、4Fの赤丸が起点となり、3Fの赤丸が終点となる。また、階段はいずれも階段(降り)となる。

なお、歩行者には単に赤線のルートを歩行してもらうだけでなく、床に張ったテープ上を歩行してもらい、決められた座標上(理想的な座標)を移動してもらった。これは、コンテキスト認識を行うにあたり、理想的な座標を利用した場合と、実際に推定された座標を利用した場合とを比較して、最終的な精度にどのような違いが表れるかを観測するためである。

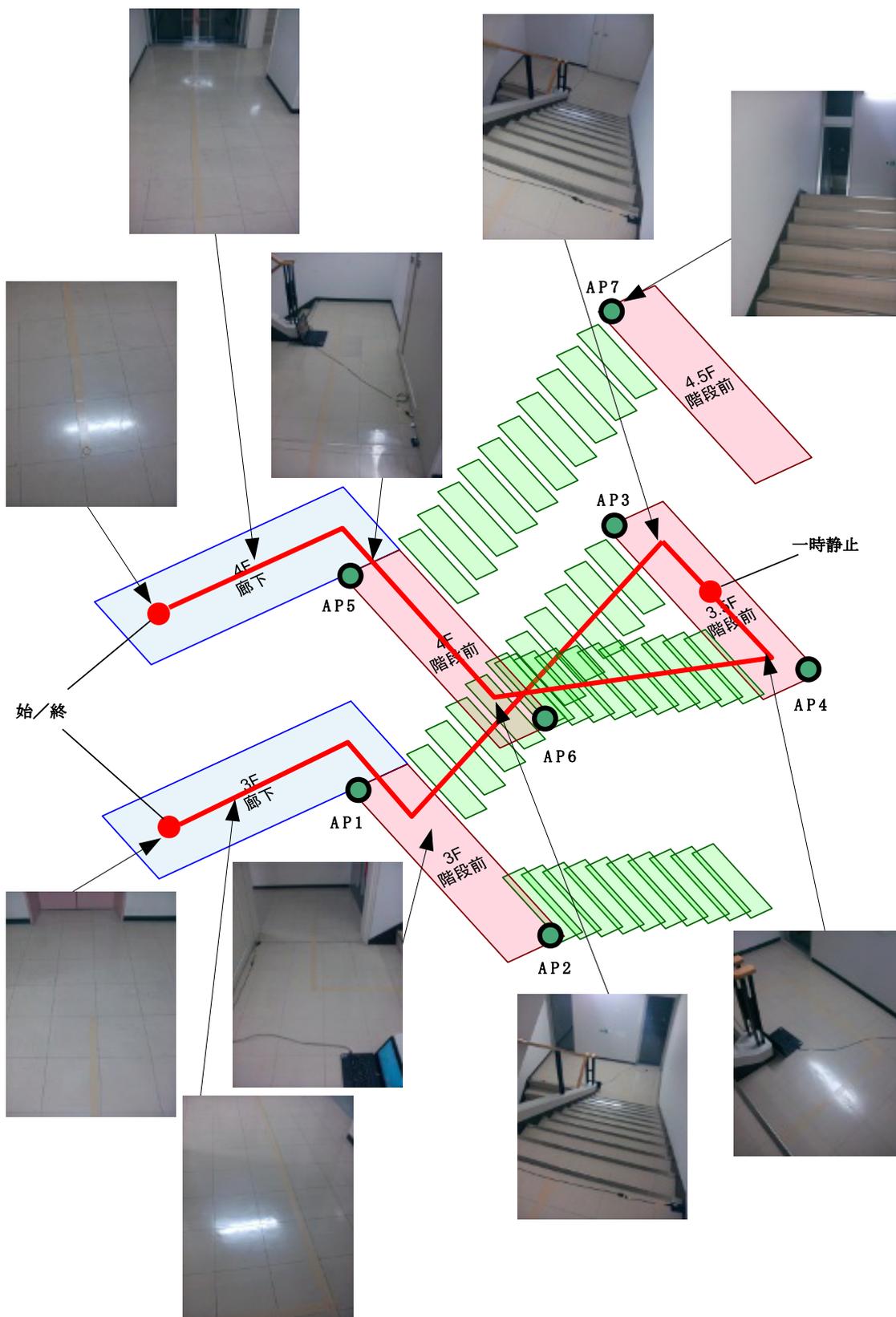


図 4.15 加速度センサ +Wi-Fi を用いた手法における歩行ルート

使用したアクセスポイントに関しては、一般向けに販売されている ELECOM 社の WRH-300 を使用した。アクセスポイントは、図 4.15 の緑点に示された場所にそれぞれ設置し、い

ずれも階段前の床上に置いた。具体的な設置座標を表 4.51 に示す（表中の単位は cm）。このとき、アクセスポイントは電源を供給するためにノートパソコンに接続され、電波を発している。なお、座標推定を行うためには、端末がアクセスポイントからの電波を受信できさえすればよいので、必ずしもインターネットに接続されている必要はない。

表 4.51 アクセスポイント (AP) の設置座標

AP 番号	x 座標	y 座標	z 座標
AP1	0	0	0
AP2	134	300	0
AP3	416	0	187
AP4	550	300	187
AP5	0	0	374
AP6	134	300	374
AP7	416	0	561

取得するデータは、鉛直方向、水平方向の加速度に加え、周囲に設置されたアクセスポイントの受信信号強度 (RSSI) と周波数帯も同時に取得され、テキストファイル (歩行データ) として出力される。このとき、SSID からどのアクセスポイントの電波かを特定し、例えば AP1 の SSID であれば、AP1 の RSSI や周波数帯として決まった列に書き出していく。図 4.16 に実際の歩行データを示す。また、タイミングを記録したテキストファイルは図 4.17 のように出力される。理想的な座標を計測するため、第 4.1 節の図 4.3 に比べて細かなタイミングを計って出力している。タイミングを計った場所をチェックポイント (CP) として青点に示し、昇りルート時を図 4.18 に、降りルート時を図 4.19 にそれぞれ示す。なお、歩行ルートとの関係は、CP1 は歩き始めの始点、CP33 は終点であり、CP17 は一時的に静止する場所となっている。また、各被験者のタイミングを記録した座標を表 4.52 に示し (表中の単位は cm)、今後は CP の座標を理想的な座標として扱う。但し、Z 軸は鉛直方向の上向きを正とする。

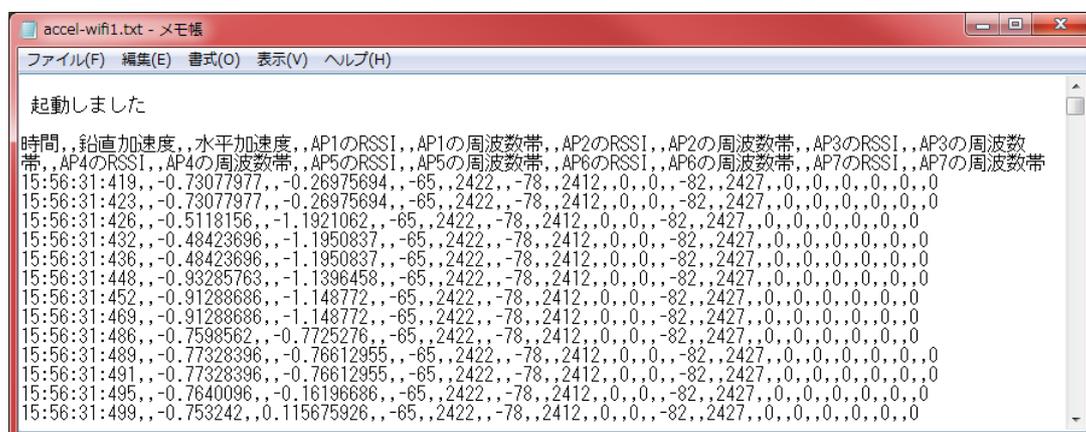


図 4.16 accel-wifi.txt の例

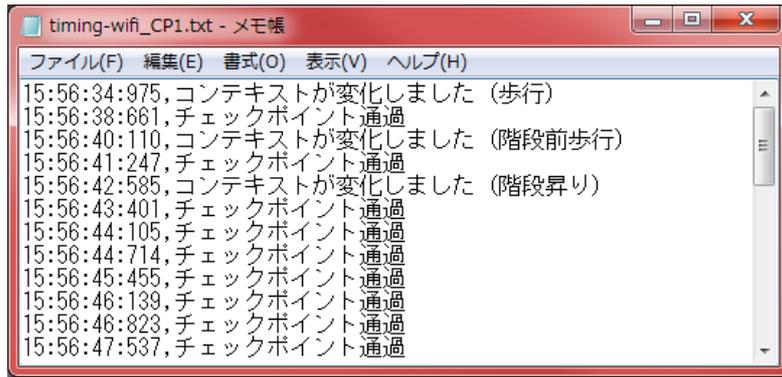


図 4.17 timing-wifi_CP.txt の例

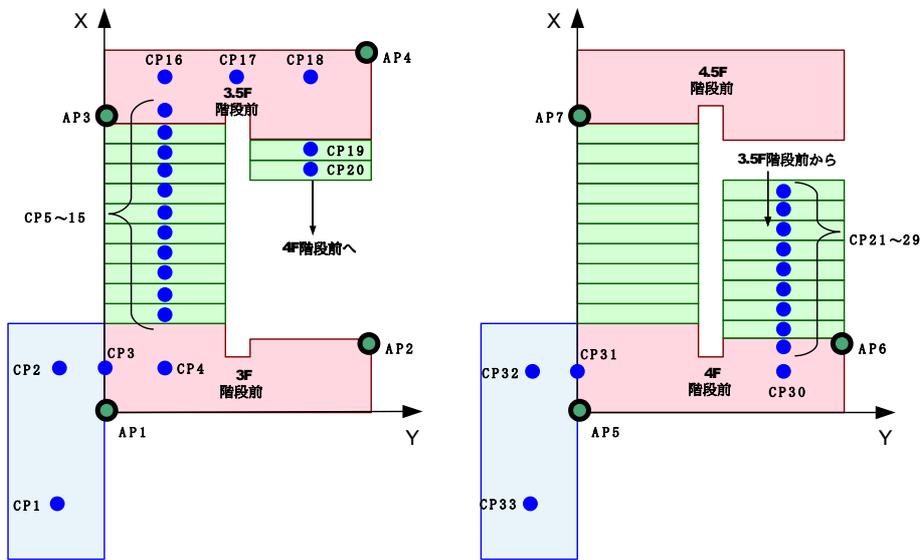


図 4.18 昇りルート時のチェックポイント (CP) 位置

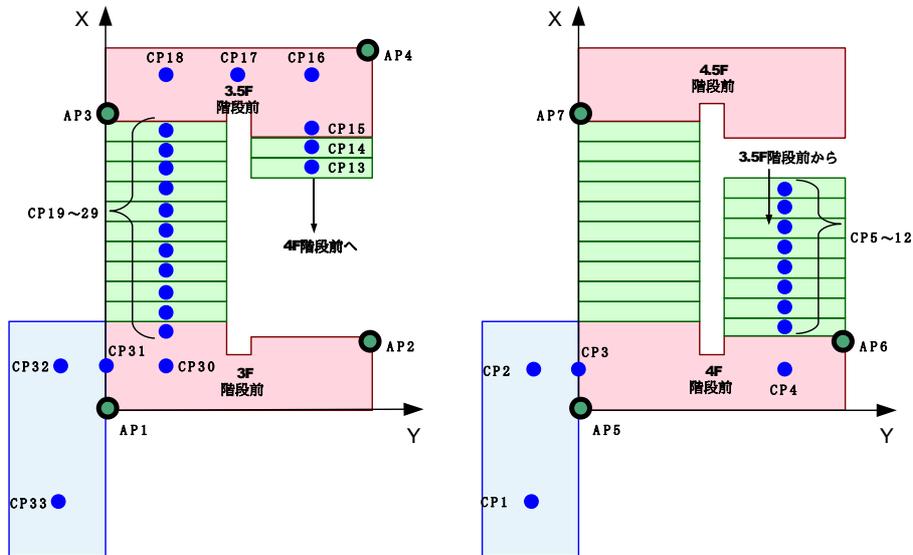
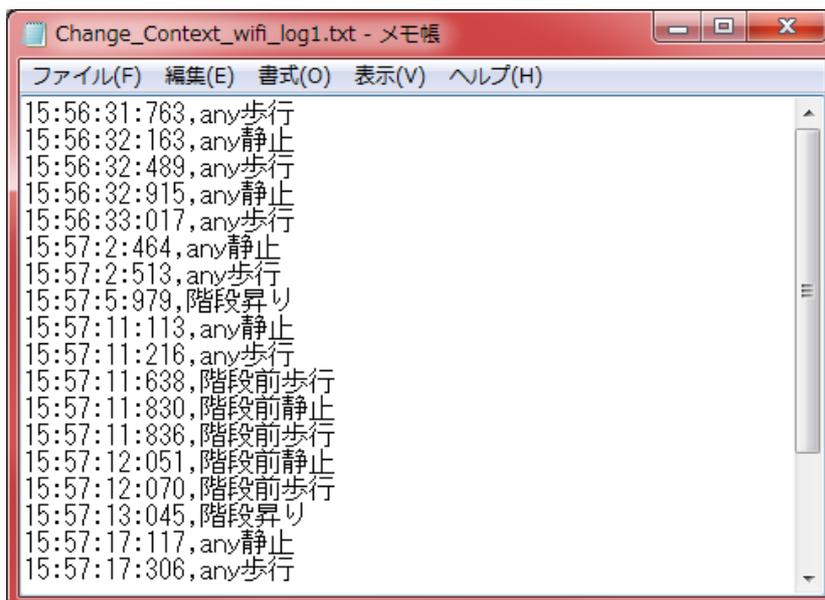


図 4.19 降りルート時のチェックポイント (CP) 位置

表 4.52 チェックポイント (CP) の設定座標

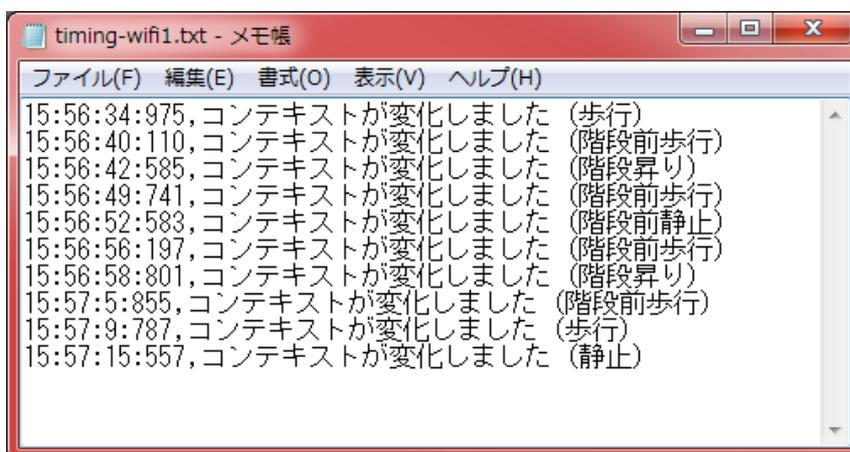
CP 番号	昇りルート			降りルート		
	x 座標	y 座標	z 座標	x 座標	y 座標	z 座標
CP1	-200	-125	0	-200	-125	374
CP2	67	-125	0	67	-125	374
CP3	67	0	0	67	0	374
CP4	67	80	0	67	224	374
CP5	157	80	17	160	224	357
CP6	182	80	34	186	224	340
CP7	208	80	51	212	224	323
CP8	234	80	68	238	224	306
CP9	260	80	85	264	224	289
CP10	286	80	102	290	224	272
CP11	312	80	119	316	224	255
CP12	338	80	136	342	224	238
CP13	364	80	153	368	224	221
CP14	390	80	170	393	224	204
CP15	416	80	187	420	224	187
CP16	506	80	187	506	224	187
CP17	506	152	187	506	152	187
CP18	506	224	187	506	80	187
CP19	393	224	204	390	80	170
CP20	368	224	221	364	80	153
CP21	342	224	238	338	80	136
CP22	316	224	255	312	80	119
CP23	290	224	272	286	80	102
CP24	264	224	289	260	80	85
CP25	238	224	306	234	80	68
CP26	212	224	323	208	80	51
CP27	186	224	340	182	80	34
CP28	160	224	357	157	80	17
CP29	134	224	374	131	80	0
CP30	67	224	374	67	80	0
CP31	67	-1	374	67	-1	0
CP32	67	-125	374	67	-125	0
CP33	-200	-125	374	-200	-125	0

ここからは、「推定された座標を利用した精度の求め方」と、「理想的な座標を利用した精度の求め方」ではそれぞれ異なっているので、話を分けて説明していく。まず、前者の流れは第4.1節における再現率・適合率の求め方と同じであり、図4.16のテキストファイルをPC上に実装した本システムに流すと、RSSIと周波数帯から座標が推定され、推定座標と鉛直加速度に応じてコンテキスト認識が行われる。扱うコンテキストは増えているものの、時間と認識したコンテキストを記した図4.20のようなテキストファイルが出力される。その後、図4.17から「コンテキストが変化～」の部分抜き出した図4.21と、図4.20を統合し、時刻順に整理した図4.22を出力する。最後に、図4.22に示したテキストファイルから、再現率 (Recall) と適合率 (Precision) を出力していく。



```
Change_Context_wifi_log1.txt - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
15:56:31:763,any歩行
15:56:32:163,any静止
15:56:32:489,any歩行
15:56:32:915,any静止
15:56:33:017,any歩行
15:57:2:464,any静止
15:57:2:513,any歩行
15:57:5:979,階段昇り
15:57:11:113,any静止
15:57:11:216,any歩行
15:57:11:638,階段前歩行
15:57:11:830,階段前静止
15:57:11:836,階段前歩行
15:57:12:051,階段前静止
15:57:12:070,階段前歩行
15:57:13:045,階段昇り
15:57:17:117,any静止
15:57:17:306,any歩行
```

図 4.20 Change_Context_wifi_log.txt の例



```
timing-wifi1.txt - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
15:56:34:975,コンテキストが変化しました (歩行)
15:56:40:110,コンテキストが変化しました (階段前歩行)
15:56:42:585,コンテキストが変化しました (階段昇り)
15:56:49:741,コンテキストが変化しました (階段前歩行)
15:56:52:583,コンテキストが変化しました (階段前静止)
15:56:56:197,コンテキストが変化しました (階段前歩行)
15:56:58:801,コンテキストが変化しました (階段昇り)
15:57:5:855,コンテキストが変化しました (階段前歩行)
15:57:9:787,コンテキストが変化しました (歩行)
15:57:15:557,コンテキストが変化しました (静止)
```

図 4.21 timing-wifi.txt の例

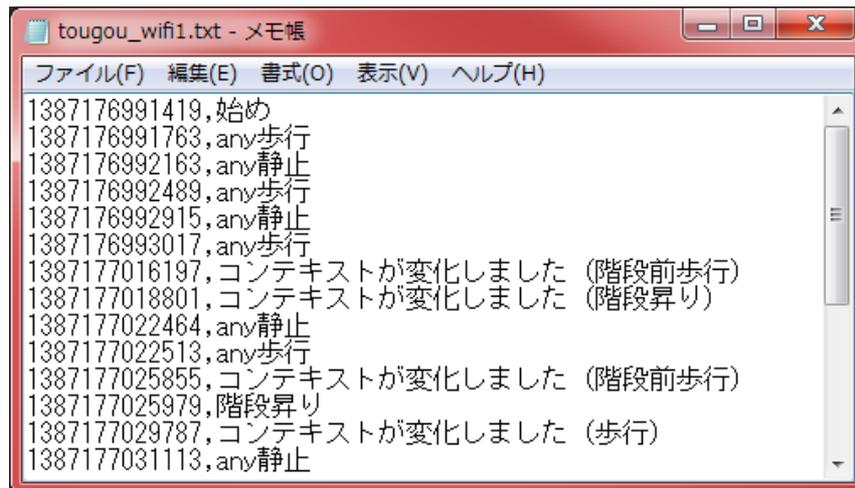


図 4.22 tougou_wifi.txt の例

一方、後者の「理想的な座標を利用した精度の求め方」では、実際の歩行データである図 4.16 と図 4.17 の時刻を照らし合わせて、一部を CP の座標に置き換えた図 4.23 のテキスト（理想的な座標が得られたとする歩行データ）を作成する。その後は、座標の推定を行わずに、理想的な歩行データの座標をそのまま利用するという違いはあるものの、前者の「推定された座標を利用した精度の求め方」と同様の流れであるため割愛する。

最後に本実験では、新たに「階段前×静止」と「階段前×歩き」のコンテキストを認識するが、「階段前」空間の x 座標、 y 座標は図 4.18 と図 4.19 に示す赤い部分であり、高さにあたる z 座標は現在の「階段前」空間からさらに $0.5F$ 高い場所にある「階段前」空間の床面までの高さとする。例えば、 $3F$ における「階段前」空間の高さは、 $3.5F$ の床面までの高さとなる。

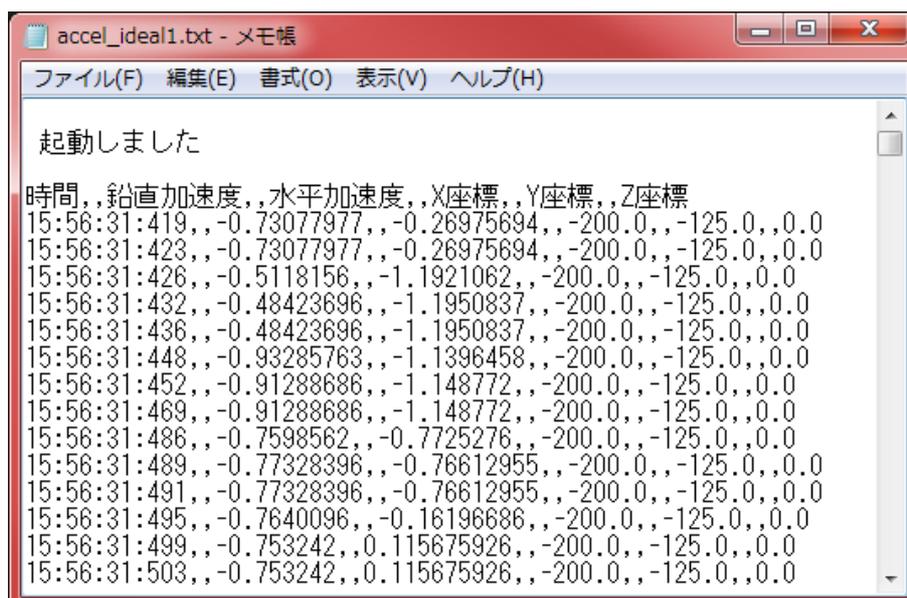


図 4.23 accel_ideal.txt の例

4.2.2 加速度 + Wi-Fi を用いた手法における再現率・適合率に関する実験

第4.1節では、各ユーザの歩行データに応じて、再現率・適合率を求めた。本項においても同様に、新たに認識を行った移動コンテキストである「階段前×静止」、「階段前×歩き」を加え、再現率・適合率の評価を行っていく。

被験者は3人で、第4.1.3項におけるNo.1～3の被験者から改めて歩行データを取得している。各被験者から得られた結果を表4.53～4.58に示す。また、得られた加速度は同じままで、推定座標を理想的な座標に置き換えた歩行データを使用した場合の結果を表4.59～4.64に示す。但し、第4.1.3項の実験結果から、「any×静止」とその他コンテキストの分類はできていたので、加速度センサで分類する際のパラメータ（「加速度センサを用いたパラメータ変動手法におけるWS, max-in, min-inの3つ」）は、表4.11に記載されている各被験者の最適化パラメータを使用した。なお、表4.56の一部が「-」という表記になっているのは、システムが「階段前×静止」と認識することがなかったため、適合率を求める式の分母が0となり、計算できなかったためである。

表 4.53 加速度 + Wi-Fi を用いた手法における No.1 の再現率

システム \ 実際	(any) 静止	(平坦) 歩き	(階段) 昇り	(階段) 降り	(階段前) 静止	(階段前) 歩き
(any) 静止	7.14%	76.32%	5.28%	0.85%	1.40%	9.01%
(平坦) 歩き	0.23%	57.13%	7.63%	20.40%	0.04%	14.56%
(階段) 昇り	0.14%	83.26%	16.60%	0.00%	0.00%	0.00%
(階段) 降り	0.36%	42.54%	7.97%	31.65%	0.00%	17.49%
(階段前) 静止	0.59%	49.41%	16.67%	16.67%	0.00%	16.67%
(階段前) 歩き	0.47%	55.83%	14.85%	18.70%	0.00%	10.14%

表 4.54 加速度 + Wi-Fi を用いた手法における No.1 の適合率

システム \ 実際	(any) 静止	(平坦) 歩き	(階段) 昇り	(階段) 降り	(階段前) 静止	(階段前) 歩き
(any) 静止	71.31%	11.94%	1.86%	2.00%	4.56%	8.33%
(平坦) 歩き	29.75%	20.69%	16.52%	9.71%	5.03%	18.30%
(階段) 昇り	12.04%	36.49%	10.67%	10.21%	4.68%	25.90%
(階段) 降り	0.73%	59.87%	0.00%	18.13%	4.08%	17.19%
(階段前) 静止	95.69%	4.31%	0.00%	0.00%	0.00%	0.00%
(階段前) 歩き	22.56%	40.30%	0.00%	13.84%	9.04%	14.26%

表 4.55 加速度 + Wi-Fi を用いた手法における No.2 の再現率

システム 実際	(any) 静止	(平坦) 歩き	(階段) 昇り	(階段) 降り	(階段前) 静止	(階段前) 歩き
(any) 静止	3.66%	70.44%	9.08%	16.81%	0.00%	0.00%
(平坦) 歩き	0.00%	43.08%	43.22%	11.44%	0.00%	0.26%
(階段) 昇り	0.00%	1.05%	61.11%	33.33%	0.00%	4.51%
(階段) 降り	0.00%	100.00%	0.00%	0.00%	0.00%	0.00%
(階段前) 静止	0.35%	39.65%	40.00%	20.00%	0.00%	00.00%
(階段前) 歩き	0.00%	36.66%	42.70%	15.31%	0.00%	5.34%

表 4.56 加速度 + Wi-Fi を用いた手法における No.2 の適合率

システム 実際	(any) 静止	(平坦) 歩き	(階段) 昇り	(階段) 降り	(階段前) 静止	(階段前) 歩き
(any) 静止	79.80%	0.00%	0.00%	0.00%	20.20%	0.00%
(平坦) 歩き	58.06%	16.83%	0.45%	5.95%	3.97%	14.74%
(階段) 昇り	5.87%	33.09%	16.30%	0.00%	9.73%	35.01%
(階段) 降り	41.26%	36.18%	9.83%	0.00%	5.01%	7.73%
(階段前) 静止	—	—	—	—	—	—
(階段前) 歩き	0.00%	3.34%	32.02%	0.00%	0.00%	64.63%

表 4.57 加速度 + Wi-Fi を用いた手法における No.3 の再現率

システム 実際	(any) 静止	(平坦) 歩き	(階段) 昇り	(階段) 降り	(階段前) 静止	(階段前) 歩き
(any) 静止	41.38%	38.18%	6.40%	0.03%	10.30%	3.71%
(平坦) 歩き	0.75%	64.35%	14.98%	9.91%	0.07%	9.93%
(階段) 昇り	0.13%	83.93%	15.94%	0.00%	0.00%	0.00%
(階段) 降り	0.00%	83.78%	0.00%	16.22%	0.00%	0.00%
(階段前) 静止	28.97%	58.33%	5.53%	7.18%	0.00%	0.00%
(階段前) 歩き	0.82%	77.10%	12.59%	9.49%	0.00%	0.00%

表 4.58 加速度+ Wi-Fi を用いた手法における No.3 の適合率

システム \ 実際	(any) 静止	(平坦) 歩き	(階段) 昇り	(階段) 降り	(階段前) 静止	(階段前) 歩き
(any) 静止	68.69%	2.55%	0.17%	0.00%	24.32%	4.26%
(平坦) 歩き	9.53%	21.18%	18.05%	17.39%	5.93%	27.93%
(階段) 昇り	17.08%	42.45%	13.72%	0.00%	2.35%	24.40%
(階段) 降り	0.18%	70.57%	0.00%	13.77%	2.63%	12.84%
(階段前) 静止	65.93%	34.07%	0.00%	0.00%	0.00%	0.00%
(階段前) 歩き	14.16%	85.84%	0.00%	0.00%	0.00%	0.00%

表 4.59 加速度+ Wi-Fi を用いた手法における No.1 の再現率 (理想)

システム \ 実際	(any) 静止	(平坦) 歩き	(階段) 昇り	(階段) 降り	(階段前) 静止	(階段前) 歩き
(any) 静止	8.54%	91.46%	0.00%	0.00%	0.00%	0.00%
(平坦) 歩き	0.27%	99.51%	0.00%	0.00%	0.00%	0.21%
(階段) 昇り	0.14%	0.83%	98.69%	0.00%	0.00%	0.34%
(階段) 降り	0.36%	0.15%	0.00%	99.29%	0.00%	0.21%
(階段前) 静止	0.00%	0.00%	0.00%	0.00%	0.59%	99.41%
(階段前) 歩き	0.00%	0.03%	0.09%	0.04%	0.47%	99.37%

表 4.60 加速度+ Wi-Fi を用いた手法における No.1 の適合率 (理想)

システム \ 実際	(any) 静止	(平坦) 歩き	(階段) 昇り	(階段) 降り	(階段前) 静止	(階段前) 歩き
(any) 静止	77.48%	17.61%	2.56%	2.35%	0.00%	0.00%
(平坦) 歩き	34.86%	64.76%	0.31%	0.05%	0.00%	0.02%
(階段) 昇り	0.00%	0.00%	99.85%	0.00%	0.00%	0.15%
(階段) 降り	0.00%	0.00%	0.00%	99.94%	0.00%	0.06%
(階段前) 静止	0.00%	0.00%	0.00%	0.00%	25.00%	75.00%
(階段前) 歩き	0.00%	0.16%	0.16%	0.09%	25.41%	74.17%

表 4.61 加速度 + Wi-Fi を用いた手法における No.2 の再現率 (理想)

システム 実際	(any) 静止	(平坦) 歩き	(階段) 昇り	(階段) 降り	(階段前) 静止	(階段前) 歩き
(any) 静止	3.65%	96.35%	0.00%	0.00%	0.00%	0.00%
(平坦) 歩き	0.00%	99.89%	0.00%	0.00%	0.00%	0.11%
(階段) 昇り	0.00%	0.00%	99.68%	0.00%	0.00%	0.32%
(階段) 降り	0.00%	0.00%	0.00%	99.87%	0.00%	0.13%
(階段前) 静止	0.00%	0.00%	0.00%	0.00%	0.29%	99.71%
(階段前) 歩き	0.00%	0.23%	0.37%	0.24%	0.00%	99.17%

表 4.62 加速度 + Wi-Fi を用いた手法における No.2 の適合率 (理想)

システム 実際	(any) 静止	(平坦) 歩き	(階段) 昇り	(階段) 降り	(階段前) 静止	(階段前) 歩き
(any) 静止	100.00%	0.00%	0.00%	0.00%	0.00%	0.00%
(平坦) 歩き	51.03%	48.88%	0.00%	0.00%	0.00%	0.10%
(階段) 昇り	0.00%	0.00%	99.37%	0.00%	0.00%	0.63%
(階段) 降り	0.00%	0.00%	0.00%	99.60%	0.00%	0.40%
(階段前) 静止	0.00%	0.00%	0.00%	0.00%	100.00%	0.00%
(階段前) 歩き	0.00%	0.07%	0.12%	0.05%	37.46%	62.31%

表 4.63 加速度 + Wi-Fi を用いた手法における No.3 の再現率 (理想)

システム 実際	(any) 静止	(平坦) 歩き	(階段) 昇り	(階段) 降り	(階段前) 静止	(階段前) 歩き
(any) 静止	51.68%	48.32%	0.00%	0.00%	0.00%	0.00%
(平坦) 歩き	0.82%	98.94%	0.00%	0.00%	0.00%	0.24%
(階段) 昇り	0.13%	2.67%	96.80%	0.00%	0.00%	0.40%
(階段) 降り	0.00%	0.00%	0.00%	99.88%	0.00%	0.12%
(階段前) 静止	0.00%	0.00%	0.00%	0.00%	28.97%	71.03%
(階段前) 歩き	0.00%	0.20%	0.06%	0.09%	0.82%	98.83%

表 4.64 加速度 + Wi-Fi を用いた手法における No.3 の適合率 (理想)

システム \ 実際	(any) 静止	(平坦) 歩き	(階段) 昇り	(階段) 降り	(階段前) 静止	(階段前) 歩き
(any) 静止	97.53%	2.24%	0.23%	0.00%	0.00%	0.00%
(平坦) 歩き	25.28%	73.32%	1.23%	0.00%	0.00%	0.17%
(階段) 昇り	0.00%	0.00%	99.89%	0.00%	0.00%	0.11%
(階段) 降り	0.00%	0.00%	0.00%	99.85%	0.00%	0.15%
(階段前) 静止	0.00%	0.00%	0.00%	0.00%	90.08%	9.92%
(階段前) 歩き	0.00%	0.17%	0.19%	0.06%	16.48%	83.10%

「any × 静止」に関しては、全体的に再現率の結果が低く、「平坦 × 歩き」や「階段前 × 歩き」と認識している間違いが多かった。仮に、「階段前 × 静止」との誤認識が多いのであれば、推定座標の精度が悪いため、「any (階段前以外)」空間と「階段前」空間の区別ができていないものと考えられる。しかし実際は、加速度の振幅を利用して分類を行っている「静止」アクションと「歩き」アクションの区別ができていないため精度が悪くなっており、これは第 4.1.3 項において、加速度で 2 つのコンテキストを区別できていた事実と反する。

誤認識の直接的な原因は、今回得られた加速度の振幅が前回よりも大きかったためであるが、No.1~3 の全員が「静止」を「歩き」と誤認識する傾向にあるため、偶発的に加速度が増加したとは考えにくい。加速度が増大した原因としては、静電誘導という現象が考えられる。静電誘導とは、一般的に電圧を持った物体は周囲に電界を作るものであり、その作られた電界が周囲の回路に影響を与える現象である。最近の加速度センサは、電子回路により静電容量の変化を観測して、電気信号に変換する仕組みであるため、歩行データ採集時に、周囲の Wi-Fi 情報を取得する際に発生した電界が、加速度センサの電子回路に影響を与えて加速度を増幅させた可能性が考えられる。

しかし、加速度が大きくなるなら、max-in, min-in のパラメータ変動域を増やし、改めてパラメータ最適化を行うことで、「any × 静止」の区別ができるようになる。但し、「any × 静止」は他の移動コンテキストに比べて最も危険性が低いため、再現率が低くても問題はないと考えられる。

「平坦 × 歩き」に関しては、再現率・適合率共に理想的な座標を得られると大きく精度が改善されることがわかる。しかし一方、推定座標の場合は「階段 × 昇り」、「階段 × 降り」、「階段前 × 歩き」の誤認識が増えてしまい精度が大きく下がっている。そのため、座標の推定の精度を高めることで、「平坦 × 歩き」の精度も向上させることができると考えられる。

「階段 × 昇り」、「階段 × 降り」、「階段前 × 歩き」に関しても同様に、理想的な座標が得られた場合は大きく精度が改善されている。そのため、全体的な精度を向上させるには、推定座標の精度を高めることが重要であると考えられる。

加速度センサの他に Wi-Fi を組み合わせることで、「階段前 × 静止」と「階段前 × 歩き」が新たに認識できるようになり、階段における危険性を事前に察知できる可能性が生まれた。し

かし、再現率・適合率は低いため、精度の向上が必要である。また、従来の加速度センサのみを用いたシステムと比べ、「any×静止」、「平坦×歩き」、「階段×昇り」、「階段×降り」の認識精度が低下しているため、「階段前」空間の認識を必要としないユーザにとっては、従来のシステムを利用した方が有用であると考えられる。

但し、正確な推定座標が得られれば、従来のシステムよりも「階段×昇り」、「階段×降り」の精度が向上するだけでなく、新たに「階段前×静止」、「階段前×歩き」の認識が可能になり、かつ認識精度を高くすることができるので、より有用なモバイル自己抑制アプリケーションを提供できる可能性がある。

4.2.3 推定座標の精度に関する実験

本項では、推定された座標の精度を調べるため、実際の座標と推定された座標の誤差を調べていく。座標の計測は、図 4.18 に示した昇りルート時の CP1~33 で行った。計測方法は、著者が端末を持って各 CP 上で立ち止まり、それぞれの場所で 5 回ずつアクセスポイントの情報を収集した。収集したデータは、各アクセスポイントからの RSSI と周波数帯であり、5 回分の平均値を利用して距離の推定を行った。また、端末を持っている高さは地上 110cm であるため、CP の z 座標に +110 した座標を測定場所とした。

その後、推定距離から推定座標を求める実験を行った。しかし、推定距離の誤差が大きかったためか、3 球面の交点が見つからず、推定座標を算出することはできなかった。そのため、推定座標の誤差ではなく、各 CP と AP 間における実際の距離と推定された距離の誤差を表 4.65 と 4.66 にまとめた。但し、表中の単位は RSSI が dBm、他が cm とする。

なお、1 つのセル内には 4 つの数値が示されているが、それぞれ左上が RSSI、右上が CP と AP 間の推定距離、左下が実際の距離、右下が誤差（推定距離 - 実際の距離）となっている。CP1 と AP1 間を例にすると、-66 は CP1 地点における AP1 からの RSSI、111 は RSSI を式 3.3 に代入して得られた CP1 と AP1 間の推定距離、260 は CP1 と AP1 間の実距離、-150 は実距離に 150cm 足りていないことを表している。一方、正の場合は、実際の距離よりも長い推定距離であることを意味している。

また、表中の「-」は、電波が届かず RSSI が取得できなかったこと、及び推定距離が計算できなかったことを示している。なお、5 回分の平均値を利用しているが、5 回中 3 回以上 RSSI が取得された場合を電波の受信が可能であったものとして扱い、2 回以下のときは受信不可として扱い、「-」と示している。

表 4.65 アクセスポイント (AP) とチェックポイント (CP) 間における推定距離の誤差 (CP1~20)

CP 番号	AP1		AP2		AP3		AP4		AP5		AP6		AP7	
CP1	-66	111	-	-	-	-	-	-	-	-	-	-	-	-
	260	-150	552	-	633	-	865	-	354	-	602	-	774	-
CP2	-42	7	-72	222	-	-	-78	439	-74	279	-	-	-	-
	179	-173	444	-223	379	-	648	-209	300	-21	505	-	584	-
CP3	-48	14	-63	79	-69	157	-70	175	-67	125	-	-	-	-
	129	-115	326	-248	357	-201	574	-399	272	-148	405	-	570	-
CP4	-57	39	-58	44	-67	124	-74	277	-71	197	-71	197	-	-
	152	-112	255	-211	366	-242	536	-259	284	-86	350	-153	576	-
CP5	-57	39	-60	56	-63	78	-62	70	-78	442	-75	313	-74	277
	217	-178	255	-199	278	-199	454	-385	303	139	332	-19	512	-235
CP6	-62	70	-60	56	-64	88	-63	78	-68	140	-77	394	-73	247
	245	-176	267	-212	251	-163	431	-353	304	-164	322	72	485	-238
CP7	-62	70	-60	56	-64	88	-63	78	-73	249	-	-	-75	310
	275	-205	282	-227	224	-136	407	-329	308	-60	315	-	458	-147
CP8	-67	124	-64	88	-64	88	-64	88	-76	351	-73	249	-75	310
	305	-181	300	-212	199	-111	385	-297	316	36	311	-63	432	-121
CP9	-70	175	-66	111	-62	70	-64	88	-71	197	-78	442	-77	391
	335	-159	320	-209	175	-106	364	-276	326	-128	310	132	406	-15
CP10	-71	197	-63	79	-54	28	-58	44	-75	313	-69	157	-73	247
	365	-168	341	-263	155	-127	345	-301	338	-25	313	-156	381	-134
CP11	-73	248	-68	140	-53	25	-58	44	-76	351	-72	222	-74	277
	395	-148	364	-224	138	-113	327	-283	353	-2	318	-96	357	-80
CP12	-74	278	-68	140	-50	18	-57	39	-76	351	-69	157	-76	348
	426	-148	388	-248	126	-109	311	-272	370	-19	326	-169	334	14
CP13	-77	392	-70	176	-48	14	-55	31	-76	351	-65	99	-75	310
	456	-64	413	-237	122	-108	298	-267	389	-38	337	-238	313	-2
CP14	-71	197	-77	394	-54	28	-56	35	-80	557	-66	111	-73	247
	487	-290	439	-45	125	-98	287	-253	409	148	350	-239	293	-47
CP15	-75	312	-73	249	-60	56	-56	35	-	-	-67	125	-74	277
	517	-206	465	-216	136	-80	280	-245	431	-	366	-241	276	1
CP16	-65	99	-71	197	-53	25	-59	49	-76	351	-65	99	-67	124
	592	-494	524	-327	163	-138	250	-201	518	-167	439	-340	290	-167
CP17	-74	278	-74	279	-55	31	-62	70	-78	442	-61	62	-75	310
	606	-328	498	-220	208	-177	190	-120	534	-92	408	-345	318	-7
CP18	-71	197	-73	249	-57	39	-55	31	-74	279	-67	125	-72	220
	628	-431	482	-233	265	-226	141	-110	559	-280	387	-263	358	-138
CP19	-71	197	-75	313	-61	62	-53	25	-78	442	-65	99	-74	277
	551	-354	414	-101	259	-196	216	-191	456	-14	277	-178	334	-58
CP20	-	-	-76	351	-60	56	-56	35	-70	176	-63	79	-69	156
	543	-	412	-61	271	-215	244	-209	433	-257	250	-171	325	-169

表 4.66 アクセスポイント (AP) とチェックポイント (CP) 間における推定距離の誤差 (CP21~33)

CP 番号	AP1		AP2		AP3		AP4		AP5		AP6		AP7	
CP21	-76	350	-76	351	-61	62	-59	49	-70	176	-58	44	-61	62
	537	-187	412	-61	286	-223	274	-224	410	-234	223	-179	318	-256
CP22	-72	221	-81	625	-61	62	-61	62	-74	279	-53	25	-72	220
	532	-312	415	210	303	-241	304	-242	387	-108	197	-173	314	-94
CP23	-71	197	-73	249	-67	124	-68	139	-72	222	-50	18	-67	124
	529	-333	420	-171	323	-198	334	-195	367	-145	174	-156	313	-190
CP24	-71	197	-72	222	-66	111	-65	98	-67	125	-56	35	-63	78
	528	-332	426	-205	344	-233	364	-266	347	-223	153	-118	315	-238
CP25	-77	392	-79	496	-70	176	-69	156	-63	79	-49	16	-64	87
	529	-137	435	61	366	-191	394	-239	330	-251	135	-120	321	-233
CP26	-	-	-83	786	-68	140	-72	220	-63	79	-51	20	-64	87
	532	-	446	340	390	-251	425	-205	314	-235	124	-104	329	-241
CP27	-78	440	-80	557	-66	111	-70	175	-62	70	-48	14	-54	28
	536	-96	459	97	415	-304	455	-281	301	-231	119	-105	340	-312
CP28	-75	312	-81	625	-75	312	-66	110	-60	56	-56	35	-58	44
	542	-230	474	151	441	-128	486	-376	291	-235	123	-88	353	-309
CP29	-70	175	-78	442	-73	248	-69	156	-60	56	-57	39	-58	44
	550	-375	490	-48	467	-219	517	-361	283	-228	134	-94	368	-324
CP30	-68	139	-80	557	-66	111	-64	88	-57	39	-59	50	-63	78
	538	-398	494	62	510	-399	572	-484	258	-219	150	-100	422	-344
CP31	-	-	-78	442	-75	312	-71	196	-52	22	-56	35	-67	124
	489	-	574	-132	458	-146	642	-446	129	-107	327	-292	357	-234
CP32	-	-	-75	313	-78	441	-76	349	-55	31	-61	62	-70	175
	504	-	648	-335	475	-34	709	-360	179	-148	444	-382	379	-204
CP33	-	-	-	-	-	-	-	-	-59	50	-66	111	-	-
	538	-	726	-	695	-	912	-	260	-211	552	-441	633	-

全体的に、実際の距離に対して距離を短く推定している傾向にあり、表中で一番 RSSI が大きい -42dBm の場合、 7cm という極端に短い推定距離を算出している。また、次に大きい -48dBm では 14cm と計算され、 6dBm 小さくなくても 7cm しか変わらず、同様に極端に短い推定距離になっている。一方、RSSI が小さい場合は、 -80dBm なら 557cm 、 -81dBm なら 625cm 、 -83dBm なら 786cm という推定を行っており、RSSI が大きいときと比べ、 1dBm 減少するだけで推定距離が極端に長くなっている。また、推定距離の誤差が $100\sim 300\text{cm}$ と大きいことから、導出される推定距離と、実際の距離には大きな隔たりがあることがわかる。

原因として考えられるのは、電波干渉の影響である。実験場所のように入り組んだ空間における電波伝搬モデルと、自由空間における電波伝搬モデルとでは、電波の伝わり方も異なるため、観測される RSSI も異なってくる。しかし、推定距離の導出に用いた式 3.3 は、自由空間を想定した式であるため、誤差が大きくなっている可能性がある。解決策としては、電波干渉が少ない周波数を選ぶ方法がある。今回実験に使用したアクセスポイントの周波数は約 2.4GHz であるが、電波干渉に強いとされる 5GHz の周波数を用いたアクセスポイントを使用することで、自由空間に近い環境にすることができると考えられる。

他の解決策としては、式 3.3 のような自由空間を想定した式の代わりに、階段空間向けの新たな式を立てることが考えられる。現在のアルゴリズムは、推定距離を基に推定座標を計算し、最終的なコンテキスト認識を行っているため、最初のプロセスである推定距離の精度が低いと、全体的なコンテキスト認識の精度も低くなってしまふ。そこで今後は、推定距離の精度を向上させるため、表 4.65, 4.66 を基に、実際の距離と推定距離が近くなるような関係式が導けないか考察を行い、導くことができたなら、改めて距離の誤差を検証していく。

他に表から読み取れることとしては、ユーザがアクセスポイントに近づくにつれて RSSI が増加し、遠くなるにつれて減少していることから、実際の距離と RSSI は比例関係にあることがわかる。大半は -70 , -60 台を推移しているが、近づくとも -50 台後半～ -40 台を推移している特徴があり、この特徴を利用することで、座標によらないコンテキスト認識を行える可能性がある。現在のアルゴリズムは、推定距離から求めた推定座標が、「階段前」空間の座標内であるか否かを判定して、コンテキスト認識を行っている。しかし、例えば「階段前」空間に設置してある AP1 と AP2 からの RSSI が -50 台後半以上を示していれば、現在は「階段前」空間であるという認識を行うアルゴリズムも考えられるだろう。

同様に、CP1 と CP33 ではほとんどの AP から電波が届かないという特徴があるので、CP1, CP33 を設置している周辺の空間（「廊下」空間）と「階段及び階段前」空間の分類に役立つ可能性がある。こうしたアクセスポイントの設置場所と RSSI の特性を利用したアルゴリズムは、モバイル自己抑制アプリケーションの汎用性を高めることが期待できる。例えば、本研究では「階段前」空間を特定の座標として定義しているが、違う定義の方が良いと考える人もいるかもしれない。そこで、具体的な座標を設けず、RSSI が大きくなるにつれて「階段前」空間が近づいているというファジィなコンテキスト認識にすることで、「階段前」空間を狭くしたいなら RSSI が -50dBm 以上、広くしたいなら -55dBm 以上のときに警告を発するなどの細かい設定ができるようになり、ユーザの考えに応じた定義へと変更可能になる。

その他には、CP5～15 へと進んでいくにつれ、3F に設置してある AP1 の RSSI が減少し、3.5F に設置してある AP3 の RSSI が増加していることが窺える。また、CP19～29 へと進むにつれて、3.5F に設置してある AP4 の RSSI が減少し、4F に設置してある AP6 の RSSI が増加している。この特徴は、「階段×昇り」、「階段×降り」のコンテキスト認識に利用できる可能性がある。現在のアルゴリズムは、一つ前に推定した z 座標と現在推定した z 座標を比較して、17cm 上昇しているときに「階段×昇り」と認識しているため、許容誤差が 17cm 以下の精度でなければ、精確なコンテキスト認識を行うことができない。しかし、全体的に推定距離の誤差が 100～300cm もあることを考慮すると、推定座標の精度も著しく下がると考えられるため、許容誤差 17cm の精度を得るのは難しいだろう。また、「階段×昇り」「階段×降り」の認識には推定座標が必要となるが、今回の実験のように推定座標が計算できなかった場合は認識ができなくなってしまう。そこで、推定座標の許容誤差を考慮する必要がないどころか、そもそも座標を計算する必要さえない、RSSI の増減に着目したアルゴリズムにすることで、「階段×昇り」、「階段×降り」の認識精度を改善できる可能性がある。今後は、より詳細な RSSI の増減を分析し、分析結果に基づいて新たなシステムを開発して、改めて再現率・適合率に改善が見られるかどうかを検証していく。

第5章

結論

近年の Web コンテンツは、モバイル環境における利用が広がることで、周囲に人間がいる状況での利用など、人間が暮らす生活空間との関係が密になってきている。そのため、「歩きスマホ問題」のような新たな問題が発生しており、従来の固定端末では見られない新しい問題に対して、モバイル環境向けの新しい自己抑制が必要であると考えた。特に階段における歩きスマホは、転落して重傷を負った事例も確認されており、危険性が高い。そこで、階段における歩きスマホ事故を未然に防ぐため、端末が階段におけるユーザの状況（階段コンテキスト）を認識するコンテキスト・アウェアネスの技術を用い、ユーザが歩きスマホをしながら階段を移動したことを検知し、画面にアラートを発することで自粛を促すモバイルアプリケーションを提案した。

従来研究でも階段コンテキスト認識は行われており、加速度センサと気圧センサを組み合わせることで高い精度が得られることがわかっている。しかし、現在普及しているモバイル端末は、気圧センサの搭載率が低いため、大半のユーザは高い精度を誇る階段コンテキスト認識の恩恵を受けることができない。一方、広く普及している加速度センサのみを利用した研究では、機械学習を行うことで、どのユーザに対しても一定の精度を維持している。だが、気圧センサも組み合わせている研究と比べると、精度が下がってしまっており、加速度センサのみで高い精度を実現するのは難しい。ここで、自己抑制の観点から鑑みると、人により危険を感じる状況は異なるため、モバイル自己抑制で重視したいコンテキストも異なると考えられる。そこで本研究では、搭載率が高い加速度センサのみを用いている点では同様であるが、新たに階段コンテキスト認識を行うためのパラメータを設定し、変動をさせることで、ユーザが重視したいコンテキストに調整可能な「加速度センサを用いたパラメータ変動手法」を提案した。また、階段へ入る前の空間（「階段前」空間）を認識して、ユーザに通知できれば、階段における歩きスマホの危険防止に役立つと考え、屋内測位システムに利用されている技術を取り入れたシステムである「加速度センサ + Wi-Fi を用いた手法」を提案した。

「加速度センサを用いたパラメータ変動手法」では、階段における歩きスマホを検知するため、ユーザの行動から得られた加速度、及び、4つの閾値とウィンドウサイズ (WS) をパラメータとして、「any (どこでも) × 静止」、「平坦 (廊下 + 階段前) × 歩き」、「階段 × 昇り」、「階段 × 降り」の4つのコンテキスト認識を行うことができる。一方、「加速度センサ + Wi-Fi

を用いた手法」では、加速度、及び、2つの閾値とWSの他に、アクセスポイントから得られる受信信号強度 (RSSI) を使って、「any (階段前以外) × 静止」、「平坦 (廊下) × 歩き」、「階段 × 昇り」、「階段 × 降り」、「階段前 × 静止」、「階段前 × 歩き」の6種類のコンテキスト認識を行うことができる。これら提案したシステムを検証するため、各手法に従って、実際にシステムを作成した。

「加速度センサを用いたパラメータ変動手法」に従って作成したシステムでは、まず著者の歩行データを使用して検証を行った。評価尺度は再現率・適合率・F値の3つを用い、F値が最大となるパラメータである最適化パラメータを求めた。また、パラメータ最適化後の各コンテキストにおける再現率・適合率を明らかにし、再現率・適合率に与える各パラメータの依存性を調べた。その結果、閾値とWSのパラメータ変動では、全コンテキストの再現率・適合率を同時に高めることは難しいが、逐次パラメータを変動させることで、ユーザが自己抑制したいコンテキストに合わせて調整できる可能性があることを確認した。その他にも、実用的なケースとして、昇りと降りを区別しない場合の認識精度 (ケース1) と、歩き、昇り、降りの区別をしない場合の認識精度 (ケース2) を求めた。結果は、ケース1では歩き以外の認識精度が高く、ケース2では全ての認識精度が高くなった。

次に、9人の被験者に対しても同様に、自己抑制したいコンテキストに合わせて調整できる可能性があるのかを検証するため、各コンテキストのF値が最大となるように最適化を行った。結果は、平均で「any × 静止」が87.22%、「平坦 × 歩き」が53.20%、「階段 × 昇り」が64.92%、「階段 × 降り」が73.85%となり、一定の精度が得られたものの、従来研究の精度には及ばない結果となった。しかし、必要な場所で必要な警告を出せるアプリケーションを提供するためには、再現率の高さが重要になることから、再現率に重点をおいてパラメータを変動させたところ、平均で「any × 静止」が87.45%、「平坦 × 歩き」が53.65%、「階段 × 昇り」が100.00%、「階段 × 降り」が98.50%という結果になり、歩きの再現率に課題は残るものの、危険性が高い昇り、降りではほぼ完璧な精度を得ることができ、モバイル自己抑制アプリケーションとして提供できるだけの精度が得られた。また、実用的なケースに関しては、著者の結果と同じように、ケース1では歩き以外の認識精度が高く、ケース2では全ての認識精度が高くなった。

「加速度センサ + Wi-Fi を用いた手法」に従って作成したシステムでは、3人の被験者に対して実験を行い、再現率・適合率の評価尺度に基づいて検証を行った。実験では、実際に推定された座標を使用した歩行データと、理想的な座標が得られた場合の歩行データを比較したところ、推定座標の精度が低いため、全体的なコンテキスト認識の精度も下がってしまうことがわかった。実際の推定座標を使用した歩行データでは、「加速度センサを用いたパラメータ変動手法」におけるコンテキスト認識の精度よりも悪くなってしまった。一方、理想的な座標が得られた場合の歩行データでは、「階段 × 昇り」、「階段 × 降り」に加え、新たに認識を行った移動コンテキストである「階段前 × 歩き」の再現率がほぼ100%になることがわかった。また、「加速度センサを用いたパラメータ変動手法」において、課題であった「平坦 × 歩き」の再現率もほぼ100%となった。さらに、再現率と同様に適合率も非常に高い結果となり、座標の精度次第ではあるが、潜在的な可能性を持っている手法であることがわかった。同時に、今後の

課題としては、推定座標の精度を向上させる必要があることが確認できた。

そこで、推定座標の精度を検証するための実験を行った。実験では、実際の座標と推定座標を比較することで、どのくらいの誤差があるのかを調べようとしたが、推定距離の誤差が大きいたことが原因で、推定座標を求めることができなかった。推定距離の計算は、推定座標を計算する最初のプロセスで行われるため、まずは推定距離の改善を目指し、実距離と推定距離の誤差を分析することにした。分析の結果、推定距離の計算に使う新たな式を定義することで、改善の可能性があることがわかった。今後の課題としては、実験結果を基に、実際の距離と推定距離が近くなるような関係式が導けないのかを考察し、新たな関係式が導けた場合は、改めて距離の誤差を検証していく。また、ユーザがアクセスポイントに近づくと RSSI が増加し、遠ざかると減少するという特徴を利用することで、距離や座標の推定を行わずとも、「階段×昇り」、「階段×降り」、「階段前×歩き」の各移動コンテキストの認識ができる可能性があることがわかった。そこで今後は、引き続き RSSI の増減を分析し、分析結果に基づいて新たなシステムを開発して、改めて再現率・適合率に改善が見られるかどうかを検証していく。

最後に、本研究における成果のまとめとして、世界各国で問題視されている歩きスマホに対し、改めて警鐘を鳴らすと共に、特に階段における歩きスマホの危険性を指摘した。そこで、階段における歩きスマホの自己抑制ができるように、モバイル自己抑制アプリケーションを提案した。また、危険を感じる状況は人によって異なることから、様々なユーザに対する汎用性を考慮して、「加速度センサを用いたパラメータ変動手法」システムの開発を行った。その結果、加速度センサだけでも、パラメータの変動によって、ユーザが重視したいコンテキストに合わせて認識精度を高めることができるようになった。さらに、加速度センサだけではなく、Wi-Fi を組み合わせてコンテキスト認識を行う「加速度センサ + Wi-Fi を用いた手法」システムを開発したことで、新たに「階段前」空間の認識ができるようになり、階段における歩きスマホの安全性を高めることができた。

謝辞

本研究に際して、様々なご指導を頂きました服部峻助教に厚く御礼申し上げます。また、日常の議論を通じて多くの知識や示唆を頂いた服部研究室の皆様に深く感謝の意を表します。そして、実験の際に被験者を快く引き受けて下さった協力者の方々に深く感謝いたします。

参考文献

- [1] 吉田 京平, 服部 峻: “自己抑制フィルタリングブラウザの試作,” 情報処理学会第 75 回全国大会 (IPSJ 2013), 1ZA-6, pp.3–607–608 (2013).
- [2] Kyohei Yoshida, and Shun Hattori: “Web Self-Controller: A Filtering Browser for Self-Control,” 4th International Conference on E-Service and Knowledge Management (ESKM 2013), pp.26–31 (2013).
- [3] 東京消防庁: “歩きスマホ等に係る事故に注意!! - 東京消防庁 - 東京都,” p.73 (2012).
- [4] 国土技術政策総合研究所: “建築空間におけるユーザー生活行動の安全確保のための評価・対策技術に関する研究 (その 1),” <http://www.tfd.metro.tokyo.jp/lfe/topics/201403/mobile.html> (2014).
- [5] 吉田 京平, 服部 峻: “自己抑制のための階段コンテキスト認識,” 2014 年電子情報通信学会総合大会 ISS 特別企画「学生ポスターセッション」, ISS-SP-344, p.173 (2014).
- [6] 吉田 京平, 服部 峻: “モバイル自己抑制のための階段コンテキスト認識システムのパラメータ最適化,” 電子情報通信学会 モバイルネットワークとアプリケーション研究会 (SIG-MoNA), 信学技報, Vol.114, No.31, MoNA2014-9, pp.127–132 (2014).
- [7] Mantyjarvi Jani, Johan Himberg, and Tapio Seppanen: “Recognizing human motion with multiple acceleration sensors,” 2011 IEEE Int’l Conf. on Systems, Man, and Cybernetics, vol.2, pp.747–752 (2011).
- [8] Andreas Krause, Daniel Siewiorek, Asim Smailagic, and Jonny Farrington: “Unsupervised, Dynamic Identification of Physiological and Activity Context in Wearable Computing,” The 7th International Symposium on Wearable Computers, pp.88–97 (2003).
- [9] Andreas Krause, Matthias Ihmig, Edward Rankin, Derek Leong, Smriti Gupta, Daniel Siewiorek, Asim Smailagic, Michael Deisher, and Uttam Sengupta: “Trading off prediction accuracy and power consumption for context-aware wearable computing,” Proceedings of the 2005 Ninth IEEE International Symposium on Wearable Computers (ISWC 2005), pp.20–26 (2005).
- [10] Apple: “Apple - Apple Watch,” <https://www.apple.com/watch/> (2014).
- [11] Google: “Google Glass,” <https://www.google.com/glass/start/> (2014).
- [12] 男澤 晃陽, 服部 峻: “モバイル端末のセンサー情報を用いた飲酒状態認識,” 電子情報

- 通信学会 情報ネットワーク研究会 (SIG-IN), 信学技報, Vol.114, No.401, IN2014-100, pp.13-18 (2015).
- [13] 大内 一成, 土井 美和子: “スマートフォンを用いた生活行動認識技術,” 東芝レビュー, Vol.68, No.6, pp.40-43 (2013).
- [14] Nishkam Ravi, Nikhil Dandekar, Preetham Mysore, and Michael L. Littman: “Activity Recognition from Accelerometer Data,” AAAI 2005, pp.1541-1546 (2005).
- [15] Toshiaki Iso, and Kenichi Yamazaki: “Gait Analyzer Based on a Cell Phone with a Single Three-axis Accelerometer,” Proceedings of the 8th conference on Human-computer interaction with mobile devices and services, pp.141-144 (2006).
- [16] 小川 晶子, 小西 勇介, 柴崎 亮介: “自立型ポジショニングシステム構築に向けて～着用型センサーを用いた人間の行動モード推定～,” 全国測量技術大会 2002 学生フォーラム発表論文集 (2002).
- [17] 森岡 英之, 新津 善弘: “複数センサを用いたコンテキスト推定法の検討,” 情報処理学会第 72 回全国大会 (IPSJ 2010), 1ZD-2, pp.3-475-476 (2010).
- [18] 浅沼 伸洋, 関田 格, 新津 善弘: “生体センサを用いたサービス推薦のための状態判定方式,” 平成 20 年度 電子情報通信学会 東京支部学生会 研究発表会 (2008).
- [19] Hui Liu, Houshang Darabi, Pat Banerjee, and Jing Liu: “Survey of Wireless Indoor Positioning Techniques and Systems,” Systems, Man, and Cybernetics, Part C: Applications and Reviews, vol.37, no.6, pp.1067-1080 (2007).
- [20] 坂本 龍哉, 大村 廉, 納谷 太, 野間 春生, 鳥山 朋二, 小暮 潔, 佐野 睦夫: “複数人物移動軌跡を観測するセンサネットワークにおける時刻同期精度の評価,” 情報処理学会研究報告 UBI15, pp.35-40 (2007).
- [21] Juan David Hincapie-Ramos, and Pourang Irani: “CrashAlert: Enhancing Peripheral Alertness for Eyes-Busy Mobile Interaction while Walking,” Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, pp.3385-3388 (2013).
- [22] 根岸 匠, 田中 二郎, 神場 知成: ““ながらスマートフォン” 抑止システムの開発,” 情報処理学会第 76 回全国大会 (IPSJ 2014), 4W-5, pp.3-301-302 (2014).
- [23] 北須賀 輝明, 中西 恒夫, 福田 晃: “無線 LAN を用いた屋内向けユーザ位置測定方式 WiPS の実装,” 情報処理学会 DICOMO, pp.349-352 (2004).