



Title	ユーザ体験を小規模で短期間に検証し改善するマイクロUX法に関する研究
Author(s)	鈴木, 健司
Degree Grantor	北海道大学
Degree Name	博士(情報科学)
Dissertation Number	甲第15085号
Issue Date	2022-03-24
DOI	https://doi.org/10.14943/doctoral.k15085
Doc URL	https://hdl.handle.net/2115/85531
Type	doctoral thesis
File Information	Kenji_Suzuki.pdf



博士論文

ユーザ体験を小規模で短期間に検証し改善する マイクロUX法に関する研究

北海道大学 大学院情報科学研究科
情報理工学専攻 複合情報工学講座
ヒューマンコンピュータインタラクション研究室

鈴木 健司

2022年1月24日

Doctoral Dissertation

**A Study on micro-UX methods for Small-scale, Short-term
Validation and Improvement of User eXperience**

by

Kenji Suzuki

Human-Computer Interaction Laboratory,
Research Group of Synergetic Information Engineering,
Division of Computer Science and Information Technology,
Graduate School of Information Science and Technology,
Hokkaido University
24th January 2022

目次

第1章	はじめに	1
第2章	関連手法とマイクロUX法の構想	6
2.1	アジャイル開発	6
2.1.1	アジャイル開発の特徴	7
2.1.2	プログラミング工程を管理する開発手法	8
2.1.3	プロジェクト進行を管理する開発手法	9
2.2	ユーザ体験	11
2.2.1	ユーザ体験の構造	12
2.2.2	ユーザ体験の調査手法	14
2.2.3	アジャイル開発とユーザ体験調査の相違点と課題	16
2.3	マイクロUX法の構想	18
2.3.1	対象の極小化	18
2.3.2	評価の単純化	18
2.3.3	マイクロUX法の手順	19
第3章	メールサービスで文章作成が難しい問題に対するマイクロUX法の試行	22
3.1	はじめに	22
3.1.1	UX課題の分解	22
3.1.2	改善指標の定義	23
3.1.3	改善手法の検討	23
3.2	関連研究	24
3.3	提案手法：Fix and Slide	25
3.3.1	キャレット操作	26
3.3.2	文字列選択	26
3.4	既存手法と提案手法の比較実験	26
3.4.1	実験で使用する小型携帯端末	27
3.4.2	タスク：文字列選択	27
3.4.3	実験条件	27

3.4.4	実験参加者と手続き	29
3.4.5	既存手法と提案手法の違いの詳細	30
3.4.6	実験結果	31
3.5	初学者の利用の観察実験	34
3.5.1	タスク：文字列選択	34
3.5.2	実験条件	34
3.5.3	実験参加者と手続き	35
3.5.4	観察結果	36
3.6	議論	37
3.7	むすび	38
第4章	地図サービスで片手操作が難しい問題に対するマイクロUX法の試行	39
4.1	はじめに	39
4.1.1	UX課題の分解	39
4.1.2	改善指標の定義	40
4.1.3	改善手法の検討	40
4.2	関連研究	41
4.3	提案手法：Bounce Back と Force Zoom	44
4.3.1	Bounce Back	44
4.3.2	Force Zoom	44
4.4	既存手法との比較実験	45
4.4.1	実験で使用する小型携帯端末	45
4.4.2	タスク：マルチスケールの対象選択	45
4.4.3	セッション：ズームアウト単独とズームアウト/イン複合	46
4.4.4	実験条件：ユーザインタフェース	47
4.4.5	実験参加者と手続き	47
4.4.6	実験用インタフェースの実装	48
4.4.7	実験結果：ズームアウト単独セッション	49
4.4.8	実験結果：ズームアウトとズームイン複合セッション	52
4.5	議論	54
4.5.1	提案手法の制限と本研究で得られた知見の範囲	55
4.6	むすび	56

第 5 章	地図サービスで屋内目的地への移動が難しい問題に対するマイクロ UX 法の試 行	57
5.1	はじめに	57
5.1.1	UX 課題の分解	57
5.1.2	改善指標の定義	58
5.1.3	改善手法の検討	59
5.2	関連研究	61
5.3	提案手法：SCAN	61
5.3.1	概要	62
5.3.2	追跡手法	63
5.3.3	キャリブレーション	63
5.4	予備実験	66
5.4.1	実験で使用する小型携帯端末	66
5.4.2	実験参加者と手続き	66
5.4.3	実験結果	67
5.4.4	考察	68
5.5	マイクロ UX 法の適切な成果物の検討	68
5.5.1	SCAN の発展手法の検討	69
5.5.2	サービスへの導入方法の検討	71
5.6	まとめ	72
第 6 章	総合的な議論	73
6.1	マイクロ UX 法の試行に関する振り返り	73
6.1.1	メールサービスで文章作成が難しい問題	73
6.1.2	地図サービスで片手操作が難しい問題	74
6.1.3	地図サービスで屋内目的地への移動が難しい問題	75
6.2	マイクロ UX 法の考察	77
6.2.1	マイクロ UX 法の手順	77
6.2.2	マイクロ UX 法の成果物	80
6.2.3	マイクロ UX 法の適用制限	81
第 7 章	まとめと将来的な展望	83
付録 A	研究業績リスト	94

第1章 はじめに

ITサービスの開発現場では重要業績評価指標（Key Performance Indicators; KPI）を達成するために、より早くより効果的な開発サイクルが求められている。スマートフォンとスマートフォン用アプリケーション配信プラットフォームの登場により、ユーザは世界中の企業や開発者が開発した膨大な数のアプリケーションから、自分が気に入ったものを手軽にインストールし利用することが可能になった。ITサービスの提供者はより多くのユーザに選ばれるために日々様々なアプリケーションをリリースし、また継続的に利用してもらうために頻繁なアップデートを行っている。このような状況に対応するために、ITサービスの開発現場では様々な取り組みが行われてきた。

ITサービスの開発現場では要求定義から始まり設計、実装、運用を逐次的に行う「ウォーターフォール開発」が主流であったが、より早いリリースや頻繁なアップデートを行うスピード感に対応することが難しくなり「アジャイル開発」を取り入れるようになった。アジャイルソフトウェア開発宣言 [1] において、アジャイル開発は開発速度を低下させる要素を効率化し、ソフトウェアを中心に反復的な設計と検証を柔軟に行うことが重要な価値であると宣言された。アジャイル開発の典型的な開発手法では、サービスの主要な価値を体現する必要最小限のプロダクトを元に、短い間隔で小さな実装を行い反復的に価値の検証と設計を繰り返す。ITサービスの開発現場では、こうしたアジャイル開発の理念や手法を環境や状況に合わせて適用し、開発サイクルの高速化や効率化を図ってきた。そうした中、ITサービスの開発現場ではサービスの品質向上や競合との差別化を図るために、ユーザがシステムを操作する際の接点であるユーザインタフェース（User Interface; UI）と、ユーザがシステムを通して得る一連の体験を指すユーザ体験（User eXperience; UX）への取り組みも進めている。UXはDonald Normanが考案し、「製品がどのように見え、学習され、使用されるか、というユーザのインタラクションのすべての側面を扱う」ための概念として生まれた [2]。UXの調査や設計には多くの企業やデザイナー、研究者が様々な手法を考案してきた。ユーザがどのようなことを体験しているか把握するための手法や、ユーザに提供すべき体験を探索するための手法、ユーザが得た体験がどのようなものであったか評価する手法などがある。いずれもユーザを観察し深い洞察を得る調査手法であり、多くのITサービスの開発現場で用いられている。

アジャイル開発とUXの取り組みは、厳しい競争や激しい変化に対応しサービスの品質

を高める点で共通しているが、これらの両立は難しい現状がある。アジャイル開発は小さく軽い開発を繰り返しながら提供すべき価値を見つけ出すアプローチを取るが、UXの取り組みはユーザの体験を観察し提供すべき価値が何かを見つけ出すアプローチを取る。すなわち、アジャイル開発は作りながら価値を探るアプローチであり、UXは作る前に価値を探るアプローチという、根本的なスタンスの違いがある。また、UX調査は対象が大きくコストが掛かる問題がある。Rotoらは、UXは予期的UX、一時的UX、エピソード的UX、これらを含む累積的UXという時間構造を持ち、サービスに触れている間だけではなくサービスに触れる前から触れた後までが対象になると述べた [3]。Jesseは、UXは抽象度に応じて戦略、要件、構造、骨格、表層という階層構造を持ち、UXは捉える抽象度によって対象の性質が変わると述べた [4]。UXはこうした時間と階層の構造的な特徴を持ち、UXの改善を行う際に取り扱う範囲は幅広く奥深い。さらに、UX調査は定性的な手法特有の問題もある。観察や実験から得られた発話や行動のログは膨大になりやすく、整理するだけでも多くの時間を要しやすい。またこのログから何を見出させるかは観察者の知識や技術に依存しており、同じ手法を用いても得られる結論は必ずしも一致しない。すなわち、結論に対する妥当性の判断が難しい問題がある。UXの改善はサービスの品質向上や競合との差別化が期待されるが、こうしたコストの大きさと効果の見通しが難しいことがリスクとなり、KPIへの貢献実績や利益に直結する案件が優先されUXの改善が後回しにされることが少なくない。本論文では、このような状況に対しアジャイル開発の開発サイクルに並走可能なUX改善のアプローチが有効なのではないかと考えた。すなわち、メインの開発と同時に実施できる規模感であり、メインの開発に適切に還元できる成果物を創出するアプローチである。これを実現するために、UX改善のコストとリスクを小さくし、UX改善の施策と効果を単純化する。本論文では、これをマイクロUX法と名付け、検討を行う。

マイクロUX法では、UX改善の対象を極小化し評価を単純化することで、アジャイル開発の開発サイクルに並走可能なUX改善のアプローチを実現する。UX改善の対象を極小化するために、UX課題をUI課題に分解するアプローチを取る。UXは製品を通じて得られる一連の体験であるが、マイクロな視点で捉えればUI操作という小さな体験の連続によって構成されている。ひとつひとつは小さな体験であるが、この小さな体験に不満を感じサービス途中で離脱してしまうユーザも少なくない。すなわち、UXの課題はこうした小さな体験の課題および累積した結果で構成されていると言える。マイクロUX法ではこの小さな体験に着目し、UXの課題をUIの課題に分解し改善を試みることで対象範囲を狭めコストの低減を目指す。また、UX改善の評価を単純化するために、定量的な指標を用いるアプローチを取る。ITサービスの開発現場の多くは、アジャイル開発を採用し設計と検証を小さく繰り返しながらプロダクトゴールへの到達を目指している。この繰り返し

において、施策と結果の因果関係が明確であることは、次のアクションを検討する上で重要な指針となる。しかし定性的な UX 改善のアプローチは属人性が排除できず、改善要因を明確にすることが難しい。そこで、マイクロ UX 法では UX の改善指標を定量的に定め、これを定量的に評価することで施策と結果の要因の明確化を図る。こうすることで UX 改善の成果物が反復的な設計と検証に効果的に組み込めるようにする。これらを実践する具体的な手順として、マイクロ UX 法は4つのステップを設ける。まず1番目に UX 課題を UI 課題に分解する。UX の課題に関与する UI の課題を抽出し選定する。2番目に UX 改善の指標を定義する。UI の課題が与える定量的な影響を検討し指標化する。3番目に改善手法の開発を行う。改善指標に対し有効な手法を検討し開発を行う。4番目に既存手法との比較を中心とした評価実験を行う。提案手法と既存手法を改善指標を基準に比較検証し有効性を分析する。このようなステップを経ることで、UX 改善の対象の極小化と評価の単純化を実現し、アジャイル開発の開発サイクルに組み込みやすい成果物の創出を目指す。本論文では、こうした構想と手順によって構成されるマイクロ UX 法を用いることで、アジャイル開発の開発サイクルに並走可能な UX 改善のアプローチの実現を目指す。これを検証するために、本論文では実際の IT サービスを題材にマイクロ UX 法を試行し、手順の妥当性や成果物の適性について検討することとする。具体的には、メールサービスで文章作成が難しい問題、地図サービスで片手操作が難しい問題、地図サービスで屋内目的地への移動が難しい問題について取り組み、複数の事例を通じて得られた知見を用いマイクロ UX 法の精緻化を試みる。

本論文の構成は以下の通りである。2章ではマイクロ UX 法の関連研究として、アジャイル開発の理念や典型的な開発手法、UX の概要や調査手法についてまとめるとともに、マイクロ UX 法を提案する背景について述べる。そしてマイクロ UX 法の構想と検証方法について述べる。

3章ではスマートフォン用のメールサービスで文章作成が難しい問題に対しマイクロ UX 法を試行する。この UX の問題を UI の問題に分解すると、小さいディスプレイでは文章全体を俯瞰しながら推敲できない問題や、ソフトウェアキーボードで高速入力が難しい問題、文章のポインティングやセレクションで誤操作しやすい問題などが考えられる。ここでは3番めに挙げた誤操作しやすい問題に着目する。タッチディスプレイでは指が対象を隠してしまう Fat Finger Problem が知られており [5]、文字をタップしてテキストカーソルを移動させたり文字を選択する際に意図しない場所をタップしてしまう主な要因と考えられる。この問題は操作に掛かる時間および操作の利便性に影響を与えると考えられることから、これらを改善指標として定義する。この指標に効果のある手法として、目的の文字の位置へテキストカーソルを移動しようとする本問題が発生することから、逆に目的の

文字をテキストカーソルの位置へ移動する Fix and Slide と呼ばれる手法を提案する。提案手法と既存手法を改善指標を基準に比較実験を行い、実験結果を統計的に分析する。そして、提案手法が文章を作成する上でどのような効果を持つ UI であるか明らかにし、マイクロ UX 法で考案した成果物に関する考察と問題点について述べる。

4 章ではスマートフォン用の地図サービスで片手操作が難しい問題に対してマイクロ UX 法を試行する。この問題に関与する UI の問題として、スマートフォンが大画面化し画面端に配置されているボタンに指を届かせるのが難しい問題や、ズーム操作を片手で実行するのが難しい問題などが考えられ、ここでは後者の問題に着目する。ズーム操作は地図サービスの様々なシーンで多用される操作の一つであるが、これを実行するには片方の手でスマートフォンを保持し反対の手で指と指の間隔を近づけたり遠ざけたりするピンチイン・ピンチアウト操作を行う必要がある。そのため片手が塞がっている状態でズーム操作を行うことは難しく、スマートフォンを両手で持ち直さなくてはならない。この問題は操作に掛かる時間および操作に行う回数、操作の利便性に影響を与えると考えられ、これを改善指標と定義する。この指標に効果がある手法として、スマートフォンに搭載されている圧力感知マルチタッチ技術に着目し、ユーザがディスプレイを押す力量に応じて飛び上がるようにズームアウトする Bounce Back と、力量に応じて遠ざかるようにズームアウトする Force Zoom と呼ばれる手法を提案する。提案手法と一般普及手法、先行研究手法とを改善指標を基準に比較実験を行い、実験結果の統計的な分析を実施する。そして、提案手法が片手で操作する上でどのような効果を持つ UI であるか明らかにし、マイクロ UX 法で考案した成果物に関する考察と問題点について述べる。

5 章では、3 章と 4 章で取り組んだマイクロ UX 法について考察し、マイクロ UX 法の効果的な成果物の形について検討を行う。マイクロ UX 法で考案した手法をサービスに導入する際、既存のシステムやデザインなどと衝突が起こる可能性がある。そこで 5 章では、マイクロ UX 法で考案した手法を実在のアプリケーションに導入し、どのような形であれば効果的に導入できるか検討を行う。5 章ではスマートフォン用の地図サービスで屋内目的地への移動が難しい問題に対しマイクロ UX 法を試行し、導入検討用の手法の実装までを行う。屋内目的地への移動が難しい問題に関与する UI の問題として、屋内の詳細な地図情報が表示されない問題や、屋内で現在地が表示されない問題がある。ここでは特に後者の問題に着目し改善を試みる。通常、多くの地図サービスは GPS の情報を用いて地図上の現在地を特定し表示している。しかし屋内では GPS 電波を十分に受信することが難しく、正確な現在地の表示を行うことが難しい。この問題は目的地へ移動する時間と距離に影響を与えると考えられ、これを改善指標と定義する。この指標に効果がある手法として、ユーザが撮影した屋内地図画像上に現在地を表示し、ユーザの移動量に応じて現在地が更新される SCAN という手法を提案する。5 章はマイクロ UX 法の効果的な成果物の形を検討するこ

とが目的であることから、SCAN の基本的な性能に関する予備実験を行う。SCAN の基本的な性能に問題がないことが確認できた後に、一般に広く普及している地図サービスのスマートフォン用アプリケーションへの SCAN の導入方法の検討を行う。この地図サービスは主要な屋内施設の地図情報が表示できることから、SCAN のカメラによる現在地取得および移動量を用いた現在地追跡機能のみを提供する。3 章と 4 章で得られた考察を元に、SCAN による既存のシステムやデザインへ与える影響を最小限にする方法を検討し、地図サービスのアプリケーションに実際に実装し検証を行う。そこで得られた知見を元に、マイクロ UX 法で考案する手法をサービスへ効率的に導入可能な形にするために考慮すべきポイントについてまとめる。

6 章では 3 章から 5 章で取り組みを検証し、マイクロ UX 法の手順や適切な成果物の形、適用制限について考察を行う。7 章では全体のまとめと将来的な展望について述べる。

本論文では、IT サービスの開発現場の現状と取り組みについて概観し、アジャイル開発と UX のアプローチに関する相違点と問題点を明らかにする。これに対し、小規模で短期間に検証し改善が行えるマイクロ UX 法を構想し、アジャイル開発の開発サイクルに並走可能な UX 改善のアプローチの実現を目指す。実際の IT サービスを題材にマイクロ UX 法を試行し検証と考察を行う。これらについて総合的な議論を行い、マイクロ UX 法の効果的な手順や成果物について考察を行う。そして、将来的な展望と全体のまとめについて述べる。

第2章 関連手法とマイクロUX法の構想

ITサービスの開発現場ではKPIを達成するために、より早くより効率的な開発サイクルが求められている。スマートフォンとスマートフォン用アプリケーションの配信プラットフォームの登場により、ユーザは世界中の企業や開発者が開発した膨大な数のアプリケーションの中から、自分が気に入ったものを手軽にインストールし利用することが可能になった。無料で利用できるアプリケーションも多く、ユーザはコストを気にせず気軽にインストールでき、不満があれば躊躇せずアンインストールできるようになった。ITサービスの提供者は、この膨大な数のアプリケーションの中からユーザに選んでもらえるよう、ユーザのニーズや時代の流行を敏感に感じ取りながら日々様々なアプリケーションを企画、開発、リリースしている。また、継続的に利用してもらうために、SNS (Social Networking Service) やスマートフォン用アプリケーション配信プラットフォームに書き込まれるレビューやレート、アプリケーションのアクセスログなどを確認しながら、ユーザの不満や要求に応えるためにアップデートを頻繁に行っている。このような厳しい競争や激しい変化に対応するために、ITサービスの開発現場では様々な取り組みが行われてきた。その一例としてアジャイル開発やユーザ体験への取り組みが挙げられる。

2.1 アジャイル開発

従来、ITサービスの開発現場では「ウォーターフォール開発」が主流であった。ウォーターフォール開発では、まずシステムや人員など開発に必要な要件を決めるために要件定義を行う。要件定義が決まったら、システムの外見や機能などシステム全体の詳細設計を行う。詳細設計が決まったら、外見や機能を設計通りに実装を進める。実装したプログラムが正しく動作するか単体テスト、結合テスト、運用テストを行い逐次的に確認していく。すべての問題が解決した後にサービスはリリースされ、運用フェーズへ移行する。ウォーターフォール開発は上流から下流へと工程を一つづつ着実に進めることで、システムの品質担保やスケジュールの進捗管理が行いやすいメリットがある。しかし、突発的な変更や頻繁な更新に柔軟に対応することが難しいデメリットもある。そうした中、変化の激しい状況に対応するためにITサービスの開発現場では「アジャイル開発」が取り入れられるようになった。

2.1.1 アジャイル開発の特徴

アジャイル開発は、アジャイルソフトウェア開発宣言 [1] で提言された 12 の原則を基本的な理念とする開発手法の総称である (図 2.1)。この原則は、顧客やチームのコミュニケーションコスト、設計書や仕様書などの作成コストなど、開発のスピードを低下させる要素をなるべく省略・効率化すること、また、動くソフトウェアをプロジェクトの軸として、無理のない進捗を維持しながら変化を柔軟に受け入れ、短い間隔で反復的に開発を行うことを主要な価値としている。



図 2.1: アジャイルソフトウェア開発宣言 (アジャイルマニフェスト) [1].

アジャイル開発は多くの開発者が様々な開発手法を考案し、多くの議論や検討を通じて発展を続けてきた。アジャイル開発には様々な開発手法があるが、典型的な開発手法としてプログラミング工程の管理を行う開発手法としてエクストリーム・プログラミング、プロジェクトの管理を行う開発手法としてスクラムが挙げられる。

2.1.2 プログラミング工程を管理する開発手法

エクストリーム・プログラミング (eXtream Programing; XP) は小規模に設計, 実装, テスト, フィードバックを繰り返し段階的に完成度を高める反復型の開発手法である [6]. XP はコミュニケーション, シンプル, フィードバック, 勇気, 尊重という 5 つの価値を重視している.

コミュニケーションを重視することで, 顧客やチームとの結び付きを強めシステムへの要求や要件の伝達が円滑になり, チーム全体の理解度やプロジェクトの効率性が向上する. シンプルな設計を重視することで, 無駄な実装は行わず将来の負債を省き, 開発にかかる時間と労力を軽減する. フィードバックを顧客から定期的に得ることで開発進行に合わせた機能検証が行え, プロジェクトの柔軟性や着実性を高める. 高い柔軟性は時に大きな変更を伴うことがあるが, 勇気を持って受け入れ前進することが重要になる. これらを円滑に進めるためには, プロジェクトに関わるメンバー全員が互いを尊重する姿勢を持つことが欠かせない. XP はこれらの価値を重視したプラクティスで構成される開発手法である.

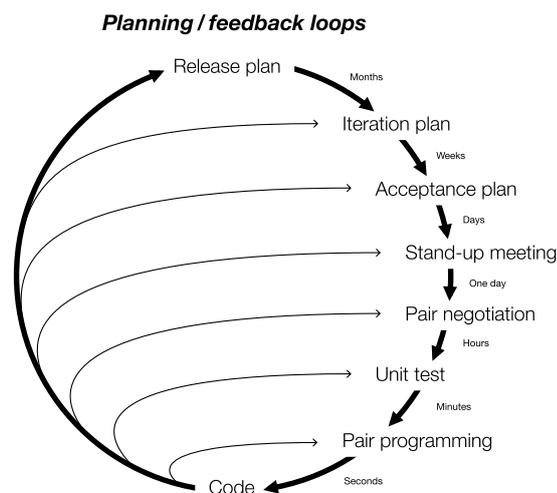


図 2.2: XP (eXtream Programing) の概念図. (この画像は, クリエイティブ・コモンズ・表示・継承ライセンス 3.0 のもとで公表されたウィキメディアの画像「File:Extreme Programming.svg」を素材として二次利用しています.)

XP はプロジェクトメンバーが行うべき・守るべきこととして, 「共同」「開発」「管理者」「顧客」の 4 つの領域に分類される 12 のプラクティスを挙げている. 共同の領域はプロジェクトに関わる全員が対象となり, プロジェクトを小さく反復的に繰り返すためのプラクティスや, プロジェクトを円滑に進めるための知識共有や環境構築のプラクティスがある. 開発の領域は主に開発者が対象となり, シンプルな設計を行うためのプラクティス

や、属人性が高くなったり保守性が下がりコストやリスクを増大する要素を回避するプラクティスがある。管理者の領域は主にプロジェクトの管理者が対象となり、短期集中を促進し負荷をかけすぎない進捗管理を行うためのプラクティスがある。顧客の領域は主に顧客が対象となり、ビジネス視点でプロジェクトやプロダクトの評価を行い適切に指示を与えるためのプラクティスがある。

XP の特徴的なプラクティスとしては、共同のプラクティスである「反復」と、開発のプラクティスである「テスト駆動開発」「ペアプログラミング」がある。反復は、開発期間を1週間から2週間の短いイテレーションに区切り、設計、開発、テスト、リリースなどを反復的に行うことで、手戻りの少ない柔軟な設計を実践するためのプラクティスである。テスト駆動開発は、システムを実装する前にテストを作成することで必要な機能を明らかにし、シンプルな設計につなげるためのプラクティスである。ペアプログラミングは、2人1組になり1人が実装し1人が確認とサポートを行うことで、コード品質と保守性を高めるためのプラクティスである。

XP は5つの価値を重視した12のプラクティスの実践によって、顧客が求めるサービスをより早く実現するための開発手法である。XP の開発手法はプログラミング工程の管理が中心であるため、IT サービスの開発現場ではプロジェクトの管理が中心であるスクラムなど補完関係のある開発手法と組み合わせたり部分的に採用することが少なくない。

2.1.3 プロジェクト進行を管理する開発手法

スクラムはチームが複雑な課題に柔軟に対応し成果を生み出すための軽量なフレームワークである [7]。IT サービスの開発現場は、不確実性が高まり問題も複雑化しており、事前の予測を立て計画を遂行することが困難になっている。このような状況に対応するため、スクラムは適応型ソリューションを採用している。スクラムは、厳密で詳細な指示ではなく、「役割」「イベント」「作成物」といった必要最低限な枠組みだけを提供する軽量なフレームワークである。また、実践と経験による知識を重視する「経験主義」と、価値を生み出さない無駄な事柄を省いて本質に集中する「リーン思考」を重視している。

スクラムは、プロジェクトの状況やプロセスを可視化する「透明性」、プロジェクトの現状を正確に把握し問題をいち早く検知する「検査」、異常を検知したら即対応し逸脱を最小限に修正する「適応」、これら3つを実現することで初めて機能する。あらゆる状況をチームメンバーが互いに把握できる状態にし、頻繁に検査を行い変化や問題の早期発見を心掛けることで異常を速やかに修正でき、プロジェクトの健全性が保たれる。

スクラムでは3つの役割をメンバーに与える。スクラムはプロダクトオーナー (Product Owner; PO) 1人、スクラムマスター1人、開発者数人で構成され、10人程度が適切な規

模とされる。スクラムチームのメンバーは、作業に必要なスキルをチームとして有しており、正しいことを実行し困難に立ち向かう勇気を持ち、互いに尊敬し合うことが求められる。POはプロダクトの価値を最大化する責任を持ち、プロダクトのゴールを実現するために必要なタスクを定めたプロダクトバックログの作成を行う。開発者はプロダクトバックログに挙げられたアイテムを実働する成果物として作成する責任を持ち、完成の定義を満たす成果物を指すインクリメントの作成を目指す。スクラムマスターはチーム及び関係者にスクラムを啓蒙し、スクラムが効果的に機能させる責任を持ち、スクラムが正しく実践される支援や環境整備を行う。

スクラムには5つのイベントがある。スクラムのイベントは成果物の作成と、それに至る検査と適応を行うことを目的に実施される。スクラムのイベントは、スプリントと呼ばれる大きな枠組みであるイベントの中に、スプリントプランニング、デイリースクラム、スプリントレビュー、スプリントレトロスペクティブのイベントが含まれる(図 2.3)。スプリントとは1週間から4週間の間で固定された期間のことで、形のないアイデアを具体的な価値に換える工程である。プロダクトゴールが達成されるまで、開発、リリース、振り返りを反復的に繰り返す。これにより予測可能性や変化への適応性が向上し、プロジェクトのリスクや複雑性を減少させる。

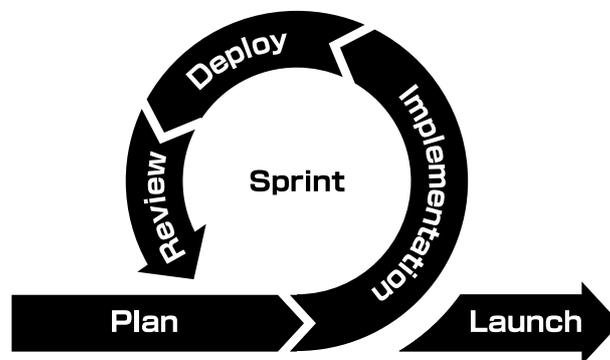


図 2.3: スクラムの概念図.

スプリントではまずスプリントプランニングを行い、このスプリントの目的や意義を決め、どのプロダクトバックログアイテムに対応するか選択する。そして具体的な成果物の作成計画としてスプリントバックログを策定する。成果物はPOと開発者らで合意した完成の定義を満たしリリース可能な状態でなくてはならない。日々の開発ではデイリースクラムを実施し、開発メンバーの進捗の検査を行い、問題があれば計画に現状を適応させる。チームで状況を共有することでプロジェクトの透明性を高める。成果物が完成したらスプリントレビューを実施する。成果物をステークホルダーに提示し、プロダクトゴールへの

進捗やプロジェクトの状況についてレビューを行い、プロダクトバックログの調整や今後のアクションについて確認する。スプリントが一巡したら、スプリントレトロスペクティブ (Sprint Retrospective) を実施し、このスプリントについてチーム全体で振り返り、次回以降のスプリントの品質や効率を向上させる。

スクラムには3つの作成物がある。「プロダクトバックログ」「スプリントバックログ」「インクリメント」の3つであり、各々に達成必須条件である確約 (コミットメント) を持つ。プロダクトバックログは、プロダクトゴールを確約とし、その達成に必要なタスクの一覧である。POが顧客の要求やビジネス環境、チーム状況を勘案し責任を持って作成を行う。スプリントバックログは、スプリントゴールを確約し、スプリントで目指す成果物を作成するために必要な具体的な作業計画である。達成方法は開発者に委ねられており、開発者が開発者のために策定する。インクリメントは、完成の定義を確約に持ち、プロダクトゴールを達成するために作成される具体的な成果物である。インクリメントは完成の定義を満たし正しく機能しリリース可能な状態でなくてはならない。

スクラムは、不確実性が高く複雑化した問題に対応するために、プロジェクトの「透明性」「検査」「適応」を重視する。これを実現するために、チームメンバーに3つの役割を持たせ、5つのイベントで適切に検査と修正を行い、適切な成果物を生み出すための3つの作成物を定めている。スクラムはこうした必要最小限の規約で構成された軽量なフレームワークであり、チームの状況に合わせて運用できる柔軟性も持ち合わせていることから、多くのITサービスの開発現場で活用されている。

アジャイル開発は、開発のスピードを低下させる無駄なコストを省略・効率化し、ソフトウェアをプロジェクトの中心に置き無理のない進捗を維持しながら変化を柔軟に受け入れ、短い間隔で反復的に開発を行うことを主要な価値とする開発手法の総称である。アジャイル開発の典型的な開発手法として主にプログラミング工程の管理を行うための開発手法としてエクストリーム・プログラミング、主にプロジェクトの管理を行うための開発手法としてスクラムがある。ITサービスの開発現場では、その環境や状況に合わせてアジャイル開発の理念や手法を適用し、より早くより効率的な開発サイクルを模索してきた。

2.2 ユーザ体験

ITサービスの開発現場の多くでアジャイル開発が導入される中、サービスの品質向上や競合と差別化する要素としてデザイン、とりわけUIやUXへの取り組みも進んでいる。ITサービスの多くはインターネットを介して提供され、海外製のサービスであっても使いたい時に使うことができる。ITサービスの提供者は世界中の競合との厳しい競争環境に

置かれており、生き残るために自サービスの差別化や優位性の確保を日々模索している。しかし競合サービスは類似サービスでもあり、提供するコンテンツや機能が似やすく差別化が難しい。こうしたことから、見た目や使い勝手といったUIやUXの改善によるサービスの品質向上や差別化に注目が集まっているが、ITサービスの開発現場ではUIやUXの改善は容易に行えない現状がある。

2.2.1 ユーザ体験の構造

Donald Norman はヒューマンインタフェースやユーザビリティという概念ではユーザとシステムの経験のすべてを捉えられないと考え、「製品がどのように見え、学習され、使用されるか、というユーザのインタラクションのすべての側面を扱う」ための概念としてUXという造語を考案した [2]。UX のコンセプトは広く受け入れられ、多くの企業やデザイナー、研究者によって様々な議論が行われてきた。

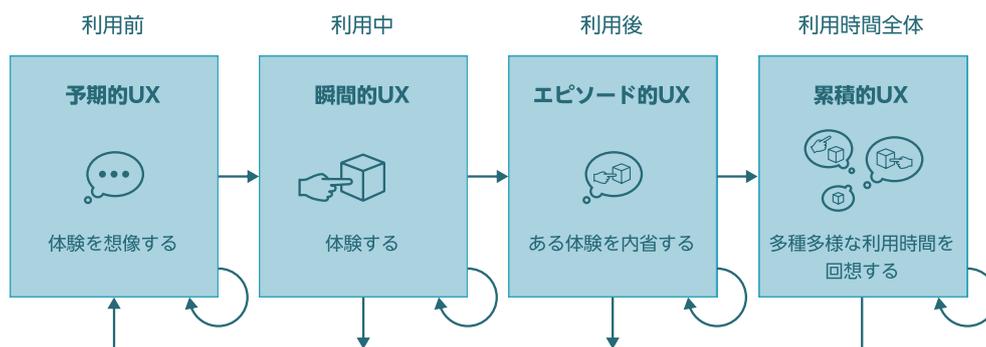


図 2.4: ユーザ体験の時間構造 [3].

Roto らは UX の期間モデルを提示した (図 2.4) [3]。Roto らはユーザが体験する期間によって知覚される UX が異なると考え、この期間を「予期的 UX」「瞬間的 UX」「エピソード的 UI」「累積的 UX」の 4 つに区分した。予期的 UX は、ユーザが製品を実際に使用する前の体験を指す。ウェブサイトレビューを読んだり、店頭で製品を眺めたり、製品に対する期待を高め体験を想像する期間である。製品の評価は製品への期待によって形成される。瞬間的 UX は、製品を実際に使用している最中の体験を指す。製品の使用中は連続的に様々なことを行うため、瞬間的 UX は状況に応じて刻々と変化する。製品への評価は直感的な反応によって形成される。エピソード的 UX は、製品を使用した後に体験したことを振り返る体験を指す。エピソード的 UX は、予期的 UX で予想した期待に対する結果であり、製品を使用するたびに積み重なる。この積み重ねによって製品に対する理解

が深まっていく。製品への評価は使用した結果もたらされた感情によって形成される。累積的 UX は、製品に関わった期間全体を振り返った際の体験を指す。ユーザは予期的 UX、瞬時的 UX、エピソード的 UX のすべてを回顧し、製品に対する総合的な評価を行う。すなわち、製品の総合的な評価は累積的 UX によって形成され、これが別の製品に対する予期的 UX に影響を与える。UX はユーザが体験する期間によって製品に対する評価の経緯や要因が変化するため、UX を検討する際にはその体験がどの期間に該当するか明確にすることが重要となる。

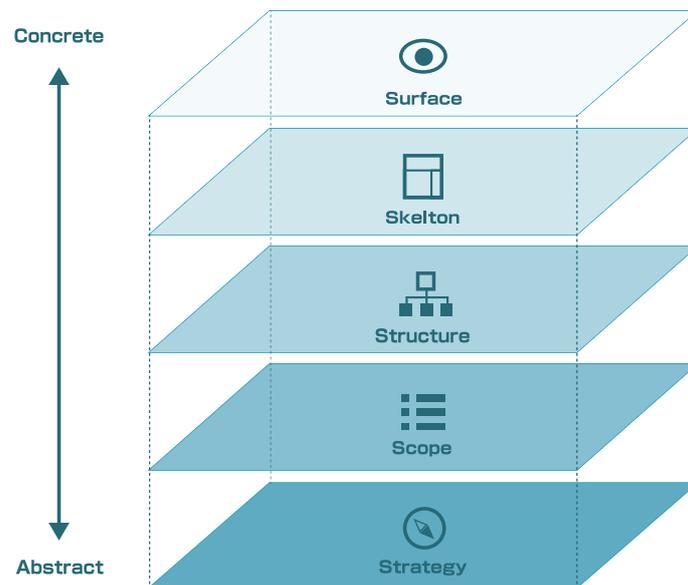


図 2.5: ユーザ体験の階層構造 [4].

Jesse は、UX を「戦略 (Strategy)」「要件 (Scope)」「構造 (Structure)」「骨格 (Skelton)」「表層 (Surface)」の 5 つの階層に分類した (図 2.5) [4]。ユーザが製品から得る UX は、製品の目に見える部分や手で触る部分のみに要因があるのではなく、製品の機能や構造の制約、製品の企画意図からも大きな影響を受ける。製品の課題を検討する際、その課題が見た目の問題であるのか、機能の問題であるのか、そもそもコンセプトの問題なのか、捉える抽象度によって検討する内容は大きく変わる。Jesse は UX を抽象度に応じて 5 つの階層に分類し、それぞれで検討すべきことをまとめた。それぞれの階層は 1 つ下の階層 (抽象度が 1 つ高い階層) に基づく構成になっており、抽象的な概念から段階的に具象化され具体的な製品へと導かれていく。戦略では、ユーザのニーズといった外因的的目的と、ビジネス上の目論見といった内因的的目的に基づいた検討を行う。製品を提供する目的やユーザベネフィットが定義され、製品が目指す根本的な指標となる。要件では、製品が価値を提

供するために必要な機能やコンテンツについて検討する。製品とユーザがどのように関わるか想定し、ユーザにとって必要性和必然性のある機能やコンテンツとする。構造では、ユーザが製品の機能やコンテンツに容易に辿り着き製品の価値を十分に享受できるよう、製品全体の構造や流れを検討する。機能やコンテンツ同士を繋げ適切に構造化することで、ユーザが製品を容易に扱えるようにする。骨格では、ユーザが製品の扱い方を理解しやすくなるよう情報の配置や強弱といった情報のデザインを検討する。ユーザが製品を見たり触れたりする部分、すなわちインタフェースに一貫性をもたせ、混乱や複雑になる要因を排除する。表層では、骨格で検討された意図を視覚的に実現するためのグラフィカルな表現について検討する。骨格で検討したインタフェースの規則と目的に従い、その効果を最大化するビジュアルデザインやインタラクションデザインを行う。UXは捉える抽象度によって検討内容や対応方法が大きく変化するため、UXを検討する際にはその体験が持つ本質的な課題がどの抽象度に依存しているか意識することが重要になる。

ユーザが製品から得る体験は、製品に触れている間だけではなく触れる前から触れたあとまでが対象となる。また、ユーザが製品から得る体験は、表層的な要因だけではなく背景にある戦略的な要因までも含めて対象となる。UXはこうした時間的な構造と階層的な構造を持ち、捉える期間や階層によって検討される体験の内容が変化する。UXが扱う領域は時間的に幅広く階層的に奥深い構造的特徴を持つ。

2.2.2 ユーザ体験の調査手法

UXはNormanによって提唱され、多くの企業やデザイナー、研究者たちによってより良い体験を提供するための様々な手法が考案されてきた。これらには、ユーザが得る体験を深く知り把握することを目的とした手法や、ユーザが求める体験が何か探索することを目的とした手法、ユーザが得た体験がどうであったか評価することを目的とした手法などがある。

UXを把握する手法

UXを把握する手法の一つにエスノグラフィ調査がある。エスノグラフィ調査は、未知の環境にいたり未知の知識を持つユーザを深く理解する手法であり、人々の生活を観察し説明的に記述することと定義されている。観察者は観察対象者の生活や周囲との関わり方などを間近に見て感じることで、ユーザに関する気づきをデザインへより多く反映することを目的としている。また、ダイアリー・スタディもある。ダイアリー・スタディでは、観察対象者に日記を渡し一定期間日々の出来事などを詳細に自分の言葉で記述してもらう。

日常生活における感情や思考の遷移、行動の根拠を知ることができる。他にも、ユーザインタビューがある。観察対象者と直接話をし、経験したことや感じたこと、考えたことを聞き取る。アンケートや観察で得た結果の裏付けを得る目的で行われたり、潜在的なニーズや思考を探る目的など様々なシーンや段階で行われる。

UX を探索する手法

UX を探索する手法の一つにペルソナがある。ペルソナは製品のユーザを代表する架空の人物であり、アンケートやインタビューから得た情報を元に典型的なユーザ像として作り上げる。ペルソナは、ユーザがどのように振る舞うかのヒントを与え、デザインの方向性や根拠を示唆する。プロジェクト内での議論の対象が明確になるメリットもある。他にも、カスタマージャーニーマップがある。カスタマージャーニーマップは、ユーザが製品を体験する一連のストーリーを段階に区切り、段階ごとのユーザの行動や思考、感情を可視化する。製品に触れている間の体験だけではなく製品に触れる前と触れた後の体験まで俯瞰した検討が行え、UX の大きな流れを意識した設計をするためのインサイトが得られる。また、KA 法もある [8]。KA 法はユーザ調査などで得られた定性的な情報から、ユーザが求めている本質的な体験や価値を探る手法である。KA 法は、定性的な情報から出来事を抜粋し、その時のユーザの心の声を想像することで、出来事の裏にある価値を抽出することを目的とする。これらを KA カードと呼ばれるカードに記載し一覧化することで、出来事と価値の妥当性もチェックできる。

UX を評価する手法

UX を評価する手法の一つにユーザビリティテストがある。ユーザビリティテストでは、観察対象者が製品もしくはプロトタイプを用いて実際のタスクを実行する様子を観察し、製品やインタフェースの完成度を評価する。ユーザビリティテストは、ユーザがエラーを起こしやすい箇所や戸惑ってしまう箇所など製品の問題点を明らかにできる。問題点の改善と再テストを反復的に実施することでデザインの品質向上が期待できる。ユーザビリティテストではプロトコル分析（思考発話法）を用いることが多い。プロトコル分析は、観察対象者にタスクを実行してもらい、タスクを達成するために行うすべての動作について、何を見て、何を感じ、何を考え、何故その行動を取ろうと思ったか口述してもらう。プロトコル分析によって製品のどの部分がこういった背景からユーザを惑わせたり不快感を与えているかを知ることができる。

また、ヒューリスティック評価もある。ヒューリスティック評価は、インタフェースの専門家が広く有用性を認められた経験則（ヒューリスティック）を参考にインタフェース

を評価し問題点を発見する手法である。この経験則は数多くのユーザビリティ問題を分析し、背景にあるユーザビリティの原則を抽出・分類したものである。ヒューリスティック評価は複数人の評価者が同じ基準でインタフェースを評価し、それらを照らし合わせ問題点や原因、改善する方向性を示す。ユーザテストと比較して短期間・低コストで実施できるメリットがあり、ユーザテスト前に実施することで既知の問題の発見と解決が行え、ユーザテストでは本質的な問題が発見されやすくなる。一部の経験則が現状のテクノロジーに則さない部分もあり、近年では専門家の知見も用いるエキスパート評価と呼ばれることが多い。

他にも、UX メトリクスがある。Whiteside らは、ユーザビリティ評価の測定法と段階的な目標値を示す手法を提案した [9]。Hartson らはこれを UX のデザインに応用し UX メトリクスを提案した [10]。UX メトリクスは改善の対象となる UX の目標、UX を評価するための UX 指標、それを測るための測定手法を定める。UX 指標は、定めた測定手法で測定した既存システムのスコアを基本水準とし、それより良いスコアを目標水準として設定する。新しいインタフェースはこの目標値の達成を目指し、測定によって UX の目標に対する実現度を確認する。UX 指標には、ユーザがタスク完了までに掛かった平均時間や、タスク実行時の平均エラー回数、SUS (System Usability Scale) などがある。SUS は、10 個の質問で構成されたアンケートを用いて有用性や効率、満足度を測定する。ユーザはアンケートの各質問に対し「まったくそう思わない」と「非常にそう思う」を両端にした 5 段階の尺度から該当する値を選択する。各質問に対する値を決められた計算法で集計すると 100 点を満点としたスコアが算出される。

UX の調査手法には、ユーザがどのような体験をしているか把握するための手法や、ユーザにどのような体験を提供すべきか探索する手法、ユーザが体験をどのように感じたのか評価する手法などがある。これらの手法を用いユーザを実際に注意深く観察することで、ユーザ自身とユーザが得た体験について深く知ることが可能となり、より良い体験を提供するための重要な気づきを得られる。

2.2.3 アジャイル開発とユーザ体験調査の相違点と課題

アジャイル開発はプロジェクトの進行を阻害する無駄なコストを省きソフトウェアを中心に設計と検証を小さく繰り返すことを理念とする開発手法の総称である。XP は、チームのコミュニケーションの円滑化や小さく早い開発を反復的に行うことで、不要なコストや手戻りのリスクを軽減する。スクラムは、メンバーに役割と責任を持たせ、適切なタイミングで検査し適宜軌道を修正しながら、目的の成果物を着実に完成させる。アジャイル

開発は、不確実性が高い状況や複雑な問題に対応するための開発手法であり、小さい開発を繰り返しながら提供すべき価値へ近づいていくアプローチを取る。

UXはユーザが製品から得る一連の体験を様々な面から捉えたものであり、より良い体験を提供するために様々な手法が考案されてきた。UXを把握する手法では、ユーザがどのような体験を得ているか実際に観察することで理解を深める。UXを探索する手法では、ユーザに関する情報を整理し分析することで潜在的なニーズを掘り起こす。UXを評価する手法では、施策に対しユーザがどのように体験したか評価することで改善点を明らかにする。UXの様々な手法は、ユーザとユーザの体験の本質を知るための手法であり、ユーザを深く理解することで提供すべき価値を見つけ出すアプローチを取る。

ITサービスの開発現場の多くはプロジェクトの効率性を向上するためアジャイル開発に取り組んでおり、特にスクラムは様々な組織に適用しやすい軽量なフレームワークであることから広く採用されている。また、サービスの品質向上や競合との差別化を図るためUXの改善にも様々な形で取り組んでいる。いずれもサービスの価値を高める重要な手法であるが、両立するにはアプローチの仕方が根本的に異なる問題がある。アジャイル開発は作りながら提供すべき価値に近づくアプローチであり、UX改善は作る前に提供すべき価値を探し出すアプローチである。そのため、アジャイル開発の開発サイクルとUX改善のアプローチは別々に行うことが妥当であるが、実際に行うのは困難な実情がある。

まず、UX改善はコストが大きくなりやすい問題がある。ユーザー一人一人を注意深く観察するため調査に掛かる時間が長くなり、数人程度の調査でも数週間掛かることも少なくない。また発話や行動などの観察ログは膨大になりやすく、分類や整理を行うだけでも多くの人員と時間が必要になる。さらにそこから様々な分析や議論を経る必要があり、調査の開始から結論を出すまでの道程はとても長い。メインとなる開発サイクルがある中、これだけのリソースを捻出するのは容易ではない。また、UX改善はリスクが大きくなりやすい問題もある。発話や行動などの定性的な情報から何を見つけ出せるかは、観察者の知識や経験、感性に依存しており、同じ手法を用いても同じ結論になるとは限らない。すなわち、手法と効果の再現性が高いとは言えず、導かれた結論の妥当性を判断することが難しい。そのため改善効果の見通しが立てづらく、改善実施に踏み切りにくい。こうしたコストとリスクの大きさから、ITサービスの開発現場ではKPIへの貢献実績のある案件が優先されやすく、UX改善は重要性を認識されながらも後回しにされることが少なくない。

こうしたアプローチの違いやコストやリスクの問題に対し、本論文ではアジャイル開発の開発サイクルと並走可能なUX改善のアプローチが有効ではないかと考えた。すなわち、UXを改善するコストとリスクが小さく、反復的な開発サイクルに組み込みやすい成果物を創出するアプローチである。広範な対象と膨大な調査といったコストを軽減し、高い属人性や低い再現性によるリスクを縮小する。そして反復的な検証を前提とした成果物を創

出し、アジャイル開発の開発サイクルへの適合性を高める。こうすることで、アジャイル開発の開発サイクルに並走しながら UX 改善を行うアプローチが実現できるのではないかと考えた。本論文ではこの UX 改善のアプローチに着目し、構想や手順について検討を行うこととした。

2.3 ミクロ UX 法の構想

本論文では、アジャイル開発の開発サイクルに並走可能な UX 改善のアプローチとして、UX 改善の対象を極小化し評価を単純化する手法、ミクロ UX 法を提案する。ミクロ UX 法は、改善対象を極小化することでコストを小さくし、評価を単純化することでリスクを低くする。そして改善の施策と効果が単純化された成果物を創出し、反復的な設計と検証を効果的に行えるようにする。このようなアプローチを確立し、アジャイル開発の開発サイクルに並走可能な UX 改善の実現を目指す。

2.3.1 対象の極小化

ユーザが製品から得る一連の体験は、ミクロな視点で捉えればインタフェースの操作やコンテンツの閲覧といった小さな体験の連続によって構成されている。ひとつひとつは小さな体験であるが、この小さな体験に不満を感じサービスの途中で利用を止めてしまうユーザは少なくない。メールサービスでメールの作成が難しかったり、決済サービスで決済に躓くなど、サービスの主要な価値に繋がる操作の失敗はサービス全体の印象に大きな影響を与える。ミクロ UX 法ではこのようなサービスの主要なコンテキストを妨げる小さな体験に着目する。メールの作成が難しい問題であれば、文面の見やすさ、文章入力の手軽さ、文字選択のしやすさなど、これらの小さな体験が要因として考えられる。すなわち、UX の課題は UI の課題に小さく分解することが可能であり、ミクロ UX 法ではこの UI の改善を通じて UX の改善を図る。こうすることで UX 改善の対象が極小化され、UX 改善を小規模かつ短期間で実施できるようになる。

2.3.2 評価の単純化

UX の調査手法は定性的なものが多い。これらの手法はユーザを注意深く観察することで体験の本質を深く理解することを助ける。しかし、観察対象者の発言や行動から何を記録し見出させるかは、観察者の知識や経験といった属人的な要素に依存している。このため、同じ手法を用いても同じ結論が得られるとは限らず、手法から導かれる結果の妥当性について判断することは難しい。ミクロ UX 法では、UX の課題が及ぼす影響を定量的に

定義し改善指標として用いる。前述のメールの作成が難しい問題で、文字選択の難しさを要因と仮定した場合、この問題は文章の入力や編集を中断させ完成に至るまでの時間に影響を与えることが想定される。すなわち、タスクを完了する時間を改善指標と定義でき、操作時間の短縮化がUI改善の主たる目的と定められる。こうした定量的な改善指標を定義することで、改善効果の実験と評価が行いやすくなる。タスク完了時間は機械的に記録するだけであり、評価は記録された時間を単純に比較することで行える。こうすることでUX改善の評価が単純化され、UX改善の要因が明らかになり理解できやすくなる。

UX改善の要因が明らかになることで、アジャイル開発の反復的な設計と検証に組み込みやすくなる効果が期待できる。反復的な開発サイクルにおいて施策と結果の因果関係が明確であれば、次の施策を検討する重要な判断材料になる。メールの作成が難しい問題の場合であれば、文字選択の操作時間を短縮する手法による改善が、メール作成数や送信数などのKPIにどのような影響を与えるか観察を行う。KPIへの貢献が認められた場合は操作性の向上施策の拡充を検討すればよいし、貢献が認められなかった場合は操作性以外の改善施策の検討へ移行することができる。このように、UX改善の評価を単純化し要因が明確化されることで、反復的な設計と検証を効率的に行える成果物の創出が可能となる。また、この成果物は事前の実験によって有効性が確認されており、サービス導入時の改善効果の見通しが立ちやすくなる。これによりUX改善のリスクも低くすることができる。

2.3.3 ミクロUX法の手順

ミクロUX法は、UX改善のコストを小さくするためにUXの課題をUIの課題へ分解し対象を極小化する。また、UX改善のリスクを小さくするためにUIの改善指標を定量的に定義し評価を単純化する。これによりアジャイル開発の開発サイクルに並走可能なUX改善のアプローチを実現する。ミクロUX法の手順をまとめると以下ようになる。

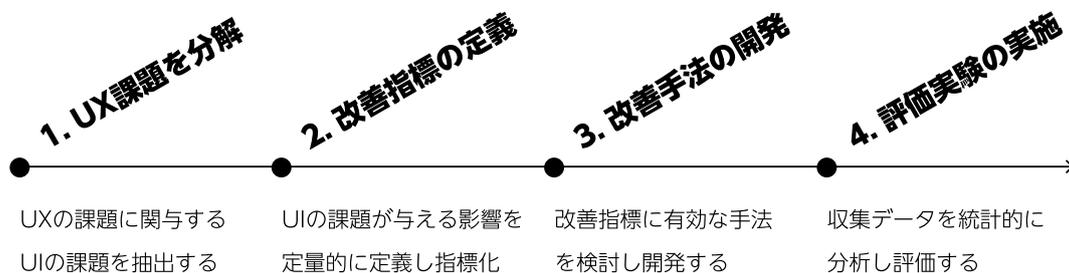


図 2.6: ミクロUX法の手順.

1. UX 課題の分解

マイクロ UX 法では、まずサービスの抱える UX の課題を抽出し、その UX 課題に関与する UI 課題の検討を行う。UX 課題を様々な観点から観察し、それに関与する可能性のある UI 課題をなるべく多く挙げ、検討の幅を広げる。その UI 課題の中から、改善による効果が大きそうなものや検討が行いやすそうな課題を任意に選び、改善対象を決定する。マイクロ UX 法は反復的な開発サイクルへの還元を前提としているため、どの課題を選ぶべきかは問題としない。

2. 改善指標の定義

改善する UI 課題が決定したら、その UI 課題がユーザに与える影響を定量的に定義する。操作の複雑さに問題があるのなら、タスク完了時間や操作回数などへの影響が考えられる。操作の難解さに問題があるのなら、タスク成功率やエラー回数などへの影響が考えられる。こうした指標には Hartson らの UX メトリクス [10] が応用できる。UI 課題の改善指標は複数設け、多角的に検証が行えるようにする。そして改善要因がより明確になるよう配慮する。

3. 改善手法の開発

改善を評価する指標が決定したら、その指標に対し有効な手法の検討を行う。UI の課題がどのように指標に影響を与えているか分析し、その要因を適切に取り除く手法について検討を行う。操作の複雑さがタスク完了時間に影響を与えているのであれば、何が操作を複雑にしているか分析し、その要因を排除した手法の検討を行う。そして、コンセプトを実証するプロトタイプの開発を行い、トライアンドエラーを繰り返しながら精度を高めていく。より良くするという曖昧な目標ではなく、時間や回数といった具体的な数値目標を意識し、目標に対しシンプルに効果のある手法の検討を行う。

4. 評価実験の実施

改善手法を考案しプロトタイプ開発が完了したら、既存手法との比較を中心とした評価実験を行う。改善手法の有効性を確認するために、改善指標を計測するタスクを改善手法と既存手法で実施し、計測された定量データを用いて比較検証を行う。この定量データは統計的に処理し、複数の指標を用いて多角的な分析を実施する。こうすることで分析結果の妥当性が高められ、改善の要因がより明確になる。

マイクロ UX 法は、UX 課題を UI 課題へ分解し、定量的な改善指標を定め、指標に有効な手法の開発を行い、実験結果を統計的に分析する。マイクロ UX 法はこのような手順で構成され、UX 改善の対象の極小化と評価の単純化を行い、反復的な設計と検証を効果的に行える成果物の創出を図る。このようなアプローチを取ることで、アジャイル開発の開発サイクルに並走可能な UX 改善を実現する (図 2.7)。

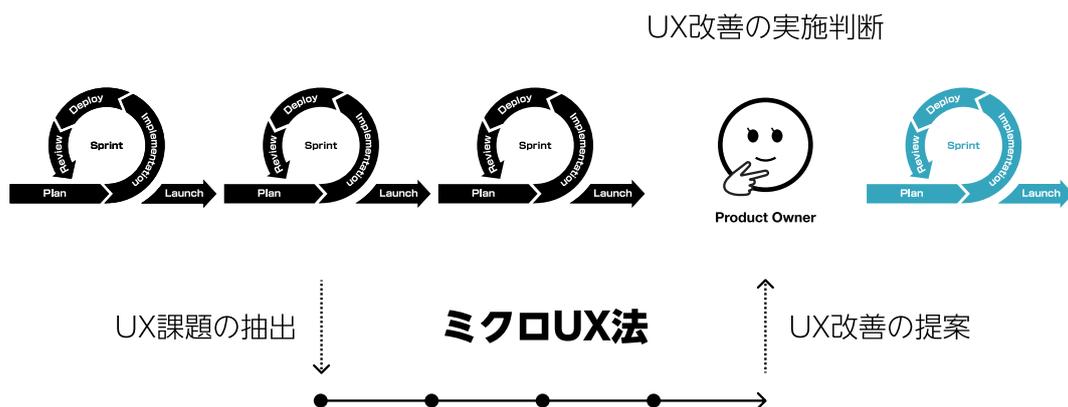


図 2.7: アジャイル開発と並走する UX 改善のイメージ。

本論文では、実際の IT サービスの UX 課題を題材にマイクロ UX 法を試行し、マイクロ UX 法の構想や手順について検証を行う。スマートフォン用のメールサービスや地図サービスを対象に、サービスの主要な価値につながる小さな体験を観察し、UX 課題の UI 課題への分解と改善を試みる。UI 課題の改善手法は定量的な指標に基づき考案し、改善効果は定量的で多角的に評価する。こうしたマイクロ UX 法のアプローチを試行し実践上の問題点や改善点を明らかにする。そしてマイクロ UX 法の構想や手順、成果物について最適化と精緻化を目指す。

第3章 メールサービスで文章作成が難しい問題に対するミクロUX法の試行

3.1 はじめに

本章では、スマートフォン用のメールサービスを題材に、メールやメッセージなどの文章作成が難しい問題に着目し、ミクロUX法を試行する。文章作成が難しい問題を分解し、その中の一つである文章のポインティングやセレクションが難しい問題の改善を行う。改善の指標としてタスク完了時間と SUS スコアを定義し、この指標に有効な手法の検討を行う。

3.1.1 UX 課題の分解

スマートフォン用のメールサービスや SNS などのメッセージングサービスにおいて、円滑なコミュニケーションの提供は最も重要な価値の一つである。しかし、スマートフォンでの文章作成はやりづらさを感じる事が少なくない。その要因として、スマートフォンの小型モニタでは文章全体を俯瞰して推敲できないといったハードウェアに由来する問題や、予測変換候補の精度が低く入力に手間取るといったシステムの実装に由来する問題など、様々な問題が考えられる。その中の一つに、文章の編集のしづらさ、特に文字の挿入や選択がしづらいという UI の問題がある。

個人が日常的に使用する情報端末としてのフロントエンドは、デスクトップ機における web ブラウザからスマートフォンにおけるアプリへとシフトしてきた [11]。これらの端末が受信するメディアは、高解像度な画像や映像にまで及び始めているが、ニュースやメールサービス等の文字を主体とするものも未だ根強い。しかし、スマートフォンに代表される小型携帯端末の主要インタフェースであるタッチパネルは文字列操作に向いていない。なぜならば、小型のモニタに映し出される小さい文字群に対して、それを指示する指先はあまりに大きく、いざ正確なポインティングをしようとする際には「指が操作対象を隠してしまう」いわゆる“Fat Finger Problem”が発生するためである [5]。

この問題は文字列の選択を行う場合にも生じる。一般的な小型携帯端末において文字列選択をする場合、ユーザはキャレットと呼ばれる文字列用のカーソルを選択開始位置に移動させる。続いて長押し等で文字列選択モードにした後、キャレットを選択終了位置まで

移動させる (図 3.1 左). しかし, Fat Finger Problem が一度のタップで正確なキャレットの配置を困難にさせるため, 選択開始位置に配置するまでに何度かタップやドラッグを繰り返す必要がある. 同様に, 文字列選択の終了位置までキャレットを移動させる場合も一度で正確な操作が難しく, ここでも調整のための操作を繰り返す必要がある. つまり, 一般的に小型携帯端末における素早く正確なポインティングや文字選択はデスクトップ機よりも困難である.

本章ではこの問題に着目し, メールサービスで文章作成が難しい問題に対し, キャレット操作の改善を行うこととした.

3.1.2 改善指標の定義

文字の挿入や選択がしづらい問題の改善を行うに当たり, 定量的な改善指標の検討を行った. 文章を作成する上で文字の削除や文章のコピーペーストは多用される操作の一つである. これらの操作を行うためにはキャレットを目的の位置へ移動する必要があるが, Fat Finger Problem により正確な操作を一度に行うことは難しい. すなわち, キャレット操作を行おうとする度に不要な操作の繰り返しや注意を払う必要があり, 文章完成までに掛かるコストを増大させている可能性が高い.

そこで, キャレットを効率的に操作できるようになれば, 文章作成が効率化され円滑なコミュニケーションの提供に貢献すると考えた. すなわち, より速くより効率的な操作性が実現できればコストの低減が図られると考えた. 速さはタスク完了時間で, 使いやすさは SUS スコアを用いることで定量的な計測が可能であり, これを改善を判断する指標とすることとした.

3.1.3 改善手法の検討

本章では, 小型携帯端末における文字列選択操作について, 対象となる文字列を直接触ってキャレットを操作する直接操作手法ではなく, 文字列全体を背景として移動させキャレット位置を間接操作する Fix and Slide と呼ぶポインティング手法を提案する (図 3.1 右). Fix and Slide では, 画面のどこをドラッグしても文字列全体を移動させることが出来る. この時キャレットはドラッグ開始時に位置する画面上の点 p で固定され, ドラッグ操作が終わり指を画面から離れた時, p に位置する文字と文字の間にキャレットがくっつき, キャレットを含む文字列全体は元の位置に戻る. このドラッグ操作中はキャレットが指で隠れないため Fat Finger Problem は発生せず, 適切な位置にキャレットを移動させることが可能となる. この Fix and Slide の有用性を調査するため, プロトタイプシステムを iOS 上で実装し, 調査実験を通して作業実行時間計測やユーザビリティ調査を行った.

まず、日常的に iOS を使用している成人に対して量的な比較検討を行い [12]、その後、小型携帯端末上での文字列ポインティングを行ったことがない児童に対して質的な観察を実施した。

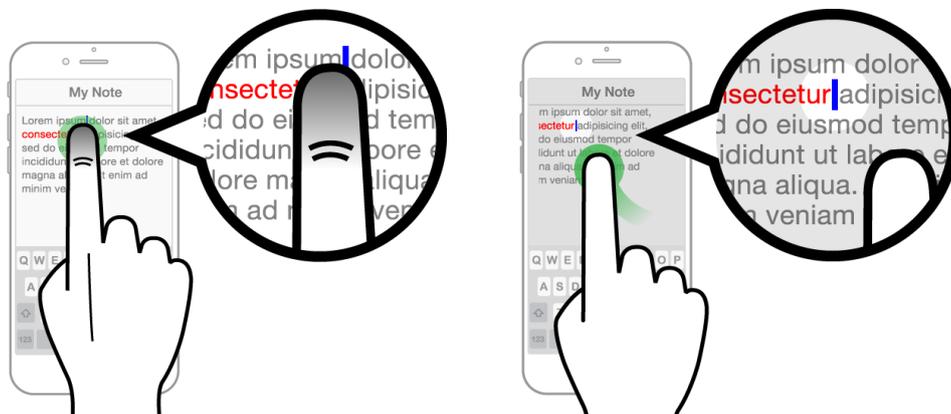


図 3.1: 一般的なcaret操作手法（左）と提案手法（右）：ユーザは文字列全体を背景としてドラッグすることで、任意の場所にcaretを移動することができる。

3.2 関連研究

Fat Finger Problem は小型携帯端末のインタフェースに関する研究においてよく知られた問題であり、これまでに様々なアプローチの研究が行われてきている。Shift はその中でも最も有名な研究であり、この手法では画面を触っている指の下の領域、すなわちユーザが画面を見る際に指が隠してしまう領域が「ふきだし」として指の周囲に表示される [5]。本手法は現在の iOS 等に搭載されている。TapTap は画面をタップすることにより、タップされた領域を拡大して表示するポップアップを利用した手法である [13]。他にもタッチ操作可能な機器に対してオフセットカーソルを利用する手法が存在するが、直接操作手法よりも操作時間が遅かった [14]。これまでは小型携帯端末の「表面」のタッチスクリーンを利用するものであったが、反対の「裏面」を活用した研究も存在する [15]。Hasan らもモバイル端末の裏面を用いた研究を行っている [16]。これらの研究はポインティング操作において Fat Finger Problem を回避するための手法であり、caret 操作や文字列選択に注目した研究ではなかった。

正確なポインティングに関する手法についても研究が行われてきている。これまでに高精度にタッチ操作をセンシングする技術 [17] や、確率モデルを採用してタッチされるポイントを予測する手法 [18]、マルチタッチスクリーン上で両手を利用して正確に対象の選択を行う技術の提案 [19]、タブレットを握る手で画面上の離れた位置をポインティングする手法の研究 [20] が行われてきている。最近になって Eady らによって小型携帯端末上の

キャラット操作に関する手法について提案 [21] があったが、端末の周囲に曲げセンサを取り付ける必要があるなど、端末のハードウェア的な拡張が必要であった。また、形態素解析を用いた日本語文字列選択手法についての提案 [22] もされてきているが、それぞれ本研究の目的とは大きく異なる。また小型携帯端末のモニタ内での仮想スティック [23] やジェスチャーを用いた文字列選択手法 [24] も提案されている。これらは小型携帯端末でのキャラット操作や文字列選択の手法であるが、本研究においては「文字列の背景を動かすことによるキャラット操作手法および文字列選択手法」の検討を行う点で特徴的である。

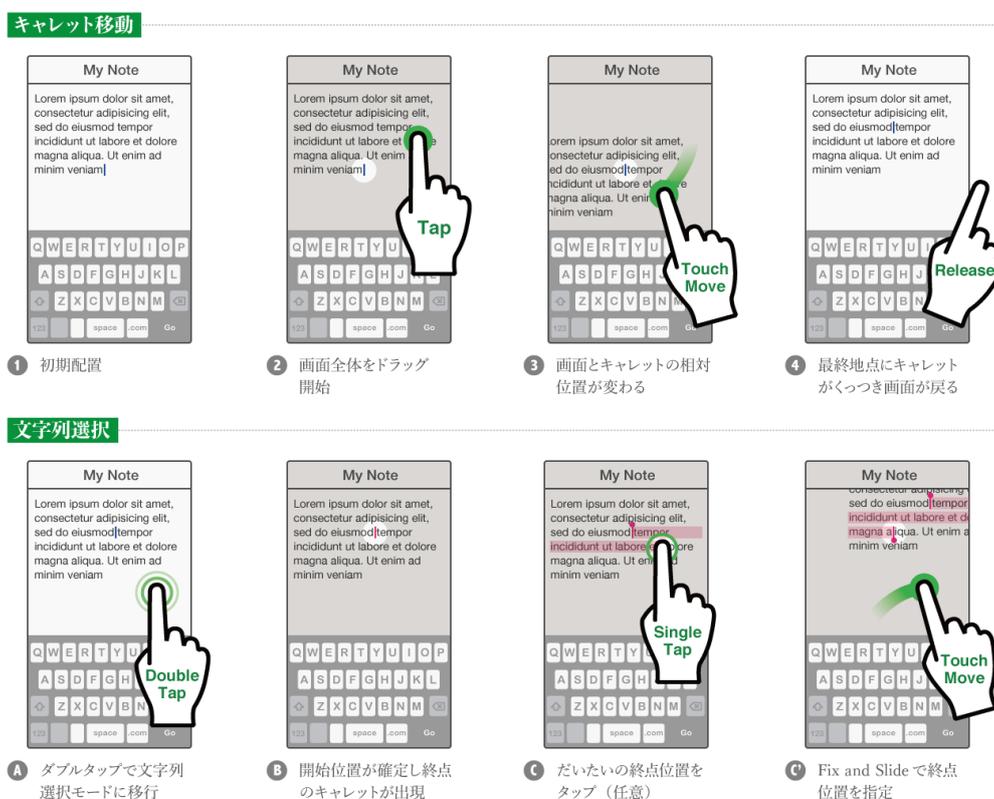


図 3.2: 提案手法の概要. キャラット移動 (上段) と文字列選択 (下段). タッチを起点に、キャラットを固定 (Fix) し、背景文字列を移動 (Slide) することで、キャラット位置を操作できる。

3.3 提案手法：Fix and Slide

本章で提案する Fix and Slide の主たるコンセプトはキャラットの移動や文字列選択手法として文字列全体を背景として操作するということである (図 3.2)。以下でそれぞれの操作について説明する。

3.3.1 キャレット操作

小型携帯端末上でキャレット操作を行う際に問題となるのが、ユーザがスクリーン上をタップしても文字列中の目的の位置に一度で正確にキャレットを置けないことである。この場合、多くは目的の位置のすぐ側に置かれてしまうため、何度かタップするか少しずつドラッグするかのどちらか、もしくは両方の操作を行うことが求められることが多い。目的の位置が指の下に来てしまうようなインタフェースの場合には更に操作が難しくなる。前述した Shift[5] を適用した方法の場合には、指の周囲の適切な場所に「ふきだし」が表示され、その中に指の下に位置する情報が提示されるが、操作の難しさを解決することには直結しない。我々はこれが直接操作手法の問題点であると考えている。

これに対して本章で提案する Fix and Slide は間接操作手法である。先ずユーザは直接操作手法と同様にスクリーンをタップしてキャレットを配置する (図 3.2 上段)。その後、スクリーン上の任意の場所にタッチしてドラッグすることで、文字列全体を背景として動かすことが出来る。この時、キャレットは最初にタップした時点での位置に固定され移動しない、すなわち背景のみが移動することになる。ユーザはキャレットを文字列中で移動させるのではなく、移動する文字列 (背景) によってキャレット操作を行うことが本提案手法の特徴である。

3.3.2 文字列選択

ユーザが文字列選択を開始したい位置にキャレットを移動させた後、文字列選択操作に移行する。画面をダブルタップ (画面を素早く 2 度触る) することで、キャレットの色が変わり、文字列選択モードに変更することが出来る (図 3.2 下段)。続いて、文字列選択を終了したい位置にキャレットを移動させる。これはキャレット操作手法と同様で、最初にスクリーンをタップし、そのままドラッグすることで文字列全体を背景として移動させ、目的の位置へと導く。この操作はダブルタップを挟んでキャレット操作手法を 2 度実行していることと同義である。

3.4 既存手法と提案手法の比較実験

提案手法の有効性を検証するために、現在一般的に利用されている既存のインタフェース、具体的には iOS 上で実装されているインタフェースとの比較実験を実施した。ここでは文字列選択をタスクとして扱うことにした。すなわち、実験参加者は Fix and Slide と iOS が標準で提供している文字列選択手法を使用して文字列選択を行うよう指示される。以降は iOS 上で実装され利用可能な文字列選択手法を既存手法と呼び、Fix and Slide を提

案手法と呼ぶ。実験は、この両手法における文字列選択操作を行う際の作業実行時間（パフォーマンス）およびユーザビリティに関するフィードバックを得ることを目的とした。

3.4.1 実験で使用する小型携帯端末

Fix and Slide は iOS バージョン 8.2 の iPhone 6 上で実装した。ディスプレイは解像度 1,344 × 750 で 4.7 インチだった。実験参加者へは自由な姿勢でスマートフォンを持つよう促し、片手で持つことも両手で持つことも指示していない。実験は既存手法と提案手法それぞれが実装された iOS アプリケーションで行い、このアプリケーション内で作業実行時間を自動で計測し収集した。

3.4.2 タスク：文字列選択

本研究で実施する実験では、文字列選択タスクを採用する。使用する言語は英語とする。英語は単語間がスペースで区切られていることが特徴である。一方で、日本語は単語間がスペースで区切られておらず、句読点で文章の終わり、もしくは文章の切れ目を表現することが特徴である。今回の実験では「日本語の文字列選択は、英語における長い単語の途中を切り出すこと」と見做すことができることに注目し、英語を使用することとした。

実験で使用する文字列はランダムに生成される。本実験では生成された文字列のうち、最初の 2 から 4 段落を使用する。ただし、後述するように特殊なケース、具体的には URL については手動で挿入する。実験において我々は 5 つの文章を用意した。

3.4.3 実験条件

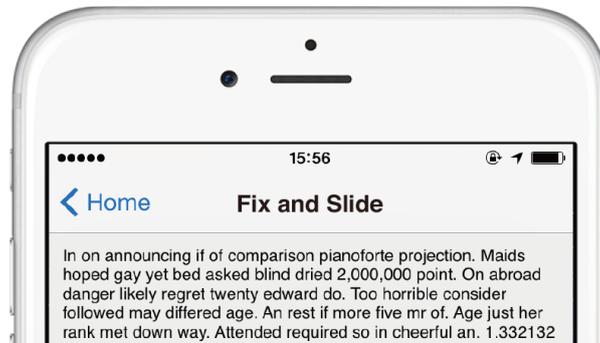
実験では 2 つの文字列選択手法（提案手法と既存手法）に加えて、以下の条件を設定する。

フォントサイズ

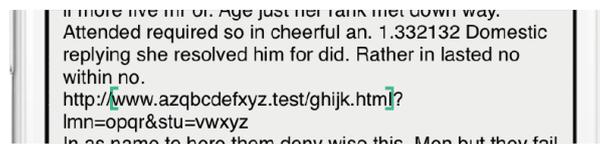
文字列選択タスクの作業実行時間はフォントサイズに大きく依存することが考えられる。このため、本実験では 12 ポイント、14 ポイント、16 ポイントの 3 つのフォントサイズを用意する（図 3.3）。これらのフォントサイズは Apple 社の iOS Human Interface Guidelines[25] に従って選択した。すなわち：

“ユーザが「極小」を選択しても、11 ポイント未満にしないでください。ちなみに「本文」スタイルでは、「大」のとき 17 ポイント（デフォルト値）で表示するようになっています”

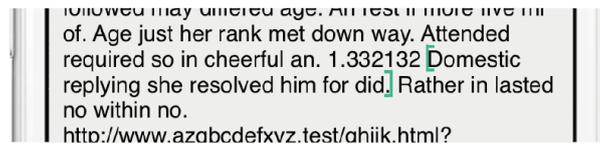
とされており, 12, 14, 16 ポイントは妥当なフォントサイズの選択であると考えている.



12pt (文字列の一般例)



14pt × 部分選択の例



16pt × 外側選択の例

図 3.3: 実験で使用するフォントサイズ 3 種:12 ポイント (上段), 14 ポイント (中段), 16 ポイント (下段). 中段の図は文字列の部分選択条件の例を示し, 下段は外側選択条件の例を示している.

文字列選択の位置

フォントサイズと同様に, 文字列中のどの部分を選択するのも重要な要素となることが考えられる. 本実験では「文字列の部分選択」と「文字列の外側選択」の 2 条件を設定する.

文字列の部分選択 現代的な小型携帯端末用の OS においては, 文字列を選択する際にユーザの指の動き, すなわちキャレットの位置を認識し, 文字列の最初 (文字列間にスペースがある部分) もしくは最後 (同様に文字列間にスペースがある部分) に賢く移動 (スナッピング) する機能が実装されている. 一方でそれは単語や文章単位で選択を行う場合のみ有効である. すなわち, 文字列の部分選択を行う際にはあまり役に立たない. 例えば, 英

語の場合には数字の部分的な選択や、長い URL でホスト名だけを抜き出す場合には有効に機能しない。日本語の場合には基本的に部分選択となる為、いわゆる賢い選択の恩恵を受けることはあまりない。さらに悪いことに、賢い選択の場合には文字列の最初と最後に吸着（スナップ）する性質があり問題が生じることもある。前述した例と同様に、部分選択の例を表 3.1 に示す。

表 3.1: 文字列選択の位置

部分選択	Admitted add peculiar get 83,566 joy doubtful. http://www.Koqiksmkis.test/a.php?q=gao d=dplzk http://www.q=gao hksllaz.test/test.html?x=ddsa
外側選択	Maids hoped yet asked blind dried 2,000,000 point. Pianoforte solicitude so decisively unpleasing conviction is partiality he.

文字列の外側選択 現代的な OS に搭載されている賢い機能（単語の最初と最後を認識し、キャレットがジャンプしたりする機能）は、一般的にユーザは文字列の最初から最後までを選択することを考えると、この機能によりユーザに素早く文字列選択することを可能としている。ここで、この様な文字列選択を「外側選択」と呼ぶ。

実験において参加者は、表 3.1 の外側選択に示される文字の緑色の背景となっている部分を選択するように指示される。実際の表示では図 3.3 のように最初と最後の緑色の括弧が表示される。実験参加者が指示通りの範囲を選択した場合にのみ、次の問題を開始するダイアログが表示される。

3.4.4 実験参加者と手続き

30名のボランティアが実験に参加した。男性は19名、女性は11名であり、年齢は23歳から42歳であった（mean=30.6, S.D.=4.9）。全ての実験参加者はスマートフォンを所有しており、1名がAndroid端末（3.3%）、23名がiPhone（76.7%）を、両方を所有しているのは6名であった（20.0%）。すなわち、1名を除いて全ての参加者がiPhoneを所有していた。また、全ての実験参加者はスマートフォン上での文字列選択を行ったことがあり、20名については文字列選択に慣れていると回答した。どのような場面で文字列選択を行うかについては、URLをコピーする場合、インターネット検索をする場合、ツイートやブログに書く場合と回答した例が多かった。さらに、多くの参加者が文字列選択する際に、指で選択する領域が隠れてしまうことに不満を持っており、同様にiOSの「賢い選択」によって不用意にキャレットがジャンプしてしまうことに不満を持っていた。

実験の手続きについて簡単に述べる。実験参加者は既存手法と提案手法の両方をそれぞれ6回ずつ使用する。実験では3つのフォントサイズと、2つの文字列選択の位置の条件が存在する。文字列選択手法2種類とフォントサイズ3種類については、参加者が経験する順番のバランスを取った。一方で、文字列選択の位置2種類については学習効果がないため固定とし、最初に部分選択タスク、続いて外側選択タスクとした。参加者は1つの試行について、5つの文字列選択タスクを行う。各タスクは実験用アプリケーション内で1タスク毎に時間を自動で計測される。以上を簡単にまとめると、1つの手法について都合30回の文字列選択を実施し、実験参加者は合計で60回の文字列選択を行う。文字列選択を行う位置や範囲（文字数）はランダムではなく、統制を取るために予め決められており、被験者間で同じ文字列を選択することとなる。それぞれの手法を利用する前には、操作に慣れるためのトレーニングの時間を設けた。この時使われている文字列は本番のものとは別のものであった。

3.4.5 既存手法と提案手法の違いの詳細

既存手法

iOSには標準機能として下に挙げる文字列選択手法が既実装されている [26].

- 単語を2回タップすると単語全体が選択される
- 単語を4回タップすると段落全体が選択される
- 二本指でタップすると文章が選択される
- 選択したい文章の先頭を2回タップするとすぐ選択範囲伸縮が開始される

上に挙げた4つのタップを用いた選択手法に加え、我々はもう1つ手法を追加した

- ロングタップで選択アイテムを持つポップアップメニューを表示

本実験では最も慣れた選択手法と提案手法を比較するために、文字列選択のやり方について指示はせず、実験参加者へ iPhone を普段通り使うよう伝えた。

提案手法

実験での提案手法は、iOSが標準で提供する文字列選択やキャレット移動等の各種機能は使えるが、文字列選択モードに変更するためにダブルタップによる選択機能を実装していない。ユーザはスクリーンをダブルタップし、モードを変更することで文字列選択を開始できる。

3.4.6 実験結果

本実験においては、各タスクの作業実行時間とユーザビリティ評価に注目して分析を行う。

作業実行時間

各条件の1試行(5タスク)における作業実行時間[秒]の平均について、フォントサイズ毎にまとめたものを図3.4に示す。既存手法および提案手法の操作手法毎に、部分選択、外側選択と比較するように示している。これに対して3要因分散分析(被験者内計画; 独立変数として、操作手法、フォントサイズ、および文字列選択の位置。従属変数として、作業実行時間)を実施した。この結果、有意な2次の交互作用(操作手法×フォントサイズ×文字列選択の位置 $[F(2, 58)=8.60, p < .01]$)と、1次の交互作用が確認された(操作手法×文字列選択の位置 $[F(1,29)=16.07, p < .01]$ 、および、操作手法×フォントサイズ $[F(2,58)=3.41, p < .05]$)。また、単純主効果の操作手法 $[F(1,29)=11.19, p < .01]$ とフォントサイズ $[F(2,58)=9.55, p < .01]$ に有意差が確認された。文字列選択の位置については有意差が確認されなかったが、有意傾向が確認された $[F(1,29)=3.13, p < .1]$ 。

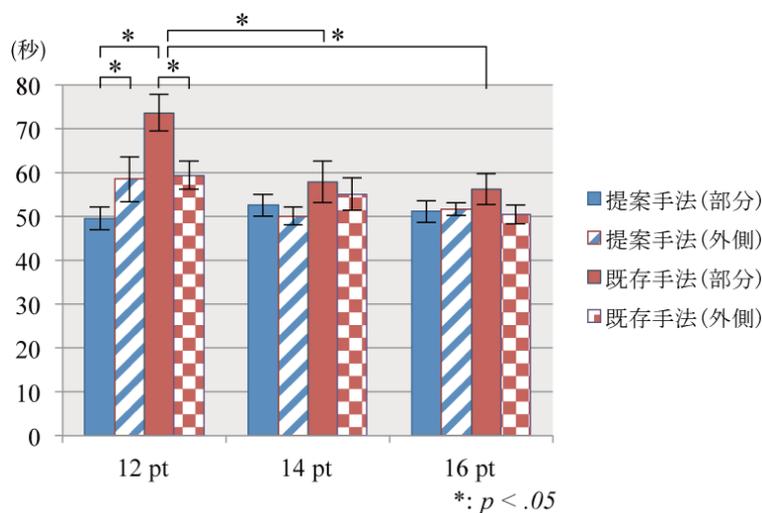


図 3.4: 3 要因分散分析の結果と標準誤差

操作手法×文字列選択の位置 2次の交互作用について、操作手法と文字列選択の位置に注目して2要因の分散分析を行った(被験者内計画)。この分析は3つのフォントサイズ毎に実施した。この結果、12ポイントの場合に交互作用が有意であった $[F(1,29)=25.75,$

$p < .01$] (図 3.4 の 12 ポイントの部分). そこで, 各要因の単純主効果を分析した結果, 操作手法×文字列選択の位置 (部分選択) [$F(1,29)=51.50, p < .01$], 文字列選択の位置×操作手法 (提案手法) [$F(1,29)=6.68, p < .05$], 文字列選択の位置×操作手法 (既存手法) [$F(1,29)=16.69, p < .01$] について有意な差が確認された.

12 ポイントの場合に, 提案手法の部分選択と外側選択との間で有意差が確認された. 実験時の様子から, これは英単語間にあるスペースの影響が考えられる. 12 ポイントのスペースは幅が狭く, スペースを含んでいることに気づかないまま選択をしてしまう, もしくは回避するために微調整が必要とされるシーンが多く見られた. これが原因として作業実行時間に有意な差が生じた可能性がある.

操作手法×フォントサイズ 図 3.5 に操作手法×フォントサイズについて, 文字列選択の位置に注目した結果を示す. 部分的な文字列選択の場合, 提案手法はフォントサイズに依存せず, 作業実行時間は一定だが, 既存手法の場合はフォントサイズが大きくなるにつれて作業実行時間が短くなる傾向が確認された. ここで部分選択に注目して 2 要因分散分析 (被験者内計画) を実施した結果, 交互作用が確認された [$F(2,58)=8.89, p < .01$]. Bonferroni 法を用いた多重比較を実施した結果, 既存手法の場合, フォントサイズが 14, 16 ポイントの場合に比べて 12 ポイントの場合に有意に作業時間が長いことが確認された ($MSe=206.26, p < .05$). また, 外側選択に注目して 2 要因分散分析を実施した結果, 主効果 (フォントサイズ) が有意であった [$F(2,58)=5.82, p < .01$]. ここで Bonferroni 法を用いた多重比較を実施した結果, 12 ポイントの場合が有意に長い結果であった ($MSe=179.54, p < .05$).

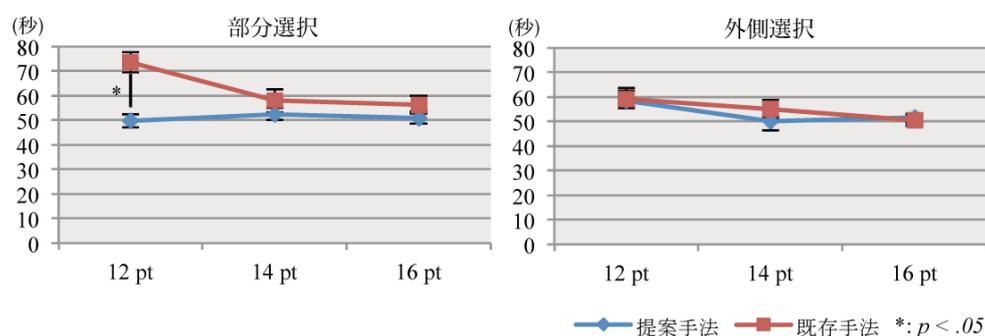


図 3.5: 操作手法×文字列選択の位置の結果と標準誤差

ユーザビリティ評価

続いて、各操作手法を使用した後に実験参加者に回答を依頼したアンケートの分析を行う。ここでは System Usability Scale (SUS) [27] をもとに、本実験用に微修正したアンケートを実施した。SUSは5段階のリッカート尺度の質問10項目で構成される、ユーザビリティ評価のためのアンケートであり、最終的に100点満点でスコアが得られるものである。この結果、提案手法は68.67点、既存手法は49.5点と評価され、これらについて分散分析を実施した結果、提案手法が有意に高い結果であることが確認された [F(1,29)=19.62, $p < .01$]。それぞれの項目についても分散分析を行った結果、Q1: “頻繁に使いたいと思った” [F(1,29)=50.27, $p < .01$], Q3: “使いやすいと思った” [F(1,29)=28.29, $p < .01$], Q9: “自信を持って使うことができた” [F(1,29)=11.15, $p < .01$] で有意な差が確認された (図 3.6)。また、Q5: “この操作方法がスマートフォンの操作方法としてよく統合されていると思った” という項目でも有意な差が確認されたことも併せて、提案手法は既存手法と比較して使いやすく、またユーザはスマートフォンのインターフェースとして提案手法を統合していくことを歓迎していることが示された。

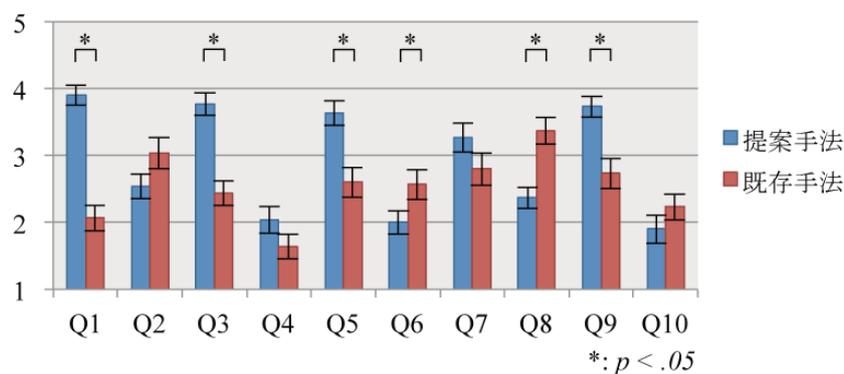


図 3.6: SUS の結果と標準誤差

実験中の参加者の観察結果

既存手法において実験参加者の多くは、まず指示された範囲の始点もしくは終点近辺をロングタップし、タッチ領域を拡大表示させた状態で指をスライドし、キャレットを目的の位置にゆっくり導いた。続いて指を離した際に表示されるポップアップメニューから選択機能を実行し、選択された範囲の前後にあるグラブポイントをドラッグしタスクを完了させることが多かった。

提案手法においては実験参加者の多くは、まず文字列全体を大きく動かし指示された範

圏の始点もしくは終点付近にキャレットを近づけ、至近距離となるとゆっくり操作することが多かった。続いてダブルタップし選択モードに移行後、同様の操作を行いもう一方の指示点に合わせていた。回数を経るに従って一回の操作で指示された位置に直接合わせられるようになった参加者も少なくなかった。

3.5 初学者の利用の観察実験

本実験では小型携帯端末の操作に不慣れである児童を対象とした観察実験を実施した。対象は Fat Finger Problem を強く経験したことがない小型携帯端末の未習熟者であり、タスクとして提案手法を自由に操作してもらい、提案手法がどう使われるか観察し、有用性を検討することを目的としている。このため、文字列操作を全く行ったことがないユーザの質的な観察が主目的であり、量的な作業実行時間は計測しなかった。本観察実験は前述の比較実験と同じ機器およびプロトタイプシステムを用い、タスクと条件を簡略化した内容で実施した。

3.5.1 タスク：文字列選択

観察実験で実施する実験では、文字列選択タスクを採用し、使用する言語は日本語とした。今回の実験では「小型携帯端末の未習熟者が提案手法をどのように扱うかの観察」が目的であるため、実験参加者が読みやすい文章として、小学校低学年向けに分類される童話とした。

3.5.2 実験条件

観察実験では2つの文字列選択タスク（提案手法と既存手法）に加えて、以下の条件を設定した。

フォントサイズ

観察実験は選択操作の観察が目的であるため、iOS Human Interface Guidelines[15]における「大」サイズ相当の18ポイントとした。これはタスク実行時に文字の読みづらさによるタスクの難易度上昇を招かない妥当なサイズと考えている。

文字列選択の位置

フォントサイズと同様に意図せぬタスクの難易度上昇を防ぐため、文字列選択の対象は文章単位とした。すなわち、選択してもらう範囲は一文の文頭から文末までとなる。文字列の選択タスクは、画面上に表示されている文章と同じものが印刷された B5 サイズの紙に蛍光ペンでマークし指定した（図 3.7）。実験参加者はこれを見ながら指定された範囲を選択する。

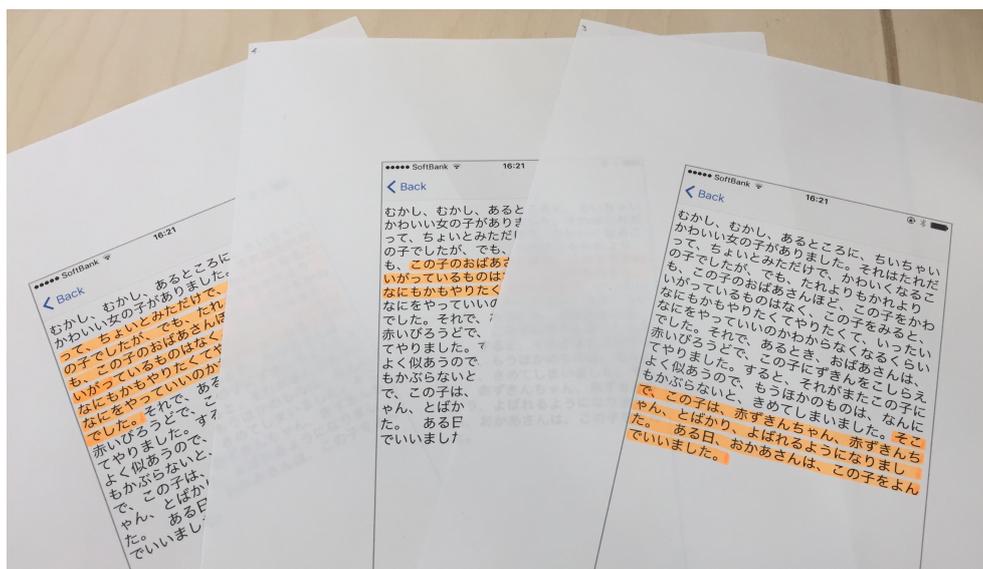


図 3.7: 文字列選択範囲を指示する紙面の例

3.5.3 実験参加者と手続き

3名のボランティアが実験に参加した。男子児童3名で、年齢は7、9、11歳であった。11歳の児童は1人で、7、9歳の児童は2人で実験を行った。

全ての実験参加者はスマートフォンを所持していなかった。7歳と11歳の男児は頻繁に、9歳の男児は時々保護者が所有するスマートフォンを用いゲームを行った経験があった。全ての実験参加者はスマートフォン上で文字列のポインティングと選択操作をほぼ行ったことがなかった。ゲームや動画等のコンテンツを検索する際は、サービス上に初期表示されるレコメンドから辿る手法を取っており、文字列操作に関する Fat Finger Problem を強く体験したことはほぼなかった。

実験の手続きについて簡単に述べる。実験参加者は2つの文字列選択手法（提案手法と既存手法）のやり方をそれぞれステップバイステップで教わる。続いて、実験参加者に選

択範囲が示された紙を渡し、そこにマークされた箇所をプロトタイプシステム上で選択してもらった(図 3.8)。実験参加者は各手法をそれぞれ 15 分程の時間で学習とタスクの実施を行った。



図 3.8: 実験参加者と実験の様子

3.5.4 観察結果

11 歳の実験参加者は 2 つの文字列選択手法(提案手法と既存手法)のどちらの操作も最初はエラーを繰り返したが、慣れるにつれ正確かつ迅速に操作するようになった。7, 9 歳の実験参加者は全体的にどちらの操作も時間が掛かったが、慣れるにつれキャレットのポインティングを正確に操作するようになった。

頻繁に確認されたエラーとして、既存手法では選択範囲の終点を移動する際に、誤って始点を触ってしまい移動不能となる Fat Finger Problem に起因されたと思われるケースがあった。これは 3 人中 3 人で発生し、学習時間に関係なく頻発していた。提案手法では学習初期に、選択の始点指定をしないまま選択モードに変更してしまい、やり直しとなるケースが多かったが、これは試行回数が増えるに従って発生しなくなった。7 歳の実験参加者は、どちらの手法でもスマートフォンを固定する手や、操作しようとしている指以外の部分が画面に当たることに起因する誤動作を頻発した。9 歳の実験参加者も時々同様の誤動作を起こした。

未習熟者にとってどちらの手法も初見では操作が困難であったが、短い学習時間を経ることでタスクを完了させられるようになった。また既存手法で起きた Fat Finger Problem に起因したと思われるエラーは提案手法では発生しなかった。この様に、提案手法は未習熟者でも短い学習時間で操作の習得ができ、また既存手法でよく見られた Fat Finger Problem によるエラーを回避する有用性も一部見られた。

実験後のインタビューにて、キャレットのポインティングに関しては既存手法の方が直接触るから早くてやりやすいという意見が出た。これは Fat Finger Problem の原因にもなり得るが、提案手法は既存手法を排他しないため、既存手法も同時に機能させることでユーザビリティを向上させられる可能性がある。

3.6 議論

本研究で実施した実験の結果は以下のようにまとめられる。

- 小型携帯端末を日常的に利用するユーザは、提案手法を初めて使用するにも関わらず、作業時間は標準的な操作手法とは有意な差が確認されなかった。ただし、フォントサイズが小さい場合には提案手法の方が有意に短い作業時間となる例もあった。
- 本実験の結果からも、既存手法の小フォントサイズ条件において作業速度が遅くなる Fat Finger Problem が観察されたが、提案手法では観察されなかった。
- SUS の結果から、ユーザビリティの観点からは提案手法は標準的な操作手法よりも有意に使いやすいインタフェースであると評価された。
- 小型携帯端末未習熟の児童の観察実験の結果から、どちらの操作手法についてもタッチ操作の問題点は観察された。一方で提案手法においてはタッチ操作に慣れているユーザと同じ様に利用できている様子が観察された。

上記の結果より、既存手法と比較した際の提案手法の有用性、有効性を示すことができたと考えている。一方で提案手法についての制限や将来課題は残っている。

まず、本研究で実施した実験は提案手法を iOS で実装されている文字列選択手法と比較しただけであり、他の OS、例えば Android 上で実装されている手法とは比較していない。今回の実験は概念実証には十分な実験であったと考えているが、他の環境で提供される手法との比較は必要であると考えている。次に、実験においては長い文章を選択する場合を想定しておらず、この場合の実験についても必要であると考えている。インタフェースの実装としては、例えば一本指でのドラッグ操作で選択、二本指でのドラッグ操作で画面スクロールを行い、長い文章の選択をすることは可能であるが、実用化に向けては事前検証

が必要であると考えている。また、システムの実装としてテキストエリア全体へ影響を与えることが考えられ、既存システムとの衝突を回避する検討が必要であると考えている。

最後に、学習に関するコストについて触れる。本研究で提案したインタフェースはこれまでの概念とは全く異なり、キャレットの移動および文字列の選択を文字列全体を背景として移動することで操作するというものであった。このため、学習コストが大きいのではないかという批判が考えられるが、比較実験においては1名を除いた全員がiPhoneを所有し日常的に使用しているユーザが参加し、これら実験参加者が初めて触るFix and Slideの操作速度の方が統計的に有意な差で早く、その他については統計的な差が確認されていない程度の速度であったことを考えると、学習コストは高くなく、ユーザビリティ評価の結果と合わせて考えると、十分に受け入れられやすいものであると考えられる(図 3.4, 図 3.6)。また、観察実験における小型携帯端末未習熟者の利用観察の結果からも、提案手法は既存手法と同様の学習コスト程度であることが示唆された。

3.7 むすび

本章では、スマートフォン用のメールサービスを題材に、メールやメッセージなどの文章作成が難しい問題に着目し、マイクロUX法を試行した。文章作成が難しい問題の要因を分解し、文字の挿入や選択する操作が難しい問題の改善を試みた。

小型携帯端末における文字列選択操作について、従来の直接対象を触って操作する直接操作手法ではなく、文字列全体を背景として移動することによるキャレット操作および文字列選択手法の提案を行った。現在標準的に利用されているインタフェースとの比較実験を行った結果、ユーザは提案手法を初めて使うにも関わらず、フォントサイズが小さい場合は提案手法の方が有意に短い作業時間となり、またその他の場合では標準のインタフェースとは有意な差が確認されなかった。さらに、ユーザビリティの観点では提案手法の方が高い評価を得た。

一方で本章で試行したマイクロUX法において、考案した手法をサービスに導入する際の実装上の懸念が残った。Fix and Slideはテキストエリア全体に影響を与えるため、サービスに導入する際には既存システムを検証し改修しなくてはならない可能性が高い。マイクロUX法で考案した手法は、実装の仕方によってはサービスのシステムに大きな影響与えることが考えられ、サービスへ導入する際の障壁になり得ることが示唆された。

第4章 地図サービスで片手操作が難しい問題 に対するマイクロUX法の試行

4.1 はじめに

本章では、スマートフォン用の地図サービスを題材に、片手で操作が難しい問題に着目し、マイクロUX法を試行する。片手操作が難しい問題を分解し、その中の一つである片手でズーム操作を行うのが難しい問題の改善を行う。改善の指標としてタスク完了時間と操作回数、SUSスコアを定義し、この指標に有効な手法の検討を行う。

4.1.1 UX課題の分解

スマートフォン用の地図サービスでは、地図上の現在地や目的地までの経路を把握し、目的地へスムーズに到達できることが重要な価値の一つである。自宅で翌日の集合場所周辺の情報を調べたり、電車の中で駅から目的地までの経路を調べたり、地図サービスは様々なシーンや場所で用いられる。しかし、これを片手で操作することは難しい。その要因として、スマートフォンが大画面化し画面の端に配置された検索窓やボタンなどに指が届かない問題や、片手でのズームイン・アウトを行うジェスチャ操作が難しい問題などが考えられる。

小型携帯端末においてズーム操作（ズーミング；Zooming）は主要な操作の1つである。ズーム操作は地図アプリの操作時や写真や動画等のメディアコンテンツ、文章等のドキュメントファイルの閲覧時に頻繁に行われる。これまで小型携帯端末における片手での操作インターフェースは多く研究されてきているが、片手でのズーミング手法については十分に検討されてこなかった。小型携帯端末での最も一般的なズーム操作はピンチ（指と指の間隔を変化させる）である。地図アプリの使用時には、2本の指をタッチスクリーン上で近づけたり遠ざけたりし、地図が適切な縮尺となるよう操作を行う。この操作は片手で携帯端末を保持し、反対側の手でピンチ操作を行うというように両手で行う必要があり、片手では小型携帯端末を落下させる危険性が高まる。他の一般的なズーム操作としてダブルタップ（画面を素早く2度触る）があるが、ダブルタップは多くの場合ズームインのみしか行えない。またズームレベルは事前に決められておりユーザは自由に制御することができず、求める縮尺への調整には他の操作手法も用いる必要がある。小型携帯端末版 Google

マップは特殊なジェスチャを用いた片手ズーム操作機能を備えている。ユーザが画面をダブルタップし2度目のタップ時に指を画面から離さないまま上下に移動させることで、上ならズームアウト、下ならズームインを実行させることができる。しかし一連の指の動きとズームイン・アウトの関係が直感的でなく、ズームインとズームアウトを間違えることも少なくない。

本章ではこの問題に着目し、地図サービスが片手操作が難しい問題に対し、ズーム操作手法の改善を行うこととした。ここでは Google マップのズーミング手法をベースラインとし、我々の提案する手法との比較を行う。

4.1.2 改善指標の定義

片手でのズーム操作を検討するに当たり、定量的な改善指標の検討を行った。地図サービスにおいてズームイン・アウトは多用される操作の一つである。しかし、ピンチイン・ピンチアウトは両手で行われることを前提としたジェスチャ操作であり、片手で行おうとするとスマートフォンの保持が不安定になり操作が困難になる。すなわち、ズーム操作に対する十分な片手操作手法が提供されておらず、片手が塞がった状況などでは著しい操作性の低下が想定される。

そこで、複雑な操作を必要としない片手でのズーム操作が実現できれば、地図サービスで片手操作が難しい問題の解消に貢献すると考えた。片手でのズーム操作には先行研究や既存手法が存在するが、操作性や直感性の面で改善の余地がある。操作の効率性を測る指標として、より速く、より少なく、より使いやすく目的が達成できれば良いと考えた。これらはタスク完了時間と操作回数、SUS スコアによって定量的に計測が可能であり、これを改善指標と定義することとした。

4.1.3 改善手法の検討

本章では、地図サービスで片手操作が難しい問題に対し、片手でのズームアウト手法を提案する。提案手法の特徴はユーザが画面に対し押下した際の圧力値を計測するアップル社の圧力感知マルチタッチ技術、Force Touch を用いる点である。我々はこの技術を用いた Bounce Back と Force Zoom と呼ぶ2つのズームアウト手法を設計した。Bounce Back はユーザが画面を押した際の圧力を計測する。そしてユーザが指を画面から離すと、その圧力に応じたズームレベルへズームアウトする。Force Zoom も同様に圧力を計測し用いるが、ユーザが画面を押し続けている間の圧力に応じて連続的にズームアウトする。これらの手法はユーザが画面を押し力を変化させることでズームレベルを調整することが出来る(図 4.1)。本研究では、我々が提案する手法と現在普及している手法および感圧式ズー

ミング手法に関する先行研究での手法 [28] とを比較し、操作の効率性とユーザビリティについて調査する。

本研究の主な貢献は以下となる。

- 小型携帯端末上で動作する感圧式ズームアウト手法の提案と実証
- 小型携帯端末での片手操作の 4 つのズームアウト手法, Bounce Back, Force Zoom, 小型携帯端末版 Google マップの手法, そして先行研究手法の GraspZoom[28] を比較する実験の実施
- 提案手法が先行研究 GraspZoom[28] よりも作業実行時間が短く, Google マップでの操作手法よりも操作回数が少ないことを確認

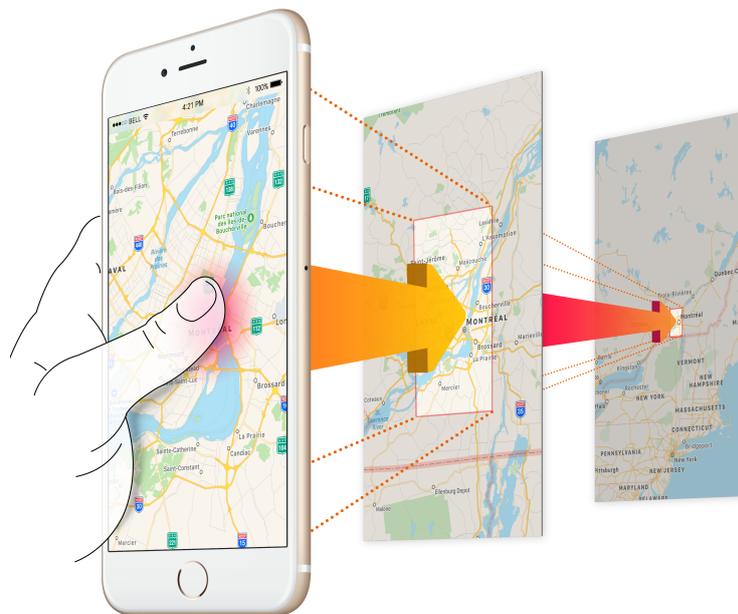


図 4.1: Force Zoom : 指が画面を押す力を計測し, 押す力の強さに比例した速度でズームアウトする。

4.2 関連研究

小型携帯端末に搭載された圧力感知タッチスクリーンに関するインタフェースはヒューマンコンピュータインタラクション (HCI) に関する研究分野において数多く研究されてきた [29]. Miyaki と Rekimoto は小型携帯端末向けの圧力を用いたズームング手法であ

る Grasp Zoom[28] を提案した。これは我々の手法と類似する点があるが、彼らはユーザインタラクションの理解よりもハードウェアとソフトウェアの構築に焦点を当てており、ユーザビリティ評価を行っていない。Spelmezan らは小型携帯端末の片側面に圧力センサを拡張した SidePress という手法を考案し、圧力センサの位置や力量とスクロールやテキスト選択といった基本的なアクションの関係について調査した [30]。他にも力の値を利用した手法がある。ForceTap は Z 軸の加速度を計測しタップの強さを計算した [31]。Force Gesture は小型携帯端末の画面との接線力 (Tangential forces) を利用した [32]。Harrison は接線力を利用した指の上下移動によるズーム操作手法を提案した [33]。Clarkson らは小型携帯端末での圧力を用いた入力手法の有用性について調査し [34]、Stewart らは小型携帯端末での圧力を用いた入力手法の特徴を解明するための一連の調査を行った [35]。近年では、小型携帯端末に組み込まれた圧力計を用いた圧力感知手法が登場してきている [36]。ForceEdge は小型携帯端末画面上の特定領域で有効となる圧力を用いたスクロール手法を提案し [37]、Force Picker は小型携帯端末のシステムが提供するピッカーユーザインタフェースの操作に圧力を用いた手法を提案した [38]。これらの研究からも力を用いたインタフェースは小型携帯端末のインタフェース開発における主要な研究分野の 1 つと考えられる。我々の研究は、小型携帯端末に標準搭載された圧力感知タッチスクリーンを用いたズーム手法を提案し、このユーザビリティを検証するものである。

本研究では特に片手での小型携帯端末操作に注目している。我々が対象とする操作はズーム操作とナビゲーション操作である。Dong らは小型携帯端末を片手で操作する地図ナビゲーション手法について調査した [39]。TapTap and MagStick はユーザに片手で対象を指定できるようにした [40]。Rekimoto は傾きを用いたナビゲーション手法を提案した [41]。Sensor synaesthesia は小型携帯端末の傾きを用いたズーム手法を提示した [42]。Holman は小型携帯端末の側面にタッチセンサを付与し指での入力を補助する手法を提案した [43]。半自動的なズーム操作は高度なユーザインタフェースになりえることが示唆されている [44]。これらのことから小型携帯端末の片手操作に関する研究は小型携帯端末のインタフェース開発において活発な研究領域と考えられる [45][46]。本研究では、近年小型携帯端末で利用可能となった画面の圧力情報を利用した片手での操作を実現するインタフェースを提案する。Fat Thumb は指とタッチスクリーンとの接地面積から算出した模擬圧力を用いたズーム手法を提示したが [47]、我々の研究は、小型携帯端末に標準搭載された圧力感知タッチスクリーンを用いたズームを提案し、このユーザビリティを検証するものである。2019 年現在で国内普及率 43.08% の iPhone シリーズのうち、圧力感知タッチスクリーンを搭載するモデルは約 50% となっており [48][49]、提案する手法および実施する実験とその結果得られる知見は当該領域の進展に貢献すると考えている。

Bounce Back



Force Zoom

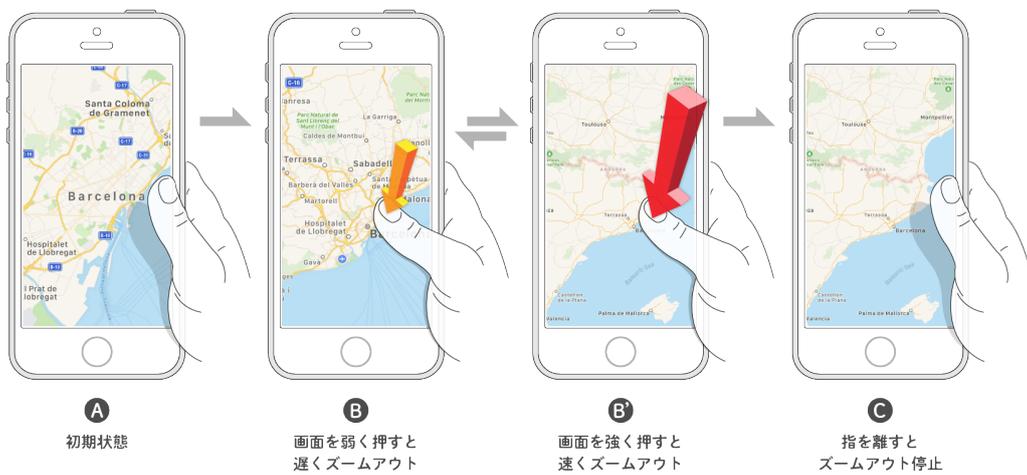


図 4.2: Bounce Back (上段) と Force Zoom (下段) の概要. 画面を押す圧力を用いズームアウト操作を行う.

4.3 提案手法：Bounce Back と Force Zoom

ここで我々の提案する2つの手法を簡単に説明する。我々の提案する新たなズームング手法の特徴は、タッチスクリーンが感知する圧力の値を用いズームアウトを行う点である。ユーザはメディアコンテンツや地図を閲覧する際にズームインも必要とするが、我々が提案する Bounce Back と Force Zoom はダブルタップの様な従来のズームイン手法との併用を想定している。圧力感知タッチスクリーンを「押し込む」という単一のアクションでズームインとズームアウトといった異なる操作を行うのではなく、ズームインとズームアウトを別々のアクションとすることで誤操作の軽減やユーザビリティの向上に貢献できるのではないかと考え、本研究ではズームアウトのみを対象としている。

4.3.1 Bounce Back

Bounce Back はバネの特徴から着想している。バネは押し縮められると、その縮められた長さに応じて反発する力が働く。机の上にあるバネを指で押し、その指を離すとバネは机から高速に飛び跳ねる。我々はこれをズームアウト操作のメタファーとして用いた。ユーザが小型携帯端末の画面を押し（親指での操作を想定しているが限定されない）、指を離すと机からバネが飛び跳ねる様に表示領域がズームアウトする（図 4.2 上段）。この時、指が押した力に応じ飛び跳ねる高さ、すなわち到達するズームアウト率が変化する。この指で押した力はユーザが押下している指先のエリアが凹み陰影がつくことで可視化され、操作後のズーム率の目安として用いることが出来る。つまり、ユーザは指先の表示の変化を見ることで力の調整が可能となる。

4.3.2 Force Zoom

Force Zoom は指で小型携帯端末の画面を押す力の値をズーム率として用いる。これを利用して例えば、画面を押すとズームインするといった使い方があるが、一般的な小型携帯端末ではズームイン操作を行うためにダブルタップやロングタップ（画面をタップしたまま指を離さない）といった様々な手法が用意されており、これらは既に広く普及している。対称的にズームアウト操作を行う手法はピンチアウト程度しか用意されていない（例えば小型携帯端末版の Google マップアプリではスクロールバーやズームイン/アウトのためのボタンは廃止されてしまった）。そこで我々は指で押す力をズームアウトに用いるようインタフェースを考案した（図 4.2 下段）。現在表示している領域よりも広い地図領域がある場合、ズーム率を調整するために圧力の値を用いる（図 4.1）。ユーザが画面を素早く押し込んだ際にはズーム率も素早く変化し、ユーザが画面をゆっくり押し込んだ際に

はズーム率もゆっくり変化する。ユーザが指を離すと、ズーム率の変化も止まる。

4.4 既存手法との比較実験

我々が提案する Bounce Back と Force Zoom の2つの手法の有効性を検証するために、センサによって計測された圧力を用いたズーミング手法として提案手法に最も関連がある先行研究の手法 (Grasp Zoom[28]) と現在広く普及している小型携帯端末版 Google マップの標準ズーミングユーザインタフェース (Default ZUI) の手法との比較実験を実施した。ここではマルチスケールの対象選択をタスクとした。すなわち、実験参加者は4つの手法をそれぞれ用い画面上の表示物を選択するよう指示される。実験ではこれら各手法での作業実行時間 (パフォーマンス) と操作回数を計測し、ユーザビリティに関するフィードバックを得ることを目的とした。そして力を用いたズーミング手法に関する知見を深めることを試みた。

4.4.1 実験で使用する小型携帯端末

全ての手法は iOS バージョン 10.3.3 の iPhone 7 上で実装した。ディスプレイは解像度 $1,344 \times 750$ で 4.7 インチだった。実験参加者はスマートフォンを利き手で持つよう指示され、もう片方の手は机の上や自身の膝などに置き操作を行わないよう指示された。

4.4.2 タスク：マルチスケールの対象選択

本研究で実施する実験では、マルチスケールの対象選択タスクを採用した [50]。実験参加者は広大な領域内のいずれかにある 120px 四方の正方形オブジェクトを 10 個選択する。このオブジェクトはそれぞれ開始位置となる画面中心から 10,176px 離れている。これは以下の式で算出されている。

$$\left(\frac{1(\text{default_zooming_level})}{0.15(\text{target_zooming_level})} \right) \sqrt{800^2 + 1300^2}$$

800 (ピクセル) は幅 (X 軸) で 1300 (ピクセル) は高さ (Y 軸) である。これは選択対象となるオブジェクトがタスク開始時の画面領域内には表示されない位置であり、画面表示領域を上下左右に倍にした範囲内に選択対象オブジェクトが存在する距離となる。すなわち、実験参加者が必ず探索が必要となる範囲で、かつ過度に探し迷うことがない範囲とした。また今回の実験では2つのセッションを実施するため、解釈が複雑にならないよう変数を固定化、すなわち選択対象となるオブジェクトの距離は一定とした。標準のズーム率は 1 で、対象物のズーム率は 0.15 とし、この値は Z 軸となる。今回実験で用いる各手

法は連続的にズーム率が変化するため、画一のズーム変化率を持たない。

4.4.3 セッション：ズームアウト単独とズームアウト/イン複合

本研究で実施する実験では、インタフェースのパフォーマンスとユーザビリティを比較するために2つのセッションを用意した。すなわち、(1) ズームアウト単独のセッション、(2) ズームアウトとズームイン複合のセッションである。(1)のセッションのタスクは主にズームアウト操作で完了可能となっている。タスク開始時の画面ズーム率は最大値であり、選択対象となる正方形は画面外に存在するため見えない。実験参加者はまず4つのいずれかの手法を用いてズームアウト操作し対象物を見つける必要がある。対象物は実験参加者が最大限ズームアウトする必要がある十分に離れた場所に配置された。本セッションは広大な領域内にある選択対象オブジェクトをいかに速く視認できるか、すなわちいかに速くズームアウトし広範囲を見渡せられるかを計ることが目的であり、ズームレベルのオーバーシュート回避行動による影響は考慮しなかった。1つのタスクは、いずれかにある対象物の四隅を画面中央部に表示されているズーム率に影響されない固定された40ピクセル四方の枠線上に乗せることで完了する。実験参加者は対象物を適切なサイズとなるようズームアウト操作とズームイン操作を行う必要があるが、ズーム率が最小の時に対象物のサイズが枠線と一致するよう実装している。実験参加者が1つのタスクを完了すると表示されていた対象物は消え、新たな対象物がいずれかに設置され、画面はズームレベルと開始位置を初期化され次のタスクが開始する。このタスクは地図アプリケーションで目的地検索をする際にユーザが必要とするズーム操作を参考に行っている。

次に(2)のセッションのタスクはズームアウトとズームインの両操作を用いることで完了可能となっている。実験参加者はまずズームアウトし対象物の位置を探し、続いてズームインし40ピクセルの枠線上に乗るよう調整を行うよう求められる。(1)と同様に、タスク開始時のズーム率は最大値であり対象物は見えない。実験参加者はまず4つのいずれかの手法を用いてズームアウトし対象物を見つける必要がある。対象物は最大限ズームアウトする必要がある十分に離れている場所に配置されており、対象物を発見したあとは最大限ズームインする必要がある小ささで表示される。すなわち、実験参加者は対象物を枠線上に乗せるためにズームアウト操作とズームイン操作を行う。実験参加者が1つのタスクを完了すると表示されていた対象物は消え、新たな対象物がいずれかに設置され、画面はズームレベルと開始位置を初期化され次のタスクが開始する。

GraspZoom



図 4.3: GraspZoom の概要.

4.4.4 実験条件：ユーザインタフェース

実験では Bounce Back, Force Zoom, GraspZoom[28](図 4.3), そしてベースラインとして Default ZUI のユーザインタフェースを条件に設定した. ここではベースラインとしてボタンを用いたユーザインタフェースを含まなかった. プラスマークを押すとズームインしマイナスマークを押すとズームアウトするボタンを用いたユーザインタフェースは主要な小型携帯端末版のマップアプリケーションでサポートされていない. 例えば, ボタンを用いたユーザインタフェースは 2012 年の Android アプリケーション版 Google マップ (バージョン 6.7) から非搭載となった. また, iOS 標準搭載の地図アプリケーションも 2010 年 (iOS 4 の頃) からプラスとマイナスボタンは非搭載となった. こうしたことから, 我々は小型携帯端末におけるボタンを用いたユーザインタフェースは考慮から外し, 本実験におけるベースライン条件に含まなかった.

4.4.5 実験参加者と手続き

24 名のボランティアが実験に参加した. 男性は 8 名, 女性は 16 名であり, 年齢は 22 歳から 35 歳であった (mean=28.88, S.D.=4.3). 全ての参加者はスマートフォンを所持しており, 7 名が Android 端末 (29.2%), 23 名が iPhone (95.8%) であった. すべての実験参加者はスマートフォン上でズーム操作を行ったことがあり, 6 名は Default ZUI の手法を使ったことがあった. すなわち, 他の 18 名は Default ZUI (小型携帯端末版 Google マップの手法) を知らなかった. 実験参加者は地図や写真, 動画, ウェブ閲覧時にズーム操作をよく用いていた. 実験参加者のズーム操作利用状況を調べたところ, ピンチジェス

チャはズーム操作としてよく用いられおり、ズームアウト操作は片手ではあまり行われていないことがわかった。

実験の手続きについて述べる。実験参加者は Bounce Back と Force Zoom, GraspZoom[28], Default ZUI の 4 つの異なるズームング手法をそれぞれ 10 回ずつ使用する。取り掛かる 4 つの手法については、実験参加者が経験する順番はラテン方格法によりバランスを取った。すべての試行で 10 回の対象選択タスクを課し、それぞれの実験参加者は全てのタスクを完了した。ここでは常にズームアウト単独のセッションを最初に行い、続いてズームアウトとズームイン複合のセッションを行うこととした。最終的に、実験参加者は 4 つの異なる手法を用い 80 個の対象物の選択を完了した。選択する対象物の位置はランダムではなく、統制をとるためにあらかじめ決められており、被験者間で同じ対象物を選択することとなる。それぞれの実験を行う前には、操作に慣れるためのトレーニングの時間を設けた。

4.4.6 実験用インタフェースの実装

ここでは簡単にそれぞれのズームング手法の実装とそのインタラクションについて述べる。本研究では片手での操作手法に注目しており、全ての手法でピンチジェスチャは実装せず、対象物を枠線上へ移動させられるようスワイプ操作によるナビゲーション機能は実装した。ダブルタップ操作によるズームイン機能は GraspZoom を除く全ての手法で実装し、力を用いたズームイン操作は GraspZoom で有効化した。また、小型携帯端末版 Google マップに搭載されているダブルタップとスライドの組み合わせによるズーム操作は Default ZUI で有効化した。Default ZUI は 2 つのズームイン手法を持つが、これは使い慣れた操作が制限されない状況にて提案手法との比較を行うためである。

Bounce Back と Force Zoom

ここでは簡単に Default ZUI との違いについて述べる。Bounce Back と Force Zoom はダブルタップ操作によるズームイン機能以外の Default ZUI で実装されている手法を実装していない。

GraspZoom (Miyaki and Rekimoto, MobileHCI' 09[28])

この手法は GraspZoom のインタフェースを模倣している。我々は GraspZoom の手法を実装するために、GraspZoom で用いられていた拡張ハードウェアの代用として、iPhone に搭載されている圧力感知センサを用いた。この手法ではズーム率の変化に圧力の値を用いており、強い押し込みがされるとズームインもしくはズームアウト操作が実行された。

このズーム手法は Force Zoom と同様であるが、GraspZoom ではユーザが拡大か縮小かを画面上をスライドすることによって変更する。すなわち、GraspZoom はズームインとズームアウト操作の両方を圧力で行うことが可能となっている。

Default ZUI：ダブルタップとスライドの組み合わせ

この手法は小型携帯端末版 Google マップのズーム手法を用いている。この手法では、ユーザが画面をダブルタップし2度目のタップの際に指を離さず、そのまま上下にスライドすることでズームする仕組みとなっている。本実験では Google マップでの実装と同様にユーザが指を上をスライドしたらズームアウト、下をスライドしたらズームインとした。

4.4.7 実験結果：ズームアウト単独セッション

作業実行時間

4つの手法の平均作業実行時間（秒）は、10個の対象選択作業を1試行とした結果を集計したものである（表 4.1 Out 行, 図 4.4）。

表 4.1: 作業実行時間の結果と標準誤差。小さい値ほど良い。

	Bounce Back		Force Zoom		Grasp Zoom		Grasp Zoom	
	<i>mean</i>	<i>s.d.</i>	<i>mean</i>	<i>s.d.</i>	<i>mean</i>	<i>s.d.</i>	<i>mean</i>	<i>s.d.</i>
Out	8.65	2.70	7.35	3.02	11.35	3.49	7.16	2.04
Out/In	10.50	3.95	9.12	1.46	16.53	4.57	10.90	4.13

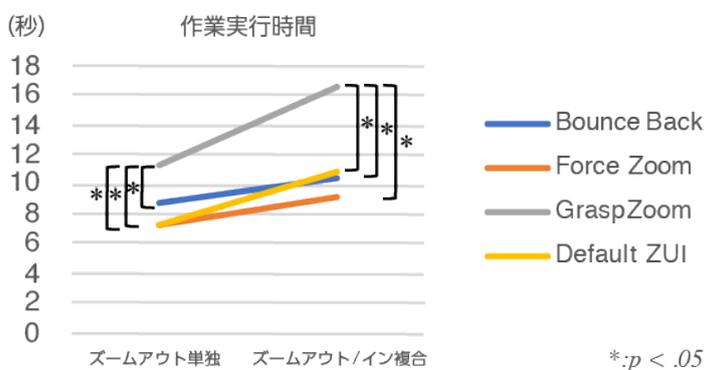


図 4.4: 作業実行時間の結果

はじめに、平均時間をバートレット検定にかけた結果、分散の非均一性は確認されなかった ($X^2(3) = 6.47, p = .09$)。続いて一元配置分散分析を行った (被験者内計画; 独立変数として、ズームアウト手法の種類。従属変数として、作業実行時間)。この結果、有意な差が確認された [$F(3,92) = 10.51, p < .01, \eta^2 = 0.26$]。このため、チューキーの多重比較検定を実施したところ GraspZoom は Bounce Back ($p < .01$)、Force Zoom ($p < .01$)、Default ZUI ($p < .01$) より有意に遅いことが確認された。

操作回数

4つズームアウト手法の平均操作回数を (表 4.2 Out 行) と (図 4.5) に示す。

表 4.2: 操作回数の結果と標準誤差。小さい値ほど良い。

	Bounce Back		Force Zoom		Grasp Zoom		Grasp Zoom	
	mean	s.d.	mean	s.d.	mean	s.d.	mean	s.d.
Out	23.79	8.95	16.75	10.56	34.96	8.78	30.42	7.62
Out/In	40.21	8.67	34.13	5.24	58.38	12.91	53.25	22.77

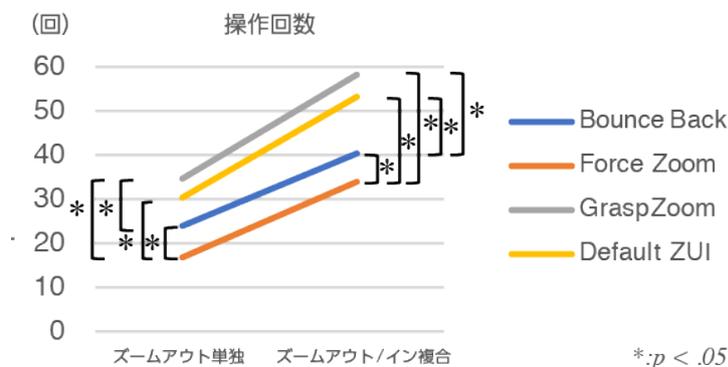


図 4.5: 操作回数の結果

4種類の操作アクションの回数を、実験用アプリケーション内で自動計測し収集した。すなわち、(1) Bounce Back とダブルタップの回数、(2) Force Zoom とダブルタップの回数、(3) Default ZUI とダブルタップの回数、(4) GraspZoom の操作回数となる。GraspZoom の操作回数にはスワイプによるズームモードの変換操作の回数は含まれていない。全ての手法でズームインもしくはズームアウトが開始され終了した時点で操作が一回行われたとした。連続的な入力 (加圧, スライド) は指が画面から離された時点で操作が一回行われたとした。これらをバートレット検定にかけた結果、分散の非均一性は確認されな

かった ($X^2(3) = 2.45, p = .48$). 続いて一元配置分散分析を行った (被験者内計画; 独立変数として, ズームアウト手法の種類, 従属変数として, 操作回数). その結果, 有意な差が確認された [$F(3,92) = 17.76, p < .01, \eta^2 = 0.37$]. このため, テューキーの多重比較検定を実施したところ Force Zoom での操作回数は Default ZUI ($p < .01$), Bounce Back ($p < .05$), GraspZoom ($p < .01$) より有意に少ないことが確認された. 併せて Bounce Back は GraspZoom ($p < .01$) より有意に少ないことが確認された.

ユーザビリティ評価

各手法での試行が完了した後に実験参加者に回答を依頼した 10 個の設問からなるアンケートの分析を行う. ここでは 5 段階のリッカート尺度のアンケートである System Usability Scale (SUS) を用いた [51]. SUS スコアを (表 4.3 Out 行) と (図 4.6) に示す.

表 4.3: SUS の結果と標準誤差. 大きい値ほど良い.

	Bounce Back		Force Zoom		Grasp Zoom		Grasp Zoom	
	<i>mean</i>	<i>s.d.</i>	<i>mean</i>	<i>s.d.</i>	<i>mean</i>	<i>s.d.</i>	<i>mean</i>	<i>s.d.</i>
Out	55.10	19.73	69.69	18.26	36.46	19.66	62.50	19.66
Out/In	55.42	18.34	69.48	15.34	28.75	15.61	70.21	17.12

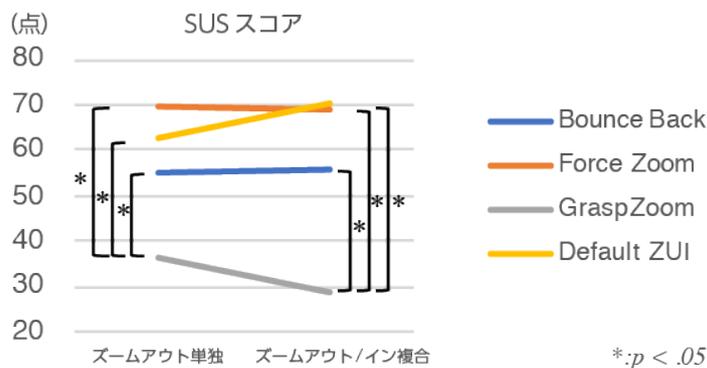


図 4.6: SUS の結果

これらの SUS スコアはフリードマン検定を用い分析された (被験者内計画; 独立変数として, ズームアウト手法の種類, 従属変数として, SUS スコア). その結果, 有意な差が確認された [$X^2(3) = 34.41, p = .01$]. このため多重比較としてボンフェローニ補正を行ったウィルコクソンの順位和検定をかけた結果, Force Zoom の SUS スコアは Bounce Back ($p < .01, r = 0.488$), GraspZoom ($p < .01, r = 0.627$) より有意に高いことが確認された.

Force Zoom の SUS スコア 69.7 点と Default ZUI の SUS スコア 62.5 点は評価基準によると Good に該当する [39]. Bounce Back は平均より低い OK に該当する 55.1 点であった. GraspZoom は 38.0 点で Poor に該当した.

個別の質問項目 (図 4.7) では, 全体的に GraspZoom は他手法より有意に低いスコアであった. Force Zoom と Default ZUI は各設問において近いスコアとなった. Bounce Back は全体を通して 3 番目のスコアであった. より詳しく見ると, Bounce Back と Force Zoom, Default ZUI は小型携帯端末でのインタラクションとしてシンプルである (Q2), 容易に扱える (Q3), よく統合されたインタフェースである (Q5), 自信を持って操作できる (Q9) と評価された.

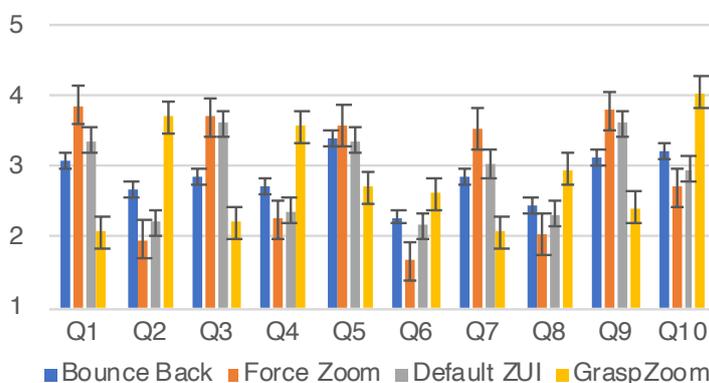


図 4.7: ズームアウト単独セッションにおける SUS の個別の質問項目の結果

4.4.8 実験結果：ズームアウトとズームイン複合セッション

作業実行時間

ズームアウト単独セッションと同様に 4 つの手法の平均作業実行時間 (秒) は, 10 個の対象選択作業を 1 試行とした結果を集計したものである (表 4.1 Out/In 行, 図 4.4). はじめに, 平均時間をバートレット検定にかけた結果, 分散の非均一性が確認され ($X^2(3) = 26.14, p = .01$), 平均時間をフリードマン検定で分析した (被験者内計画; 独立変数として, ズームアウト手法の種類, 従属変数として, 作業実行時間). その結果, 有意な差が確認された [$X^2(3) = 30.95, p = .01$]. ボンフェローニ補正を行ったウィルコクソンの順位和検定をかけた結果, GraspZoom は Bounce Back ($p < .01, r = 0.89$), Force Zoom ($p < .01, r = 1.0$), Default ZUI ($p < .01, r = 0.72$) より有意に遅いことが確認された.

操作回数

4つズームアウト手法の平均操作回数を(表 4.2 Out/In 行)と(図 4.5)に示す。これらをバートレット検定にかけた結果, 分散の非均一性が確認され($X^2(3) = 48.17, p=.01$), 平均操作回数をフリードマン検定で分析した(被験者内計画; 独立変数として, ズームアウト手法の種類, 従属変数として, 操作回数)。その結果, 有意な差が確認された [$X^2(3) = 41.53, p=.01$]。ボンフェローニ補正を行ったウィルコクソンの順位和検定をかけた結果, Force Zoom の操作回数は Default ZUI ($p < .01, r=0.83$), Bounce Back ($p < .05, r=0.58$), GraspZoom ($p < .01, r=0.88$) より有意に少ないことが確認された。Bounce Back の操作回数は有意に Default ZUI ($p < .05, r=0.62$), GraspZoom ($p < .05, r=0.79$) より少ないことが確認された。

ユーザビリティ評価

このセッションにおける SUS スコアを(表 4.3 Out/In 行)に示す。これらの SUS スコアはフリードマン検定を用い分析された(被験者内計画; 独立変数として, ズームアウト手法の種類, 従属変数として, SUS スコア)。その結果, 有意な差が確認された [$X^2(3) = 37.945, p=.01$]。ボンフェローニ補正を行ったウィルコクソンの順位和検定をかけた結果, Force Zoom の SUS スコアは Bounce Back ($p < .01, r=0.47$), GraspZoom ($p < .01, r=0.657$) より有意に高いことが確認された。GraspZoom の SUS スコアは Bounce Back ($p < .05, r=0.611$), Default ZUI ($p < .01, r=0.635$) より有意に低いことが確認された。平均 SUS 質問項目を(図 4.8)に示す。全体的な傾向として, ズームアウト単独セッションと近いスコアとなった。

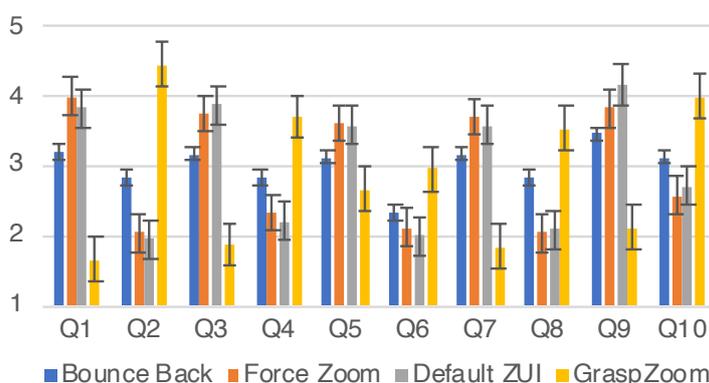


図 4.8: ズームアウトとズームイン複合セッションにおける SUS の個別の質問項目の結果

4.5 議論

本実験全体を通して実験参加者は Bounce Back と Force Zoom を正常に扱えた。本研究で実施した実験の結果は以下のようにまとめられる。

- すべての実験参加者は Bounce Back と Force Zoom を初めて使うのにもかかわらず、パフォーマンスは Default ZUI と有意な差が確認されなかった。対照的に、GraspZoom がその他3つの手法よりパフォーマンスが有意に低いことが確認された。
- 操作回数は Bounce Back と Force Zoom が Default ZUI と GraspZoom より有意に少ないことが確認された。
- GraspZoom のユーザビリティは Force Zoom, Bounce Back, Default ZUI より有意に低く評価されたことが確認された。実験参加者は Force Zoom が容易に扱え、自信を持って使えると感じた。

我々は実験後に簡単なインタビューを行った。その結果、実験参加者は GraspZoom に良くない印象を持ったことが分かった。インタビューで挙げられた主な原因として「現在のズームモードが拡大縮小どちらなのか分からなかった」、「現在のズームモードを知るには、一度画面を押してズームさせてみるしかなかった」といったものがあった。簡単な実験時の行動観察では、少なくとも人数の実験参加者が GraspZoom のズームモードの変換に失敗していた。これは変換時に視覚的なフィードバックがないことに起因していた。同様に、圧力の値をズームアウトとズームイン両方に用いたことで、実験参加者はズームモードの変換を行った後に現在のズームモードがどちらか把握できなくなっていた。このことからズームインとズームアウトを変換するための操作には多くの問題があり、モード変換を要するインタラクションはユーザを混乱させやすいと我々は考える。また簡単な実験時の行動観察から、実験参加者は Default ZUI においてダブルタップとスライドを組み合わせた操作でのズームインとズームアウトを行えることより、ダブルタップ単体でのズームインを好む傾向があり、Force Zoom においては力を加えズームインしようと誤って操作した実験参加者はいなかった。すなわち力を用いたズーム手法において、GraspZoom や Default ZUI のように同一のアクションによるズームインとズームアウト操作はユーザに混乱を招きやすく、タイプの異なる手法でのズームインとズームアウト操作は誤操作の軽減とユーザビリティが向上することが示唆されたと考える。タイプの異なる手法である場合でも、力を加える操作がズームインとズームアウトのどちらか迷うことは考えられるが、これはジェスチャアクション全般に起こりえる学習性に関する問題であり提案手法に特徴的なインタフェース上の問題ではないと考える。

Force Zoom は好意的に受け止められた。「Force Zoom は指で繰り返し操作する必要がなかった」、「Default ZUI は指をたくさん動かす必要があるが、Force Zoom はそれがなかった」との意見があった。操作回数の実験結果が示すように、Force Zoom はズームアウト操作を少ない操作回数で行える。数人の女性実験参加者からは「自分の手は小さく、指で何度も操作する必要があった」、「指が疲れてしまった」との意見が出た。興味深い意見として、「自分は入浴中にスマートフォンを使うが、非常に慎重に持たなくてはならない。この時片方の手は濡れていることが多く、スマートフォンを持っている手しか操作に使えない問題がある。スマートフォンが浴槽へ落下する危険性があり、指をスマートフォンから離すことができない。」との意見があった。Default ZUI の操作には、指を浮かし大きく移動する動作が何度も必要になる。対症的に、Force Zoom は少ない指の動き（ズームインにダブルタップ、ズームアウトに画面押し込み）で操作が完了する。また他の実験参加者から「ソファで寝ながらスマートフォンを操作する際にしばしば手から滑り落ち自分の顔にぶつける。だから Force Zoom を使いたいと思った。」との意見を出した。Bounce Back はいくつか「ズーム変化率を力で微調整するのは難しかった」との意見が出た。Bounce Back は押し込むと対象が凹むという現実のメタファを用いたため跳ね上がることを想起しやすかったとの意見もあったが、加えた力に対してどの位ズームアウトするかは指を離さないと分からず、望んだ比率とならない場合も多かった。これに対し Force Zoom は加えた力に即座に連続的に反応するため、ズームアウト処理中に微調整が行えた。これらが操作回数とユーザビリティ評価に影響したと考えられ、実験結果の通り Force Zoom が Bounce Back より好意的に受け止められた要因ではないかと考える。

4.5.1 提案手法の制限と本研究で得られた知見の範囲

Bounce Back と Force Zoom は概念実証段階のプロトタイプであり、ユーザインタフェース設計と実験結果には制限がある。本研究で用いたユーザインタフェースは実験的に試作したものであり、将来的な実用化における OS 等との統合のためにはさらなる設計上の課題があることが考えられる。例えば、既存の小型携帯端末版地図アプリケーションでは長押しで地図上にピンを立てる等のサービス独自の様々なジェスチャアクションが実装されており、それらとのコンフリクトを回避する設計が必要となる。また、本実験では選択対象のオブジェクトの配置距離が固定されていない場合、最大ズームレベルが設定されていない場合のパフォーマンスは明らかにしていない。実験結果に関しては、我々はピンチジェスチャといった両手を用いたズーミング手法との比較を行っていない。今後の研究として片手と両手でのパフォーマンスの比較を行う必要があると考えているが、一方で我々が提案した手法は実験で用いたアプリケーション同様に、1つのアプリケーション内で標

準的なズームユーザインタフェースと共存させられると考えている。本研究では小型携帯端末におけるメディアコンテンツ閲覧手法の可能性を提示したが、SUSの結果からも、Force Zoomは標準的な操作インタフェースによく統合されていると考えられ、また簡単なインタビューと行動観察からForce Zoomはダブルタップ操作によるズームイン機能とよく統合されていることが示唆された。

4.6 むすび

本章では、スマートフォン用の地図サービスを題材に、地図サービスで片手操作を行うことが難しい問題に着目し、マイクロUX法を試行した。片手操作が難しい問題の要因を分解し、片手でのズーム操作が難しい問題の改善を試みた。

本章では圧力を感知するマルチタッチ技術をズームアウト機能に用いたBounce BackとForce Zoomと呼ぶ手法を提案した。これらの手法は小型携帯端末の片手操作を支援することを目的として設計された。我々は提案手法の効率性とユーザビリティを調査するため、我々の提案する手法と先行研究で提案された手法、ベースラインとして小型携帯端末版Googleマップとを比較する実験を行った。実験の結果、Force Zoomと小型携帯端末版Googleマップの手法は実験参加者から近いSUSスコアを獲得した。同様に、パフォーマンス指標としての作業実行時間は、Bounce BackとForce Zoom、小型携帯端末版Googleマップの手法とで近い数値となった。Bounce BackとForce Zoomの操作回数は小型携帯端末版Googleマップの手法と先行研究であるGraspZoomの手法より有意に少なかった。SUSの結果はBounce BackとForce Zoomは単純かつ簡単なインタフェースであり、ダブルタップによるズームイン機能とよく統合されていることが示唆された。

一方でマイクロUX法の課題も残った。本章では圧力感知マルチタッチ技術を用いスマートフォンのディスプレイを押し力に応じてズーム操作を実行するBounce BackとForce Zoomという新しいインタフェースを実現した。しかし、実在のスマートフォン用地図サービスの多くには様々なジェスチャ操作が実装されており、地図を長押しし目的地のピンを立てるなどの既存ジェスチャとの衝突が懸念される。これを解決するためには、ズーム機能を実行する圧力の閾値を上げ、弱い力のときは長押しと判定するといった回避策が考えられるが、操作性が低下する恐れがある。もしくはサービス側の機能とジェスチャ設計を見直し衝突を回避する方法もありえるが、インタラクションデザインの全体的な見直しが必要になる。すなわち、マイクロUX法で考案した手法はシステムだけでなくデザイン面でもサービス側に影響を与える可能性があり、サービスへの導入コストを最小化する成果物の設計の重要性が示唆された。

第5章 地図サービスで屋内目的地への移動が難しい問題に対するマイクロUX法の試行

5.1 はじめに

本章では、3章と4章で試行したマイクロUX法に関する考察を振り返り、マイクロUX法の成果物の適切な形について検討を行う。3章で考案したFix and Slideは既存手法より有意に速いキャレット操作を実現したが、文字列全体をキャレットの位置に移動するアプローチであったため、サービス導入時には画面構成や実装構造など既存システムの大規模修正が必要になる可能性が高かった。4章で考案したBounce BackとForce Zoomは既存手法より有意に少ない回数でズームアウト操作を実現したが、サービスに実装されているスライドや長押しなどのジェスチャ機能と衝突する恐れがあり、サービス導入時には既存の機能やインタラクションデザインの全体的な見直しが必要になる可能性が高かった。マイクロUX法は、アジャイル開発の開発サイクルに並走可能なUX改善のアプローチであり、UX改善の実施コストが小さくても導入コストが大きくては本末転倒である。すなわち、マイクロUX法の成果物はサービスへの導入コストを限りなく小さくすることが求められており、これを実現する方法について検討を行う必要がある。このことから本章では、マイクロUX法で考案した手法を実在のサービスアプリケーションに導入し、どのような成果物であれば効率的に導入できるか検討を行うこととした。そこで本章ではまず、スマートフォン用の地図サービスを題材に、屋内の目的地に移動することが難しい問題に着目し、マイクロUX法を試行する。改善手法の基本的性能を確認した後に、実在のサービスアプリケーションへの導入を試み、検証を行う。

5.1.1 UX課題の分解

スマートフォン用の地図サービスでは、地図情報を用いて目的の場所に関する情報を得て、目的の場所へ迷わず到達できることが重要な価値の一つである。しかし、商業施設や地下鉄構内などの屋内目的地への移動が困難な問題がある。その要因として、地図サービスで屋内の地図情報が表示されなかったり、屋内では地図上に現在地が表示されないUI

の問題が考えられる。

スマートフォンの地図アプリケーションはGPSなどの衛星測位システムを用いてユーザが今いる場所の緯度経度を計測し、アプリケーション内の地図上のどの位置にユーザがいるか算出しその場所を表示する。しかしGPSなどの衛星測位システムの電波は建物の中や地下施設内には届きづらく、ユーザの現在地を測位し地図上に表示することが難しい問題がある。このことに関連し、地図アプリケーションの多くは商業施設や地下鉄構内の詳細な地図情報に対応していない。商業施設や地下鉄構内などでは施設内の構造や出入口を示す屋内地図看板が設置されていることが多いが、これを完全に記憶し移動を行える施設利用者は多くない。商業施設や地下鉄駅のウェブサイトや屋内地図情報が掲載されていることがあるが、地図上に現在地が表示されることはなく周囲を見渡し推測する必要がある。Wi-Fiの電波や地磁気、RFIDを用いた手法もあるが、精度やコストが大きい問題がある。

本章ではこの問題に着目し、地図サービスで屋内目的地への移動が難しい問題に対し、屋内ナビゲーション手法の改善を行うこととした。

5.1.2 改善指標の定義

屋内ナビゲーション手法の改善を行うに当たり、定量的な改善指標の検討を行った。スマートフォン用の地図アプリケーションの登場により目的地へ移動する効率性は大きく向上した。ユーザは移動中いつでも地図が閲覧でき、常に現在地と目的地を把握しながら移動できるようになった。しかし、屋内ではGPSの電波が十分に届かないため、正確な現在地が地図上に表示されない。この問題によってユーザは最適な経路を把握できず、移動にかかるコストが増大していることが想定される。

そこで、屋内でも現在地を把握できるようになれば、屋内目的地への移動が難しい問題が改善すると考えた。この改善とは、GPS電波が届く屋外でのナビゲーション体験と同等以上の状態になることと考えられ、これをベースラインにするのが妥当だと考えた。ナビゲーションの効率性は目的地への移動に掛かる時間と距離に影響を与えられ、改善指標として妥当だと考える。また、GPSを用いたナビゲーション機能と同等以上に容易に扱えることも重要な改善指標として考えられる。これらは総移動距離と総移動時間、SUSスコアによって定量的に計測が可能であり、これを改善指標として定義することとした。ただし、本章ではサービスの導入方法の検討が主な目的であり比較実験は実施しない。

5.1.3 改善手法の検討

本章では、地図サービスで屋内目的地への移動が難しい問題に対し、ユーザが撮影した屋内地図の写真上にユーザの現在地を表示し追跡するナビゲーション手法「SCAN」を提案する。SCANは施設に設置されている屋内地図を撮影した画像を用いることで地図情報がない問題を回避する。撮影された屋内地図画像は形状補正が行われ、真正面から見たような地図画像として表示される(図 5.1)。同時に、撮影した屋内地図と撮影者との位置関係の計算も行い、屋内地図画像上に現在地と向いている方向の表示を行う。現在地情報取得後は、慣性計測ユニット (Inertial Measurement Unit; IMU) データの監視を行い、撮影地点を基準とした相対的な移動量を算出し、現在地と向いている方向を最新の状態に更新する(図 5.2)。IMU の値に誤差が生じ現在地や方向が実際とは異なる場合が発生する可能性があるが、その場合は施設内の看板や出入り口などのランドマークを再撮影することで位置情報の補正が行えるようにする(図 5.3)。こうすることで、施設内に特別な装置を設置したり特別な計測を行わなくても、ユーザは屋内でも現在地を把握しながら移動することができると考えた。本章ではサービスの導入方法の検討を行うために、SCAN のコンセプトが実証可能か確認を行った。具体的には、iOS 上で動作するプロトタイプアプリケーションの開発を行い、基本的な性能を確認する予備実験を行った。

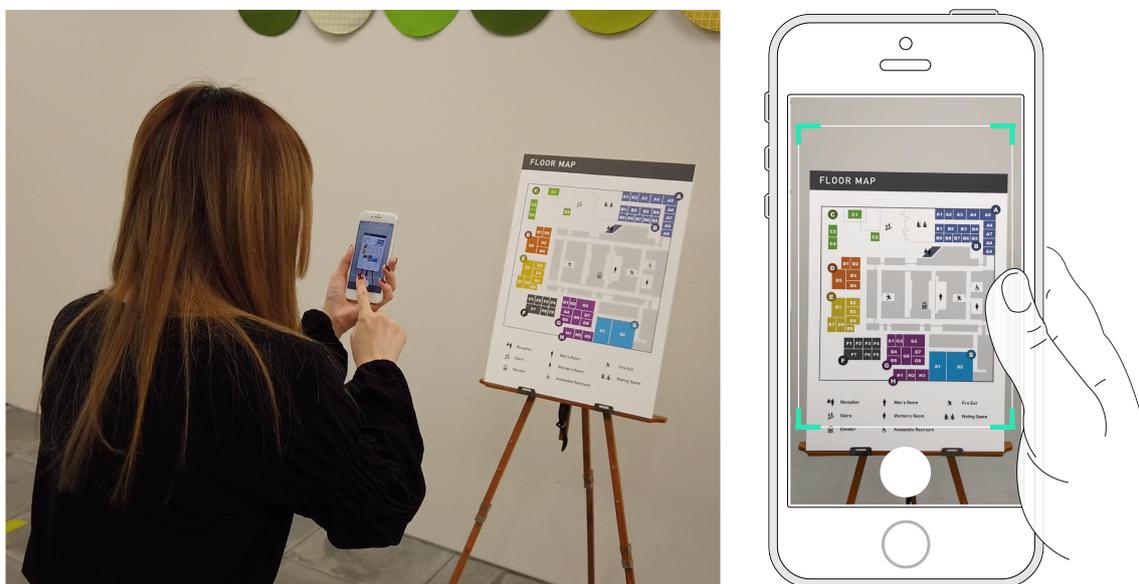


図 5.1: SCAN：ユーザが撮影した屋内地図の画像上に自身の現在地と向いている方向を表示する。



図 5.2: ユーザの移動はIMUによって計測され、撮影した屋内地図の画像上の現在地を更新する.

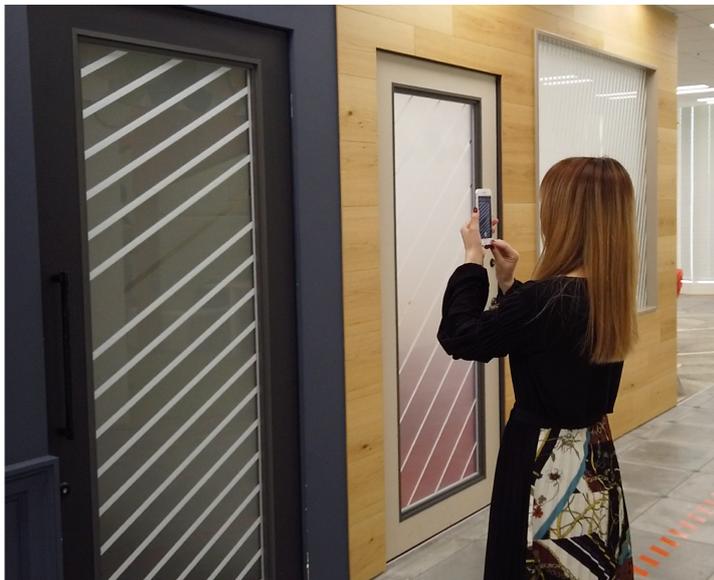


図 5.3: 地図上の現在位置の再補正方法. ランドマークを撮影し現在地の再計算を行う.

5.2 関連研究

屋内のナビゲーション手法については様々なアプローチが取られてきた。FootPath[52]はOpenStreetMap[53]と専用に作成された施設内地図を組み合わせ、スマートフォンに搭載された加速度センサと電磁コンパスを用いてユーザの位置と経路を表示する。Mulloniらは施設内に設置された複数あるマーカーをスマートフォンで読み取ることで、施設内の三次元地図上に経路を表示するシステムを提案している[54]。Kimらはワイヤレスカメラから入力された周辺画像を元にユーザの現在地を推定し、頭部装着ディスプレイ（HMD：Head Mounted Display）に表示された小さな地図にユーザの現在地と方向を重ね合わせて表示するシステムを提案した[55]。これらの手法は屋内地図上にユーザの現在地を追跡するが、システムで用いる地図データを事前に作成する必要がある。

Ozdenizciらはユーザが施設内に置かれた近距離無線通信（NFC：Near field communication）を読み込むと経路を表示するシステムを提案している[56]。Changらはユーザが施設内に設置されたQRコードからジオコードを読み取りサーバと通信を行い、携帯電話や携帯情報端末（PDA：Personal Digital Assistant）にテキストや画像を用いて経路を表示するシステムを提案した[57]。Bahlらは無線周波数（RF：Radio Frequency）を用いたRADARを提案した。これは対象となる領域に設置した複数の無線基地局システムを使用し、信号強度情報を事前に記録し処理を行うことでユーザの位置を特定する[58]。これらの手法は屋内に設置されたマーカーを読み込み現在地を明らかにするが、事前に施設内に発信機やマーカーなどの追加設備を設置する必要がある。

Wartnerらはユーザが撮影した画像と特徴点を照合し、ユーザと画像内に写るランドマークとの距離を算出した[59]。SCANはランドマークとの距離とユーザの位置の両方を算出でき、またIMUを用いるためリアルタイムにユーザの現在地を追跡できる。Betkeらは球体画像内の複数のランドマークを識別し、それらに対する現在の位置を算出した[60]。SCANはスマートフォンに搭載されたカメラを用いるため、特別なカメラを必要としない。

SCANのアプローチの特徴は事前に専用の屋内地図を作成する必要がなく、ユーザが撮影した屋内地図の画像上にユーザの位置と向いている方向を表示する点にある。また、一般的なスマートフォンに搭載されているカメラやIMUといったハードウェアから読み取った情報を元にユーザの位置を算出できるため、施設に特別な装置を接しする必要がない点も特徴的である。

5.3 提案手法：SCAN

SCANの主たるコンセプトはユーザが撮影した屋内地図の画像上にユーザの位置と向いている方向を表示し追跡することである。これによりGPSなどの衛星測位システムが使

用できない環境においても、ユーザは地図上の目的地及び自身の位置と向いている方向を常に把握することができる。以下にその動作手順と仕組みについて述べる。

5.3.1 概要

SCAN の概要を図 5.4 に示し、以下にその手順について説明する。

1. まず、ユーザは SCAN が実装されたスマートフォンアプリケーションを用いて任意の場所から屋内地図の撮影を行う (図 5.4 ①, 図 5.1)。
2. 撮影が行われるとシステム内で撮影された画像とリファレンスデータの特徴量マッチングが行われ、撮影された屋内地図が何か判別する (図 5.4 ②)。
3. 続いて Camera Calibration を行い、撮影された屋内地図の位置と撮影者 (ユーザ) の位置関係を算出する (図 5.4 ③)。
4. ユーザは任意の場所から任意の角度で屋内地図を撮影するため、撮影された屋内地図画像を真正面から撮ったかのように射影変換を行う。補正された屋内地図画像上に③で算出したユーザの現在地と向いている方向をマッピングする (図 5.4 ④)。
5. 以降はユーザが撮影した地点を基準とした相対的な移動量と方向を IMU から取得されたデータを元に算出し、連続的に屋内地図画像上のユーザの現在地を更新し反映する (図 5.4 ⑤, 図 5.2)。

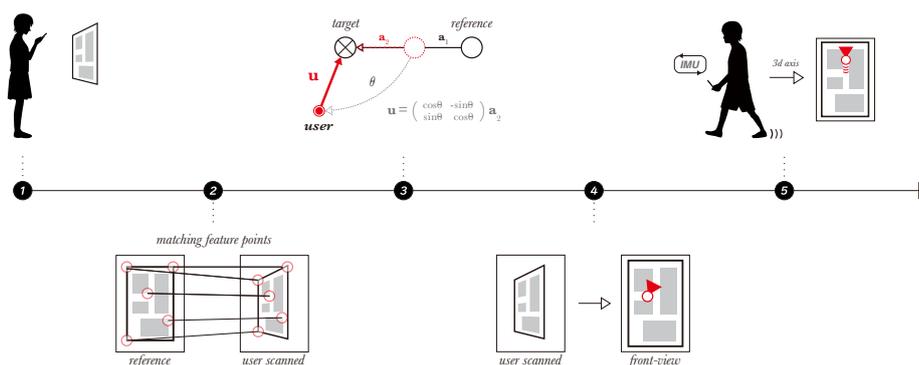


図 5.4: SCAN の概観

また、SCAN は IMU に誤差が生じた場合や、地図上の位置と自身の現在地とに差異が生じた場合に補正を行うことができる。ユーザが位置補正ボタン (図 5.2 の SCAN 画面

内左下のボタン)を押下するとシステムがカメラを起動する(図 5.3)。ユーザがカメラで周囲を撮影すると、その画像からランドマークと判別しランドマークとユーザの位置を算出する。処理が完了するとシステムは再び地図画像を表示し、修正されたユーザの位置と向いている方向を表示する。

5.3.2 追跡手法

SCANは、多くのスマートフォンに搭載されているIMU(慣性計測ユニット)を用いてユーザの移動した距離と方向を算出する。ユーザが屋内地図を撮影するためにSCANを起動すると、画面中心部が世界座標の基準点と設定され、この基準点からの距離が連続的に測定される。距離の単位はメートルを用いる。屋内地図の実寸と補正された地図画像の縮尺は既知であり、実空間での移動距離を地図画像上での移動距離に変換することでリアルタイムな現在地の表示と更新を実現する。

5.3.3 キャリブレーション

SCANは事前にターゲットとして登録された地図をユーザが撮影することでユーザの現在地を算出を行う。システムは自己位置推定のためにユーザが撮影した屋内地図画像から特徴点を検出し特徴量を抽出する。ここではAKAZEアルゴリズムを用いて計算されたスケールや回転、アフィン不変の特徴を用いた。SCANはこの特徴量と事前に用意した屋内地図の特徴量をマッチングさせ、最も適合率の高い屋内地図画像データを検出する。そして特徴量マッチングの結果を用いてホモグラフィ行列を推定し、ユーザが撮影した屋内地図画像を正面から見たかのような理想的な画像となるよう射影変換を行う。以降はこの画像を自己位置を重畳させる地図として用いる。また、推定されたホモグラフィ行列を用いてより正確なカメラの自己位置を計算し、それを地図画像上に重畳する処理を行う。

事前準備

SCANは事前にリファレンスとなる屋内地図画像の特徴量データと、その屋内地図が屋内地図画像上のどの座標に位置するかといった情報が必要となる。こうした事前準備に必要な手順を図 5.5 に示し、以下にその手順について説明する。

1. SCANを用いたサービス提供者は、まず施設内に設置されている屋内地図の画像の撮影を行う(図 5.5 ①)。
2. ①で撮影された画像に射影変換処理を行い、真正面から撮った画像のように形状を補正する(図 5.5 ②)。

3. ②の画像特徴を抽出し、②の画像上における地図設置位置と②の画像を撮影した位置の座標 (x,y) のデータを1セットとして保存する (図 5.5 ③).
4. IMU 誤差等の補正時に用いるランドマークのデータセットも同様の処理を行い収集する (図 5.5 ④).
5. 設置されている地図のサイズ (メートル) と地図画像のサイズ (ピクセル) の比率を計算し、IMU から取得する移動量 (メートル) の変換係数として用いる (図 5.5 ⑤).

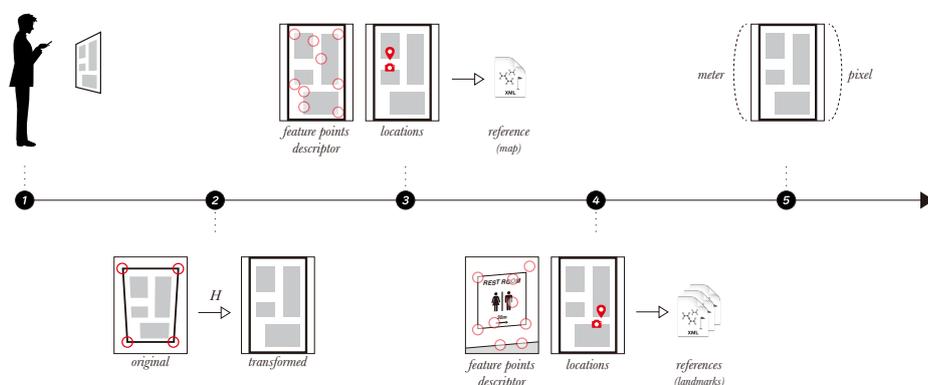


図 5.5: 事前データの収集方法

後述する予備実験実施時に行った大型の商業ビルの1フロアでの事前準備では30分ほどで撮影及びデータ収集が完了した。撮影には特別な機材は必要なく、スマートフォンに標準的に搭載されているカメラアプリケーションだけで行える。SCANはこれらのデータをリファレンスデータとして扱い、ユーザが撮影した屋内地図画像からユーザの地図上の現在地の算出を行う。

ユーザ位置算出の仕組み

SCANはユーザが撮影した屋内地図画像を用いユーザの現在位置を算出する。入力された屋内地図が写った画像とリファレンスとなった画像(特徴)による位置算出の仕組みについて図 5.6 に示し、以下にその手順について説明する。

1. ユーザが屋内地図を撮影した位置と、リファレンスデータである屋内地図が設置されている位置、リファレンスデータである屋内地図を撮影した位置があり、この3者

- の相対的な位置関係を Camera Calibration によって 3 次元空間上に求める (図 5.6 ①).
- ①で求めた 3 次元座標情報を地面に射影し, 撮影対象までの距離比率を計算する (図 5.5 ②).
 - Camera Calibration によって各々のカメラの光軸方向も求め, 各々の光軸のなす角度, すなわちユーザが向いている方向を計算する (図 5.5 ③).
 - 以上の計算結果から撮影対象とユーザの位置している距離と方向が既知となり, まず二次元座標上の距離を求める (図 5.5 ④).
 - 続いて④をユーザの向いている方向へと回転させ, 地図画像上のユーザの現在地を算出する (図 5.5 ⑤).

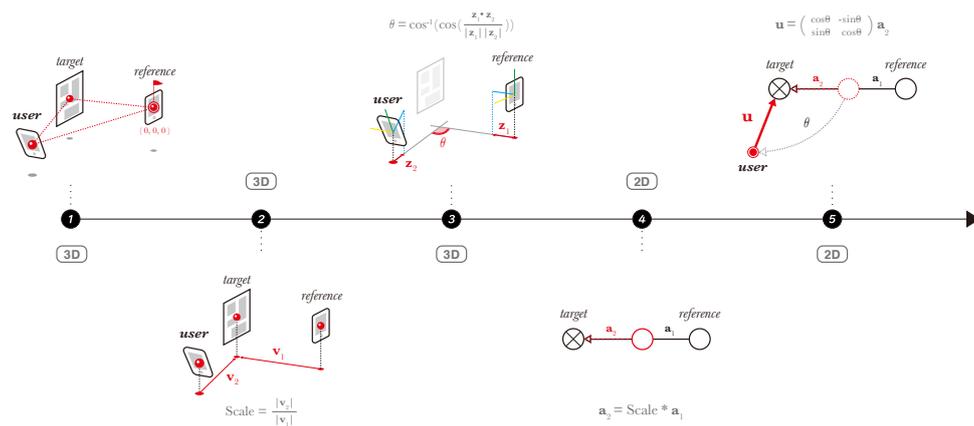


図 5.6: ユーザ位置算出の仕組み

この仕組みによってより簡単な操作でユーザの自己位置を正確に求めることが可能となる。また, 通常 Camera Calibration では事前に対象物の 3 次元情報や複数の視点から撮影された画像が必要となるが, この仕組みによって対象となる屋内地図やランドマークを 1 枚撮影するだけで自己位置の計算を行うことができる。

ユーザ位置の誤差補正

SCAN ではユーザが移動した際に地図画像上のユーザの現在地を更新するために IMU が計測するデータを用いて反映処理を行うが, IMU の計測値は誤差が蓄積することが想定される。IMU の計測値に誤差が蓄積し続けると地図画像上のユーザの現在地と現実のユー

ザの現在地の差が大きくなり、ユーザは違和感を感じるようになる。このため IMU の誤差を補正するためにユーザの現在地を再計算する必要がある。SCAN ではユーザがユーザの現実の現在地と地図上の現在地とのズレが大きくなったと感じられた場合に、周辺のランドマークを撮影することで現在地の誤差を補正することもできる。ランドマークは事前にデータを収集し設定されたものであり、ユーザにはどういったものが誤差補正用のランドマークに該当するか明示する。ユーザが現在地の誤差を補正するためにランドマークの撮影を行うと、その撮影画像から特徴点を抽出し特徴量を計算する。SCAN はこの特徴量と予め用意されたランドマークの特徴量とを照合し、最も適合率が高いランドマークを検出する。この結果を元に 5 点アルゴリズム [61] を用いて基本行列を計算し、自己位置と対象ランドマークとの相対的な位置関係を求める。ランドマークの実際の位置とランドマークを撮影した位置は既知であり、ランドマークの実際の位置と現在の撮影した位置との位置関係も算出できる。この処理によって明らかになった現在位置を新たな基準点とし、以後の IMU から得られるデータの補正值として適用することで誤差が補正される。

5.4 予備実験

我々はユーザが SCAN が搭載されたアプリケーションをどのように利用するか観察し、SCAN が提供する目的地へのナビゲーションをどのように理解したか調査するための予備実験を行った。我々はユーザが SCAN が搭載されたアプリケーションを用いて屋内地図の撮影を行い、その屋内地図の画像上に表示される自身の現在地を確認しながら屋内の目的地へ到着するまでの行動を観察し、SCAN の操作性や屋内ナビゲーションの有効性について調査した。

5.4.1 実験で使用する小型携帯端末

SCAN は iOS バージョン 12.3 を搭載した iPhone 7 上で実装した。ディスプレイは解像度 $1,344 \times 750$ で 326ppi の 4.7 インチだった。実験は SCAN が実装された実験用アプリケーションをインストールした iPhone 7 のデモ機を実験参加者に手渡し実施した。実験参加者の移動や操作は実験参加者の後ろを追いながら静かに観察した。

5.4.2 実験参加者と手続き

この実験には 4 人のボランティアが参加した。男性が 2 人、女性が 2 人、年齢は 24 歳から 41 歳で平均年齢は 33.5 歳 (s.d.=7.14) であった。参加者全員がスマートフォン (iPhone) を所有しており (100.0%)、スマートフォンの地図アプリケーションの操作経験は豊富で

あった。

実験の手順を簡単に述べる。実験参加者各自は指定の時間になったらオフィスビルの特定のフロアに集合し、SCANの機能やインターフェースについてSCANが搭載されたアプリケーションを見ながら口頭で説明を受けた。続いてSCANを利用して現在いるフロアの指定された部屋へ移動するよう指示された。移動中に室内地図画像上のユーザの現在地に誤差が生じたと感じられた際には、現在地を修正するために再撮影による誤差の補正を行うよう指示した。



図 5.7: 予備実験のスタート地点。この屋内地図を撮影したSCANを使用しながら、指定された部屋へ移動を開始する。

5.4.3 実験結果

実験参加者全員がSCANが実装されたアプリケーションを用いて屋内地図の撮影が行なえ、形状補正された地図画像上に現在地と向いている方向が概ね正しく表示された。実験参加者が移動した際には地図上の現在地が概ね正しく追跡された。実験参加者全員がSCANが実装されたアプリケーションを用いて目的地へ移動することができた。また、実験参加者が屋内地図上の現在地に誤差を感じた際には、ランドマークの撮影によって現在地の補正を行うことができた。実験後の簡単なアンケートでは、実験参加者全員が屋内でも自分の現在地と向いている方向を認識し、自信を持って目的地へ向かうことができた。

回答した。実験参加者のうち1名は、目的地となった部屋の入口の位置が分からず部屋がある区画を一周したが、これは設置されている屋内地図そのものの視覚情報が足りないことが要因と考えられる。

5.4.4 考察

予備実験では、実験参加者全員がSCANが実装されたアプリケーションを用いて目的地へ自信を持って移動することができた。また、地図画像上の現在地に誤差を感じた際に、ランドマーク撮影による位置の誤差補正を自主的に行うことができた。

一方で、実験後の簡単なインタビューでは実験参加者のうち3名から、位置の誤差補正を行うランドマーク撮影において何が対象であるか、どの程度の範囲を撮影すべきか分かりづらかったという意見があった。また、写真を撮影し位置を補正するという操作に馴染みがなく、操作と結果のイメージが掴みづらかったという意見もあった。これらのことから、ランドマーク撮影以外の現在地補正手法の検討も行う必要があると考えられる。また、スマートフォン用の地図アプリケーションの一部には、主要なデパートや地下鉄駅構内、地下商店街などの屋内地図情報を表示するものもある。SCANは屋内地図撮影時に屋内地図と撮影者の距離と方向を正確に算出でき、移動に伴う移動距離と方向をリアルタイムに計測する機能を持つ。SCANのこの主要な機能を応用することで、地図アプリケーションで屋内屋外問わないナビゲーションが実現できる可能性が示唆された。SCANの発展手法ではこれらの点を考慮し改良を加えることで、より効率性と汎用性の高い手法になる可能性がある。

SCANは一部に改善の余地が確認されたが、SCANの基本的な操作性や機能性について概ね問題がないことが確認され、SCANはGPSを用いたナビゲーションと同等の体験が得られるナビゲーション手法である可能性が示唆された。

5.5 ミクロUX法の適切な成果物の検討

予備実験においてSCANの基本的な操作性や機能性に重大な問題は見つからなかったことから、SCANを実際のサービスアプリケーションに導入するための成果物の形について検討を行う。3章で考案したFix and Slideはより速いキャレット操作を行えるが、サービスの既存システム構造を大きく修正する必要があった。4章で考案したBounce BackとForce Zoomはより少ない回数でズーム操作を行えるが、サービスのインタラクションデザインを全体的に見直す必要があった。これらのことから、本章ではデザイン面とシステム面においてサービス側の導入コストが最小限になる成果物の形を検討し、SCANに適用することとした。

本章では、実在する地図アプリケーションを対象に SCAN の導入を試み、適切な成果物について検証を行う。導入を行うに当たり、まず予備実験で得られた考察を元に SCAN の発展手法を検討し、続いて SCAN の発展手法が効率的に導入できる成果物の形について検討を行う。

5.5.1 SCAN の発展手法の検討

本章では、日本国内で広く普及しているスマートフォン用の地図アプリケーション (Yahoo! 地図 - ヤフー株式会社) を対象に、SCAN の導入を試みることにした。この地図アプリケーションは主要な商業施設やターミナル駅などの施設内地図情報を提供しており、SCAN の主要機能を応用することで施設の外から中へのシームレスなナビゲーションが実現できると考えた。そこで、SCAN の屋内地図を撮影し写真上に現在地を表示する機能は省略し、ランドマーク撮影で現在地を計測する手法と VIO (Visual-Inertial Odometry) を用いた現在地を追跡する手法を用いることにした (図 5.8)。

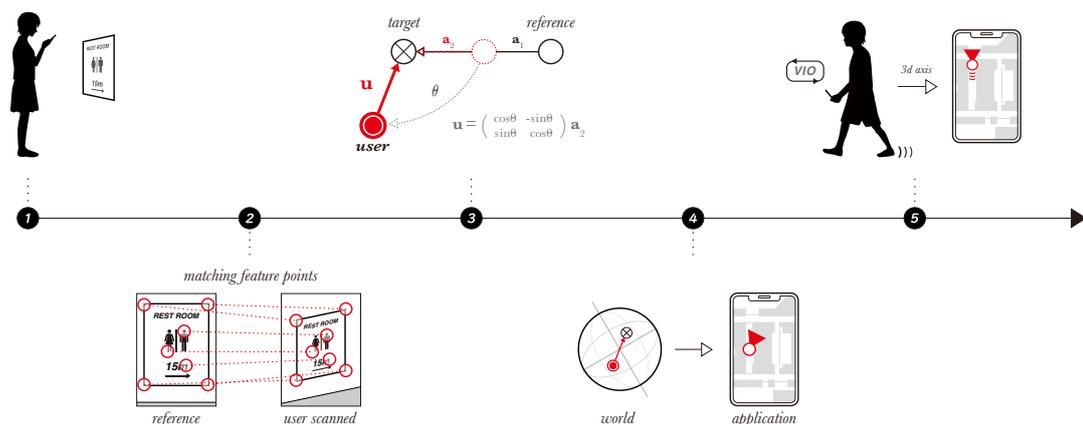


図 5.8: SCAN の発展手法の概要.

現在地の取得方法

SCAN はユーザが撮影した屋内地図画像からユーザの現在地を算出し表示を行うが、発展手法では屋内地図ではなく屋内に設置されているランドマークの撮影画像から現在地を算出する。屋内施設では施設内案内などの看板はデザインが統一されており、またエスカレータやエレベータ付近など一定の規則性を持って配置されている。予備実験で得られたランドマークの対象が分かりづらいという問題に対し、発展手法では視認性が高く行動起点に置かれていることが多い案内看板がランドマークに適切だと考え採用することとした。

アプリケーションがGPSの受信強度が低く室内だと判定した際に表示されるSCANを起動するボタンをユーザが押下すると、SCANのカメラ画面がモーダル表示される(図 5.9)。そしてランドマークとなる屋内案内看板を撮影しSCANが撮影地点の緯度経度を算出するとカメラ画面が閉じられ、現在地が算出された緯度経度となっている地図アプリケーションの地図画面が表示される。



図 5.9: 発展手法における現在地測定手法。ランドマークを撮影し現在地を算出する。

現在地の補正方法

SCANでは現在地の誤差を補正するために、VIOから得られる移動量の基準点を新たに書き換えることで誤差のリセットを行ってきた。発展手法ではランドマーク撮影によって現在地を取得するが、誤差が発生した場合にも同様の方法でリスタートすることができる。しかし予備実験の結果より、ユーザは現在地の補正とカメラによる撮影という行動がリンクしづらかったという問題があった。そこで発展手法では再撮影によるリセットの他に、地図上の現在地を指で動かし補正する手法も考案した(図 5.10)。地図アプリケーションの地図画面上に表示されたボタン(図 5.10 左画面の赤丸)を押下すると、位置補正モードになる(図 5.10 中央画面)。この状態で地図画面を指で動かすと、点線の丸が表示される(図 5.10 右画面)。ユーザはこの点線の丸を正しい現在地へと移動させ、再度位置補正モードボタンを押下するとその位置に現在地が更新される。



図 5.10: 発展手法における現在地補正手法. 位置補正モードに移行し, 指で現在地を正しい位置に移動させる.

5.5.2 サービスへの導入方法の検討

この地図アプリケーションにSCANの発展手法を導入方法を検討するに当たり, サービス側に大規模な改修が必要にならないようソフトウェア開発キット (Software Development Kit; SDK) での提供が適切ではないかと考えた. すなわち, 現在地を取得するためのカメラ機能を有し, 撮影後は現在地を常時取得し返答するインターフェースを持つ SDK である. 多くの地図アプリケーションは, スマートフォンが提供する位置情報 API にアクセスし現在の緯度経度情報を取得することで, 地図上に現在地を表示している. しかし, GPS の電波を受信できないなど位置情報が取得できない状況になると, 地図上の現在地を非表示にする仕様であることが多い. この点に着目し, 位置情報 API から緯度経度が取得できない際には SCAN から位置情報が提供される構造とすれば良いと考えた. こうすることで既存の位置情報関連の処理が再利用でき, 実装するための改修コストを抑えることが可能となる. また, 多くの地図アプリケーションに搭載されている現在地の追跡表示機能の有効無効を切り替えるボタンは, 位置情報が取得できない際には非表示もしくは無効状態になるが, その場合には SCAN の有効無効を切り替えるボタンに差し替わる設計とすれば良いと考えた. こうすることで, 位置情報を取得するロジックは異なるものの, ユーザから見れば現在地を追跡するためのボタンのままであり, 既存のコンテキストに沿ったインタラクションが実現できる. このような SDK とすることで, サービス側へ容易に導入が行えると考えた. また, 不要となった際にも容易に取り外しができると考えられ, 反復的な設計と検証を行う上で試すリスクを軽減する効果も見込める.

検討したサービスへの導入方法を元に, カメラ機能を持つ UI と現在地取得を行う API

を持つ SCAN SDK の開発を行い、地図アプリケーションへの実装を行った。地図アプリケーションの既存システムを流用することで位置情報取得処理部の修正だけで完了することができた。そして SCAN SDK が実装された地図アプリケーションを実際の地下商店街で動作検証を行い、屋内空間で現在地の取得および追跡が機能することを確認した。



図 5.11: SCAN を拡張し実装した Yahoo!地図アプリの画面。赤丸の部分が SCAN が提供する UI パーツ。

5.6 まとめ

本章では、マイクロ UX 法の適切な成果物の形を検討するために、成果物を実在の地図アプリケーションに導入し検証を行った。まず本章では、地図サービスで屋内目的地への移動が難しい問題に対し、撮影した屋内地図画像上に現在地を表示し追跡する手法「SCAN」を考案した。予備実験を実施し、SCAN の基本的な操作性や機能性に問題がないことを確認した。続いて実在の地図アプリケーションへの導入方法について検討を行い、3 章と 4 章の考察を元に現在地を取得するカメラ機能と現在地を算出する API を持つシンプルな SDK での提供を行うことにした。また、予備実験の考察を元に SCAN を発展させた。SCAN SDK を開発し、地図アプリケーションへの実装を行った。そして実際の地下商店街で動作検証を行い、屋内空間でも現在地取得および追跡が機能することを確認した。マイクロ UX 法の成果物はデザインとシステムの両面で影響が最小化されるよう設計することで、サービスへの導入コストが小さくなる可能性が示唆された。

第6章 総合的な議論

本論文では、ITサービスの開発現場でより早くより効率的な開発サイクルを実現するために、アジャイル開発の開発サイクルに並走可能なUX改善のアプローチが有効だと考え、マイクロUX法を構想した。マイクロUX法は、UX課題をUI課題へ分解することでUX改善の対象を極小化し、UX改善のコストとリスクを低減する。また、UX改善指標を定量的に定義することでUX改善の評価を単純化し、UX改善の施策と結果の要因を明確にする。こうすることで、アジャイル開発の開発サイクルと並走可能な規模感であり、反復的な設計と検証に効果的に組み込める成果物が創出できるUX改善のアプローチを実現できると考えた。

本論文では、マイクロUX法の構想を検証するために、3章から5章にて実際のITサービスを題材にマイクロUX法を試行した。本章では各研究を振り返り、マイクロUX法の総合的な議論を行う。

6.1 ミクロUX法の試行に関する振り返り

本論文では、3章から5章にてスマートフォン用のメールサービスと地図サービスを題材に、マイクロUX法を試行した。3章では、メールサービスで文章作成が難しい問題に対しマイクロUX法を試行し、文章のポインティングやセレクションで誤操作が起きやすい問題の改善を試みた。4章では、スマートフォン用の地図サービスで片手操作が難しい問題に対しマイクロUX法を試行し、片手でのズーム操作が難しい問題の改善を試みた。5章では、3章と4章の考察を元にマイクロUX法の最適な成果物の形について検討を行った。まず地図サービスで屋内目的地への移動が難しい問題に対しマイクロUX法を試行し、屋内で現在地が表示されない問題の改善手法を開発した。この改善手法を実在する地図アプリケーションに実装し、サービス側の導入コストを最小化する成果物の形について検討を行った。

6.1.1 メールサービスで文章作成が難しい問題

3章ではスマートフォン用のメールサービスで文章作成が難しい問題に対しマイクロUX法を試行した。マイクロUX法では、まず文章作成が難しい問題に関与するUIの問題の抽出を行う。画面が小さく文章全体を俯瞰し推敲するのが難しい問題や、ソフトウェアキー

ボードでは高速に入力することが難しい問題、文章のポインティングやセレクションが誤操作されやすい問題などが考えられる。マイクロ UX 法は反復的な開発サイクルを前提としているため、どの問題を選ぶべきかは問わない。ここでは文章のポインティングやセレクションが誤操作されやすい問題に着目した。

タッチディスプレイでは指が対象を隠してしまう Fat Finger Problem が知られており、本問題でも要因の一つとして考えられる。この問題は操作に掛かる時間および操作の利便性に影響を与えていると考えられ、タスク完了時間と SUS スコアを改善指標として定義した。この改善指標に有効な手法の検討を行い、目的の文字の位置へキャレットを移動しようとする Fat Finger Problem が発生することから、逆に目的の文字をキャレットの位置へ移動する Fix and Slide という手法を提案した。改善指標を基準に提案手法と既存手法の比較実験を行った結果、提案手法は既存手法よりタスク完了時間が有意に短く、SUS スコアが有意に高いことが確認された。

マイクロ UX 法の試行に対する考察

3章では、メールサービスで文章作成が難しい問題を分解し、文章のポインティングやセレクションの操作が難しい問題の改善を行い、より速く操作が実行できる手法 Fix and Slide を実現した。一方で課題も残った。Fix and Slide はテキスト領域全体に影響を及ぼす実装であることから、サービス導入の際に既存システムを大規模に改修しなくてはならない可能性が高い。また、テキストの入力エリアはサービスによって様々な形状をしており、入力エリアが文章 1 行分の高さしかない場合なども考えられ、Fix and Slide が十分に機能しない可能性がある。すなわち、Fix and Slide はサービスの状況に対する適応性が低く、サービスごとに最適化を行わなくてはならない懸念がある。本試行では、マイクロ UX 法で UX 課題の分解と改善は実現したが、改善手法がサービスの既存システムに与える影響について課題が残った。

6.1.2 地図サービスで片手操作が難しい問題

4章ではスマートフォン用の地図サービスで片手操作が難しい問題に対しマイクロ UX 法を試行した。片手操作が難しい問題に関与する UI の問題として、スマートフォンが大画面化し画面の端に配置されている検索窓やボタンなどに指を届かせるのが難しい問題や、ズームを行うためのジェスチャ操作が片手で実行するのが難しい問題などが考えられる。ここではズーム操作の問題について着目し改善を試みることにした。

ズーム操作は地図サービスの様々なシーンで多用される操作の一つであるが、これを実行するには片方の手でスマートフォンを保持し反対の手で指と指の間隔を近づけたり遠ざ

けたりするピンチイン・ピンチアウト操作を行う必要がある。そのため、片手が塞がっている状態でズーム操作を行うことは難しく、十分な操作を行うためにはスマートフォンを両手で持ち直さなくてはならない。この問題は操作に掛かる時間および操作を行う回数、操作の利便性に影響を与えていると考えられ、タスク完了時間と操作回数、SUS スコアを改善指標として定義した。この改善指標に有効な手法として、複雑な操作を必要とせず指一本で操作可能なアプローチを検討した。そこで、スマートフォンに搭載されている圧力感知マルチタッチ技術に着目し、ユーザがディスプレイを押す力量に応じた縮小率で飛び上がるようにズームアウトする Bounce Back と、力量に応じた速度で遠ざかるようにズームアウトする Force Zoom という片手ズーミング手法を提案した。改善指標を基準に提案手法と既存手法の比較実験を行った結果、提案手法は他手法より有意に操作回数が少なく、先行研究手法より SUS スコアが有意に高かったことが確認された。

マイクロ UX 法の試行に対する考察

4章では、地図サービスで片手操作が難しい問題を分解し、片手でのズーム操作が難しい問題の改善を行い、より少ない回数で操作が完了できる手法 Bounce Back と Force Zoom を開発した。一方で課題も残った。Force Zoom と Bounce Back は圧力感知マルチタッチ技術を用い、スマートフォンのディスプレイを押した力の強さに応じてズーム操作が実行されるインタラクションを実現した。しかし、実際のスマートフォン用地図サービスによっては様々なジェスチャー機能が実装されており、地図を長押しし目的地を指定するといった既存ジェスチャーとの衝突が懸念される。これを解決するためには、ズーム機能を実行する圧力の閾値を上げ、弱い力の時は長押しと判定するなどの回避策が考えられるが、操作性が低下する恐れがある。すなわち、提案手法を実際にサービスに導入するには、既存ジェスチャーとの衝突を回避するためにサービス全体のインタラクション設計の見直しが必要となり、UX 改善のコストが大きくなることが想定される。本試行では、マイクロ UX 法を用いて有用な手法を考案したが、その手法がサービスのインタラクションデザイン全体に与える影響について課題が残った。

6.1.3 地図サービスで屋内目的地への移動が難しい問題

5章では、3章と4章で取り組んだマイクロ UX 法について考察し、マイクロ UX 法の効果的な成果物の形について検討を行った。3章と4章で考案した手法は UX の課題を分解しシンプルな改善を実践できたが、既存のシステムやデザインとの衝突が想定され、サービスに組み込む際のコストが大きくなる懸念があった。そこで5章では、マイクロ UX 法で考案した手法を実在のアプリケーションに導入し、どのような形であれば効率的に組み込め

るか検討を行った。

導入を行うに当たり、5章ではスマートフォン用の地図サービスで屋内目的地への移動が難しい問題に対しマイクロ UX 法を試行し、改善手法の開発まで実施した。屋内目的地への移動が難しい問題に関与する UI の問題として、屋内の詳細な地図情報が表示されない問題や、屋内で現在地が表示されない問題がある。ここでは後者の問題に着目し改善を試みることにした。通常、多くの地図サービスは GPS の情報を用いて地図上の現在地を特定し表示している。しかし屋内では GPS の電波が届きづらいため、正確な現在地を表示することが難しい。この問題は目的地へ移動する時間と距離に影響を与えていると考えられ、総移動時間と総移動距離、SUS スコアを改善指標と定義した。この指標に有効な手法として、ユーザが撮影した屋内地図画像上に現在地を表示し、ユーザの移動量に応じて現在地が更新される SCAN という手法を提案した。5章はマイクロ UX 法の適切な成果物の形を検討することが目的であることから、SCAN の基本的な性能に関する予備実験を行うことにした。商業施設内の目的地へ移動するタスクを課した結果、すべての実験参加者が SCAN を用いて問題なく移動できたことが確認され、GPS を用いたナビゲーションと同等の体験が提供できる可能性が示唆された。

サービスに導入するためのアウトプットの検討

SCAN の基本的な性能を確認できたことから、一般に広く普及している地図サービスのスマートフォン用アプリケーションへ SCAN を導入する方法について検討することとした。この地図サービスは主要な屋内施設の地図情報が表示できることから、SCAN のカメラによる現在地取得および移動量を用いた現在地追跡機能のみを提供することとした。これまでの取り組みから、既存システムへの影響が大きい実装は避け、既存デザインのコンテキストから大きく逸脱しないアプローチが有効だと考えた。この地図サービスでは GPS の取得状況を分岐点として現在地表示関連の処理を切り替えていることに着目し、GPS が無効な状況では SCAN が代わりに位置情報を提供する構造とすれば良いと考えた。具体的には、GPS が無効な状況になると現在地取得ボタンは SCAN 起動ボタンに切り替わり、現在地表示ロジックは GPS から取得した位置情報ではなく SCAN から取得した位置情報を用いた処理に切り替わる。このように既存システムを流用することで改修コストを最小限にでき、既存デザインのコンテキストにも適合できるようになる。こうした成果物を創出するために、ここではソフトウェア開発キット (Software Development Kit; SDK) での提供が適切ではないかと考えた。これは、現在地を取得するためのカメラ機能と、移動量から算出された現在地を返すインタフェースを持つ、シンプルな SDK である。サービス側のコストを考慮し、SDK には GPS 無効時に表示される SCAN 起動ボタンの UI も含

めるが、サービス側の自由度を高めるために差し替えが可能な構成とした。

このように実装された SDK を用いて、実在の地図サービスアプリケーションへの実装を試みた。SCAN を既存の機能を代替するシンプルな SDK とすることで、地図アプリケーション側の修正を最小限に止めて導入することができた。実装完了後、SCAN SDK が実装された地図アプリケーションを用いて、大規模な地下商店街の店舗から店舗へ移動する動作検証を実施した。その結果、SCAN を用いたランドマーク撮影による現在地取得とユーザ移動による現在地更新という基本的な機能が問題なく動作することが確認された。これらの検討と実装、動作検証の結果から、マイクロ UX 法の成果物はシステムとの接点は必要最小限でありデザインのコンテキストは極力変更しないアプローチとすることで、サービス導入のコスト低減に有効である可能性が示唆された。

6.2 ミクロ UX 法の考察

本論文では、IT サービスの開発現場でより早くより効率的な開発サイクルを実現するために、アジャイル開発の開発サイクルに並走可能な UX 改善のアプローチとして、マイクロ UX 法を構想した。これを検証するために実際の IT サービスを題材にマイクロ UX 法を試行した。これらから得られた考察を元に、マイクロ UX 法の手順と成果物について考察を行う。

6.2.1 ミクロ UX 法の手順

マイクロ UX 法は、UX 課題を UI 課題に小さく分解することで、UX 改善を行う対象を極小化し小規模で短期間に UX 改善を実現する。また、UX 改善の指標を定量的に定めることで、UX 改善の評価を単純化し UX 改善の要因を明確化する。これにより UX 改善の施策と効果の因果関係が明らかになり、反復的な設計と検証が効果的に実施できる。

これを実現する具体的なマイクロ UX 法の手順として、1. UX 課題の分解、2. 改善指標の定義、3. 改善手法の開発、4. 評価実験の実施、の 4 つのステップを構想した。ここでは、マイクロ UX 法の試行結果を踏まえ、マイクロ UX 法の手順の詳細をまとめ、考察する。

1. UX 課題の分解

マイクロ UX 法では、小規模で短期間に UX 改善を実現するために、UX 改善の対象を極小化する。UX の課題に関与し得る UI の課題を抽出し、その中から改善対象を選択する。どの課題から取り組むべきかは反復的な設計と検証を行う上で重要ではないが、UX 課題と関連性が強い UI 課題や、有効性の高い改善が望める UI 課題を選択するのが妥当と考

える。

3章では、メールサービスで文章作成が難しい問題に対し、画面が小さく文章全体を俯瞰した推敲が難しい問題、ソフトウェアキーボードで高速入力が難しい問題、文章のポインティングやセクションが難しい問題などが関与する UI 課題として想定された。4章では、地図サービスで片手操作が難しい問題に対し、スマートフォンが大型化し画面隅の UI パーツに指が届かない問題、片手でのズーム操作が難しい問題などが関与する UI 課題とした想定された。5章では、地図サービスで屋内目的地への移動が難しい問題に対し、屋内地図情報が表示されない問題、屋内で現在地情報が表示されない問題などが関与する UI 課題として想定された。

UX 課題を UI 課題に分解する際、ソフトウェア上の UI 課題に限らずハードウェアや OS 標準 API まで視野を広げて検討する。対象となるサービスの目に見える現象や、手で触った感触などを注意深く観察することで、具体的な課題を抽出することができる。様々な観点で検討を行うことで、UX が抱える課題を多角的に把握することができる。

2. 改善指標の定義

改善対象となる UI 課題を決定した後、その UI 課題がユーザに与える影響を定量的に定義する。操作に時間が掛かったり、操作を何度も繰り返すなど、定量的に計測可能な影響を見つけ出す。

3章では、文章のポインティングやセクションが難しい問題に取り組み、文章作成に掛かるコストへの影響が考えられることから、タスク完了時間と SUS スコアを改善指標とした。4章では、片手でのズーム操作が難しい問題に取り組み、ズーム操作自体に掛かるコストへの影響が考えられることから、タスク完了時間と操作回数、SUS スコアを改善指標とした。5章では、屋内で現在地情報が表示されない問題に取り組み、移動に掛かるコストへの影響が考えられることから、移動総距離と総時間、SUS スコアを改善指標とした。

UI の定量的な改善指標を定義するには、Hartson らの UX メトリクス [10] などが応用できる。パフォーマンスに関する指標としてはタスク達成時間、タスク成功率、エラー率、アクション数などが挙げられる。これらの指標を複数設定することで、UI 改善の有効性が多角的に評価され、改善の妥当性が高まる。また、これらの指標を軸に UX 課題の分解を行うことで、パフォーマンスに関する UI 課題の発見が容易になる効果も期待できる。

3. 改善手法の開発

UI 課題と改善指標が決定した後、その指標に有効な改善手法の検討を行う。UI 課題がどのような要因によって改善指標に影響を与えているか俯瞰して観察し、その要因を適切

に排除する。

3章では、誤タップを回避しより速く操作が行えるよう、キャレット操作を容易にする Fix and Slide という手法を開発した。4章では、片手でより効率的にズーム操作が行えるよう、1本の指で力を加えるだけでズーム操作可能な Bounce Back と Force Zoom という手法を開発した。5章では、屋内で道に迷わずスムーズな移動が行えるよう、撮影した屋内地図写真上に現在地を表示する SCAN という手法を開発した。

より効果的な改善手法を考案するには、より高精度な要因抽出が求められる。要因がシンプルになるほど対策もシンプルになり、より直接的な効果がある改善手法が着想しやすくなる。同時に、微視的に考えすぎ他の指標に悪影響が出ないように留意する必要がある。

4. 評価実験の実施

改善手法を決めた後、改善手法の有効性を確認する実験計画を立てる。改善手法と既存手法との比較を中心とした改善指標の比較実験を行う。実験計画では、実験参加者の属性、実験参加者に課すタスク、タスクに適用する条件など、実験と分析に用いる実験データの定義を行う。実験計画が定まった後、改善手法と既存手法でタスクが実施できる実験用アプリケーションの開発を行う。実験用アプリケーションの完成目処が立ったら、実験参加者の募集や実験環境の準備を行う。すべての準備が整った後、実験を実施しデータの収集を行う。実験データを十分に収集した後、実験データを整理し分析を行う。実験条件と改善指標を組み合わせ、統計的な処理を行い分析する。分析結果をもとに実験条件と改善指標ごとの定量的な特徴を多角的に評価し、どのような要因によって改善を実現したか明らかにする。

3章では、Fix and Slide の有用性を確認するために、文字列選択タスクを課しタスク完了時間の平均時間を計測し、既存手法との比較を行った。その結果、提案手法は有意に速い操作手法であることが確認された。4章では、Bounce Back と Force Zoom の有用性を確認するために、特定のズーム率に合わせるタスクを課しタスクを完了した平均時間と平均操作回数、SUS スコアを計測し、既存手法と先行研究手法との比較を行った。その結果、提案手法は有意に操作回数が少ない手法であることが確認された。

実験は改善手法と既存手法を正しく比較できるよう、公正で平等な条件で実施されるよう計画を立てる。また、実験開始後の計画変更はコストが大きすぎるため、実験計画の精査や実験用アプリケーションのデバッグは入念に行う。実験は改善手法の有効性を確認する貴重な機会であり、善し悪しに関わらず多くの気づきが得られる。実験の効果が最大化されるよう、先行研究や類似研究を参考にしながら、十分に計画を立てる。

本論文では、2章で構想したマイクロUX法の手順を、実際のITサービスを題材に3章4章5章で試行した。それぞれの題材において、UX課題の分解を行い、改善指標を定義し、改善手法の開発、評価実験の実施を行った。マイクロUX法の試行を振り返った結果、マイクロUX法の手順を用いて各題材に有効なUIの改善が行えたことが確認された。これらのことから、2章で構想したマイクロUX法の手順を用いることで、UX改善のコストとリスクが小さくなり、反復的な設計と検証を効率的に行える成果物の創出が行える可能性が示唆された。

6.2.2 ミクロUX法の成果物

本論文では、3章と4章で実際のITサービスの抱える問題に対しマイクロUX法を試行し、有効な改善手法を提案することができた。一方で、改善手法をサービスに導入する際に既存デザインやシステムの大規模な改修が必要になる可能性が高い問題が残った。そこで、3章と4章で考察された成果物に関する知見を元に、5章ではサービスへの導入コストを最小化するための成果物の形について検討を行った。成果物の形の検討を行うに当たり、5章ではスマートフォン用の地図サービスで屋内目的地への移動が難しい問題に対しマイクロUX法を試行し、SCANという手法を提案した。予備実験を行い基本的な操作性と機能性に大きな問題がないことを確認し、SCANのサービスへの導入コストを最小化する方法について検討を行った。5章では、日本国内で広く普及しているスマートフォン用の地図アプリケーションへの実装を試み、マイクロUX法で創出する成果物がどのような形であることが適切か検討を行った。

サービスに導入しやすい成果物の検討

ITサービスの開発現場でUX改善の実施が難しい要因の一つにコストの問題がある。これに対し、マイクロUX法はUX改善の実施コストを低減するアプローチを実現したが、UX改善の成果物の導入コストが大きくなる可能性について課題が残った。すなわち、ITサービスの開発現場でUX改善が実施しやすくするためには、マイクロUX法はUX改善の実践コストと成果物の導入コストの両方が小さくある必要がある。また、アジャイル開発の反復的な設計と検証を効率的に行うためには、マイクロUX法の成果物は組み込みやすく取り外ししやすい構造であることが求められる。

これらを踏まえ、5章では現在地を取得するカメラ機能と現在地を返すインタフェースを持つシンプルなパッケージであるSCAN SDKを開発し、実在の地図サービスのアプリケーションへ導入を試みた。SDKとすることで組み込みやすく取り外ししやすい構造を実現した。地図アプリケーションの既存の位置情報取得処理部を拡張し、位置情報が取得で

きない場合には GPS の代わりに SCAN が位置情報を提供する実装とした。既存システムを流用することで、改修範囲の最小化を実現した。また、地図アプリケーションの既存の現在地取得ボタンを拡張し、位置情報が取得できない場合には SCAN による現在地取得処理が実行される UI とした。ユーザにとってはいずれも現在地を表示するためのボタンであり、コンテキストに沿ったデザインを実現した。

5章で試みた実在の地図アプリケーションへの SCAN の導入を振り返り、マイクロ UX 法で創出する成果物ではシステム面とデザイン面で留意すべき点があると考えられる。

システム面 システム面では既存システムへの影響を最小限にするために高い独立性が求められる。サービスの処理の流れを大幅に変更したりサービスの機能と衝突しやすいシステムは、改修のコストだけでなく事故のリスクも増大させる。サービスの基本的な処理構造を正しく理解し、必要最低限のインタフェースで必要十分な機能を提供するシステムの設計が重要になる。マイクロ UX 法で創出する成果物のシステムは、サービスのシステムの構成を最大限再利用しつつ、提供する機能本体は単独で動作可能な状態であることが望ましい。

デザイン面 デザイン面では既存デザインのコンテキストを阻害しないために高い汎用性が求められる。サービスの既存デザインを大きく変更したり、操作の流れを大きく阻害する UI は、システム面同様にサービス全体のインタラクションの再設計が必要になる可能性が高い。サービスのデザインコンテキストを正しく読み取り、ユーザがサービスを利用する流れの中に自然に組み込まれるようなインタラクションの設計が重要になる。マイクロ UX 法で創出する成果物のデザインは、サービスのデザインルールやコンテキストに極力沿いつつ、他の要素に依存せずインタラクションが機能するデザインであることが望ましい。

3章4章5章の取り組みから、マイクロ UX 法で創出する成果物は、高い独立性を持つシステムと高い汎用性を持つデザインとすることで、サービスへの導入コストが低減される可能性があることが示唆された。

6.2.3 ミクロ UX 法の適用制限

本論文では、アジャイル開発の開発サイクルに並走可能な UX 改善のアプローチとしてマイクロ UX 法を構想した。実際の IT サービスを題材にマイクロ UX 法を試行し、サービスの抱える UX 課題を UI 課題に分解し改善手法を提案を実現した。実在の IT サービスアプ

リケーションに改善手法の導入を試み、成果物に独立性の高いシステム性と汎用性の高いデザイン性を持たせることで、サービスへの導入コストの低減を実現した。

一方で、本論文で試行した IT サービスの UX 課題の多くは、UI の操作性に関する改善が主なものであった。比較検証を行うためにタスク完了時間や操作回数、SUS スコアといった効率性を測る指標を多く用いたが、認知や行動といった異なる指標での比較評価については試行していない。操作の効率性は調査と比較が容易であり、小規模で短期間に UX 改善を実現するのに適した指標であるが、マイクロ UX 法の汎用性を高めるためには様々な指標を用いた評価方法の検討を行う必要がある。

また、本論文ではマイクロ UX 法で改善したマイクロ UX をサービスに導入し、サービスの KPI に貢献するアプローチについて検証していない。マイクロ UX 法はアジャイル開発の開発サイクルに並走可能な UX 改善のアプローチであり、反復的な設計と検証によってサービスの KPI への貢献を目指すものであるが、この反復的な設計と検証の効果的な実施方法について検討できていない。アジャイル開発の反復的な開発サイクルに合わせマイクロ UX 法も反復的に実施し、IT サービスの開発現場での効果的なマイクロ UX 法の実践方法について検証を行う必要がある。

これに関連し、本論文ではどのようなマイクロ UX の改善がどのような KPI に貢献するかについて検証が行えていない。アジャイル開発は設計と検証を小さく早く繰り返しながら提供すべき価値を見つけ出すアプローチであり、KPI への貢献が確認されるまでマイクロ UX の改善を繰り返すアプローチも考えられる。しかし、毎回ゼロベースで検証を繰り返すのは効率が悪い。様々なサービスでのマイクロ UX 法の実践を通し UX 改善の施策と効果をまとめ、マイクロ UX パーツの再利用性を高める必要がある。

第7章 まとめと将来的な展望

本論文では、アジャイル開発の開発サイクルに並走可能な UX 改善のアプローチ、マイクロ UX 法を構想し検証を行った。IT サービスの開発現場では、変化が激しく競争が厳しい環境に対応するために、より早くより効率的な開発サイクルが求められている。IT サービスの開発現場では、激しい変化に対応するためにアジャイル開発に取り組み、競合との差別化を図るために UX 改善に取り組んでいる。本論文では、アジャイル開発と UX 改善の特徴や手法について概観し、両者のアプローチは根本的に異なるため両立が難しい現状について指摘した。本論文ではこの点に着目し、UX 改善の対象の極小化と評価の単純化によってコストとリスクを小さくし、アジャイル開発の反復的开发サイクルに適用しやすい成果物の創出を実現する、マイクロ UX 法を構想した。

マイクロ UX 法は、サービスが抱える UX の課題を小さく分解し UI の課題にすることで、UX 改善の対象と範囲が極小化され UX 改善のコストが軽減される。また、UI 改善の有効性を定量的な指標を用いて評価することで、改善要因の明確化を図る。こうすることで UX 改善施策の効果が想定しやすくなり UX 改善のリスクが低減する。また、改善要因が明確化されることで改善施策と効果の因果関係が明らかになり、アジャイル開発の反復的な設計と検証を効果的に行えるようになる。本論文では、これらを実現するために必要な手順について検討し、実際のスマートフォン用の IT サービスを題材にマイクロ UX 法を試行し検証を行った。

マイクロ UX 法の検証では、スマートフォン用のメールサービスで文章作成が難しい問題、スマートフォン用の地図サービスで片手操作が難しい問題、スマートフォン用の地図サービスで屋内目的地への移動が難しい問題に対し、マイクロ UX 法を試行して検証を行った。メールサービスで文章作成が難しい問題では、キャレット操作が難しい問題に着目し改善を行った。タスク完了時間と SUS スコアを改善指標に、文字にキャレットを移動するのではなく文字をキャレットに移動させる手法、Fix and Slide を考案し、評価実験を行った。実験の結果、既存手法より有意に速く操作が行える手法であることが確認された。一方で、Fix and Slide をサービスに導入するには、サービスのシステムを大規模に改修しなくてはならない可能性が高い点が課題として残った。地図サービスで片手操作が難しい問題では、片手でのズーム操作が難しい問題に着目し改善を行った。タスク完了時間と操作回数、SUS スコアを改善指標に、ディスプレイを押す力に応じてズームアウトする手法、

Bounce Back と Force Zoom を考案し、評価実験を行った。実験の結果、既存手法より有意に少ない回数で操作が行える手法であることが確認された。一方で、Bounce Back と Force Zoom をサービスに導入するには、サービスのインタラクションデザインを全体的に見直さなくてはならない可能性が高い点が課題として残った。これらの課題を踏まえ、次に取り組む地図サービスで屋内目的地への移動が難しい問題では、改善手法の考案とサービス導入に適した形について検討を行った。まず、地図サービスで屋内目的地への移動が難しい問題では、屋内で現在地表示がされない問題に着目し改善を行った。総移動距離と総移動時間、SUS スコアを改善指標に、ユーザが撮影した屋内地図写真上に現在地を表示・追跡する手法、SCAN を考案した。予備実験の結果、基本的な操作性や機能性に大きな問題は確認されなかった。続いて、実在の地図サービスアプリケーションに SCAN の導入を試み、導入コストを低減する方法について検討した。前述の導入課題の考察から、マイクロ UX 法で創出する成果物は独立性の高いシステムと汎用性の高いデザインを持ち、組み込みやすく取り外しやすいシンプルな構造とすることで、導入コストの低減が行えると考えた。そこで SCAN を、現在地取得を行うカメラ機能と現在地算出を行うインタフェースを持つシンプルな SDK としてパッケージ化した。SCAN に関連する UI はカスタマイズ可能にし、サービス側で適用しやすいようデザインの余地を残した。地図アプリケーションに SCAN SDK を実装した結果、既存システムへの改修は最小限に既存デザインのコンテキストに沿った形で成果物の導入が実現できた。

総合的な議論を行うために、マイクロ UX 法の構想とマイクロ UX 法の試行結果を振り返った。実証を通じ、UX 課題を UI 課題に小さく分解することで、小規模で短期間に UX 改善が行えることが確認できた。また、定量的な改善指標を定義することで、UI 改善の要因を明確に評価できることが確認できた。UX 改善の対象を極小化し評価を単純化することで、目的に直結する改善手法が考案できた。マイクロ UX 法の試行を通じ、マイクロ UX 法で創出する成果物のサービス導入上の問題が判明したが、組み込みやすく取り外しやすいシンプルな構造を採用することでこの問題を回避できることが確認できた。

本論文では、IT サービスの開発現場でより早くより効率的な開発サイクルを実現するために、アジャイル開発の開発サイクルに並走可能な UX 改善のアプローチ、マイクロ UX 法を構想した。実際の IT サービスを題材にマイクロ UX 法を試行し、マイクロ UX 法の効果的な手順や成果物について示した。マイクロ UX 法は、アジャイル開発と並走可能な UX 改善のアプローチであるが、実際のサービスでの効果的な運用方法について検証されていない。マイクロ UX 法は、アジャイル開発の開発サイクルに並走可能な UX 改善のアプローチであり、反復的な設計と検証によってサービスへの貢献を目指すものであるが、改善を実施するたびにゼロベースで探るのは効率が悪い。今後の展望として、多種多様なサービスでマイクロ UX 法を実践し、UX 改善の施策と効果についてまとめ、UX パーツの再利用性

を高めたいと考えている。多くの IT サービスの開発現場でマイクロ UX 法が実践され、効果が明らかな UX パーツが広く普及することで、より効率的で合理的な UX デザインが実現できるよう、マイクロ UX 法の普及活動や発展研究を行いたい。そして、より良い体験が得られるサービスが1つでも多く生まれることに期待したい。

謝辞

本論文をまとめるにあたり、ヒューマンコンピュータインタラクション研究室の坂本大介准教授には、本研究の機会を与えていただき、長年に渡り御指導いただきました。また、社会人博士課程への入学と在学中の研究に関しても懇切なる御指導と御助言をいただき、学会誌への投稿および国際会議での発表など貴重な機会を与えていただきました。ここに深く感謝の意を表します。また、ヒューマンコンピュータインタラクション研究室の小野哲雄教授には、本研究の遂行において多角的な御助言をいただきました。本研究に行き詰まりそうになった時、新たな視点を持ち臨むことができました。心より感謝申し上げます。

副指導教員として御指導いただきました知能ソフトウェア研究室の野田五十樹教授、自律系工学研究室の山本雅人教授、調和系工学研究室の川村秀憲教授にも、厚く御礼申し上げます。本研究を構成する各研究に共に取り組んで下さいました共著者の皆様にも、深く感謝しております。社会人博士課程進学を御支援いただき、本研究を行う機会を与えていただきましたヤフー株式会社にも、厚く御礼申し上げます。本研究への様々な御助言や御支援をいただきましたヤフー株式会社の同僚の皆様、北海道大学と多摩美術大学の学友の皆様、そして家族に、心より感謝申し上げます。本当にありがとうございました。

参考文献

- [1] Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, and Dave Thomas. Manifesto for agile software development. <https://agilemanifesto.org/iso/ja/manifesto.html>, 2001.
- [2] Donald A Norman. *The invisible computer: why good products can fail, the personal computer is so complex, and information appliances are the solution*. MIT press, 1998.
- [3] Virpi Roto, Effie Lai-Chong Law, Arnold Vermeeren, and Jettie Hoonhout. 10373 Abstracts Collection – Demarcating User eXperience. In Jettie Hoonhout, Effie Lai-Chong Law, Virpi Roto, and Arnold Vermeeren, editors, *Demarcating User eXperience*, No. 10373 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2011. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany.
- [4] Jesse James Garrett. *The Elements of User Experience: User-Centered Design for the Web and Beyond*. New Riders Publishing, USA, 2nd edition, 2010.
- [5] Daniel Vogel and Patrick Baudisch. Shift: a technique for operating pen-based interfaces using touch. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 657–666, 2007.
- [6] Kent Beck. *Extreme programming explained: embrace change*. addison-wesley professional, 2000.
- [7] Ken Schwaber and Jeff Sutherland. The scrum guide. *Scrum Alliance*, Vol. 21, No. 1, 2011.
- [8] 浅野志帆, 安藤昌也, 赤澤智津子. Ka 法を初心者が理解・実践するための研究. 日本デザイン学会研究発表大会概要集, Vol. 63, p. 229, 2016.

- [9] John Whiteside, John Bennett, and Karen Holtzblatt. Chapter 36 - usability engineering: Our experience and evolution. In MARTIN HELANDER, editor, *Handbook of Human-Computer Interaction*, pp. 791–817. North-Holland, Amsterdam, 1988.
- [10] Rex Hartson and Pardha S Pyla. *The UX Book: Process and guidelines for ensuring a quality user experience*. Elsevier, 2012.
- [11] 総務省. 平成 26 年通信利用動向調査. <http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h27/html/nc372110.html>, 2015.
- [12] Kenji Suzuki, Kazumasa Okabe, Ryuuki Sakamoto, and Daisuke Sakamoto. Fix and slide: Caret navigation with movable background. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services*, pp. 478–482, 2016.
- [13] Anne Roudaut, Stéphane Huot, and Eric Lecolinet. Taptap and magstick: improving one-handed target acquisition on small touch-screens. In *Proceedings of the working conference on Advanced visual interfaces*, pp. 146–153, 2008.
- [14] Andrew Sears and Ben Shneiderman. High precision touchscreens: design strategies and comparisons with a mouse. *International Journal of Man-Machine Studies*, Vol. 34, No. 4, pp. 593–613, 1991.
- [15] Daniel Wigdor, Clifton Forlines, Patrick Baudisch, John Barnwell, and Chia Shen. Lucid touch: a see-through mobile device. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*, pp. 269–278, 2007.
- [16] Khalad Hasan, Xing-Dong Yang, Hai-Ning Liang, and Pourang Irani. How to position the cursor? an exploration of absolute and relative cursor positioning for back-of-device input. In *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services*, pp. 103–112, 2012.
- [17] Christian Holz and Patrick Baudisch. The generalized perceived input point model and how to double touch accuracy by extracting fingerprints. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 581–590, 2010.
- [18] Xiaojun Bi and Shumin Zhai. Bayesian touch: a statistical criterion of target selection with finger touch. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*, pp. 51–60, 2013.

- [19] Hrvoje Benko, Andrew D Wilson, and Patrick Baudisch. Precise selection techniques for multi-touch screens. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pp. 1263–1272, 2006.
- [20] Katrin Wolf and Niels Henze. Comparing pointing techniques for grasping hands on tablets. In *Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services*, pp. 53–62, 2014.
- [21] Alexander Keith Eady and Audrey Girouard. Caret manipulation using deformable input in mobile devices. In *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction*, pp. 587–591, 2015.
- [22] 三浦元喜, 最所賢至, 清弘祥太. タブレット端末における日本語形態素を考慮した文字列範囲選択手法. *Proc. WISS*, pp. 121–122, 2014.
- [23] Jean-Baptiste Scheibel, Cyril Pierson, Benoît Martin, Nathan Godard, Vittorio Fuccella, and Poika Isokoski. Virtual stick in caret positioning on touch screens. In *Proceedings of the 25th Conference on l’Interaction Homme-Machine*, pp. 107–114, 2013.
- [24] Vittorio Fuccella, Poika Isokoski, and Benoit Martin. Gestures and widgets: performance in text editing on multi-touch capable mobile devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 2785–2794, 2013.
- [25] Apple Inc. ios human interface guidelines: 色とタイポグラフィ. <https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/ColorImagesText.html>.
- [26] Lifehacker. Five shortcuts for faster text selection in ios. <http://lifehacker.com/five-shortcuts-for-faster-text-selection-in-ios-1641490708>, 2015.
- [27] Patrick W Jordan, Bruce Thomas, Ian Lyall McClelland, and Bernard Weerdmeester. *Usability evaluation in industry*. CRC Press, 1996.
- [28] Takashi Miyaki and Jun Rekimoto. Graspzoom: Zooming and scrolling control model for single-handed mobile interaction. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services, MobileHCI ’09*, New York, NY, USA, 2009. Association for Computing Machinery.

- [29] Jun Rekimoto and Carsten Schwesig. Presenseii: Bi-directional touch and pressure sensing interactions with tactile feedback. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '06, p. 1253–1258, New York, NY, USA, 2006. Association for Computing Machinery.
- [30] Daniel Spelmezan, Caroline Appert, Olivier Chapuis, and Emmanuel Pietriga. Side pressure for bidirectional navigation on small devices. In *Proceedings of the 15th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '13, p. 11–20, New York, NY, USA, 2013. Association for Computing Machinery.
- [31] Seongkook Heo and Geehyuk Lee. Forcetap: Extending the input vocabulary of mobile touch screens by adding tap gestures. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '11, p. 113–122, New York, NY, USA, 2011. Association for Computing Machinery.
- [32] Seongkook Heo and Geehyuk Lee. Force gestures: Augmenting touch screen gestures with normal and tangential forces. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, p. 621–626, New York, NY, USA, 2011. Association for Computing Machinery.
- [33] Chris Harrison and Scott Hudson. Using shear as a supplemental two-dimensional input channel for rich touchscreen interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, p. 3149–3152, New York, NY, USA, 2012. Association for Computing Machinery.
- [34] EC Clarkson, S Patel, J Pierce, and GD Abowd. Exploring continuous pressure input for mobile phones. gvu techreport. Technical report, GIT-GVU-06-20, Georgia Inst. of Tech, 2006.
- [35] Craig Stewart, Michael Rohs, Sven Kratz, and Georg Essl. Characteristics of pressure-based input for mobile devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, p. 801–810, New York, NY, USA, 2010. Association for Computing Machinery.
- [36] Ryosuke Takada, Wei Lin, Toshiyuki Ando, Buntarou Shizuki, and Shin Takahashi. A technique for touch force sensing using a waterproof device's built-in barometer.

In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '17, p. 2140–2146, New York, NY, USA, 2017. Association for Computing Machinery.

- [37] Axel Antoine, Sylvain Malacria, and Géry Casiez. Forceedge: Controlling autoscroll on both desktop and mobile computers using the force. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, p. 3281–3292, New York, NY, USA, 2017. Association for Computing Machinery.
- [38] Christian Corsten, Simon Voelker, Andreas Link, and Jan Borchers. *Use the Force Picker, Luke: Space-Efficient Value Input on Force-Sensitive Mobile Touchscreens*, p. 1–12. Association for Computing Machinery, New York, NY, USA, 2018.
- [39] Aaron Bangor, Philip Kortum, and James Miller. Determining what individual sus scores mean: Adding an adjective rating scale. *Journal of usability studies*, Vol. 4, No. 3, pp. 114–123, 2009.
- [40] Anne Roudaut, Stéphane Huot, and Eric Lecolinet. Taptap and magstick: Improving one-handed target acquisition on small touch-screens. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '08, p. 146–153, New York, NY, USA, 2008. Association for Computing Machinery.
- [41] Jun Rekimoto. Tilting operations for small screen interfaces. In *Proceedings of the 9th Annual ACM Symposium on User Interface Software and Technology*, UIST '96, p. 167–168, New York, NY, USA, 1996. Association for Computing Machinery.
- [42] Ken Hinckley and Hyunyoung Song. Sensor synaesthesia: Touch in motion, and motion in touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, p. 801–810, New York, NY, USA, 2011. Association for Computing Machinery.
- [43] David Holman, Andreas Hollatz, Amartya Banerjee, and Roel Vertegaal. Unifone: Designing for auxiliary finger input in one-handed mobile interactions. In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction*, TEI '13, p. 177–184, New York, NY, USA, 2013. Association for Computing Machinery.
- [44] Sven Kratz, Ivo Brodien, and Michael Rohs. Semi-automatic zooming for mobile map navigation. In *Proceedings of the 12th International Conference on Human*

Computer Interaction with Mobile Devices and Services, MobileHCI '10, p. 63–72, New York, NY, USA, 2010. Association for Computing Machinery.

- [45] Pekka Parhi, Amy K. Karlson, and Benjamin B. Bederson. Target size study for one-handed thumb use on small touchscreen devices. In *Proceedings of the 8th Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '06, p. 203–210, New York, NY, USA, 2006. Association for Computing Machinery.
- [46] Keith B. Perry and Juan Pablo Hourcade. Evaluating one handed thumb tapping on mobile touchscreen devices. In *Proceedings of Graphics Interface 2008*, GI '08, p. 57–64, 2008.
- [47] Sebastian Boring, David Ledo, Xiang 'Anthony' Chen, Nicolai Marquardt, Anthony Tang, and Saul Greenberg. The Fat Thumb: Using the Thumb's Contact Size for Single-Handed Mobile Interaction. In *Proceedings of the 14th International Conference on Human-Computer Interaction with Mobile Devices and Services Companion*, MobileHCI '12, p. 207–208, New York, NY, USA, 2012. Association for Computing Machinery.
- [48] ウェブレッジ. スマートフォン・シェアランキング (top10). https://webrage.jp/techblog/sp_share/, 2019.
- [49] DeviceAtlas. <https://deviceatlas.com/blog/most-popular-smartphones#japan>, 2019.
- [50] Sylvain Malacria, Eric Lecolinet, and Yves Guiard. Clutch-free panning and integrated pan-zoom control on touch-sensitive surfaces: The cyclostar approach. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, p. 2615–2624, New York, NY, USA, 2010. Association for Computing Machinery.
- [51] John Brooke, et al. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, Vol. 189, No. 194, pp. 4–7, 1996.
- [52] Jó Ágila Bitsch Link, Paul Smith, Nicolai Viol, and Klaus Wehrle. Footpath: Accurate map-based indoor navigation using smartphones. In *2011 International Conference on Indoor Positioning and Indoor Navigation*, pp. 1–8, Sep. 2011.

- [53] OpenStreetMap contributors. Planet dump retrieved from <https://planet.osm.org> . <https://www.openstreetmap.org>, 2017.
- [54] Alessandro Mulloni, Hartmut Seichter, and Dieter Schmalstieg. Handheld augmented reality indoor navigation with activity-based instructions. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services, MobileHCI '11*, p. 211–220, New York, NY, USA, 2011. Association for Computing Machinery.
- [55] Jongbae Kim and Heesung Jun. Vision-based location positioning using augmented reality for indoor navigation. *IEEE Transactions on Consumer Electronics*, Vol. 54, No. 3, pp. 954–962, August 2008.
- [56] Busra Ozdenizci, Kerem Ok, Vedat Coskun, and Mehmet N. Aydin. Development of an indoor navigation system using nfc technology. In *2011 Fourth International Conference on Information and Computing*, pp. 11–14, April 2011.
- [57] Yao-Jen Chang, Shih-Kai Tsai, Yao-Sheng Chang, and Tsen-Yung Wang. A novel wayfinding system based on geo-coded qr codes for individuals with cognitive impairments. In *Proceedings of the 9th International ACM SIGACCESS Conference on Computers and Accessibility, Assets '07*, p. 231–232, New York, NY, USA, 2007. Association for Computing Machinery.
- [58] Paramvir Bahl and Venkata N Padmanabhan. Radar: an in-building rf-based user location and tracking system. In *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064)*, Vol. 2, pp. 775–784 vol.2, March 2000.
- [59] Martin Werner, Moritz Kessel, and Chadly Marouane. Indoor positioning using smartphone camera. In *2011 International Conference on Indoor Positioning and Indoor Navigation*, pp. 1–6, Sep. 2011.
- [60] M. Betke and L. Gurvits. Mobile robot localization using landmarks. *IEEE Transactions on Robotics and Automation*, Vol. 13, No. 2, pp. 251–263, April 1997.
- [61] Richard Hartley and Andrew Zisserman. Multiple view geometry in computer vision. Technical report, 2003.

付録 A 研究業績リスト

査読付学会誌論文

1. 鈴木 健司, 坂本 竜基, 坂本 大介, 小野 哲雄: 片手での小型携帯端末操作のための圧力情報を利用したズームアウト操作インタフェース, 情報処理学会論文誌, Vol.61, No.2, pp.233-243 (2020).
2. 鈴木 健司, 岡部 和昌, 坂本 竜基, 坂本 大介: Fix and Slide: 文字列全体の移動によるキャレットのポインティング手法, 情報処理学会論文誌, Vol.60, No.2, pp.354-363 (2019).

査読付国際会議プロシーディング

1. Kenji Suzuki, Ryuuki Sakamoto, Daisuke Sakamoto, and Tetsuo Ono. 2018. Pressure-sensitive zooming-out interfaces for one-handed mobile interaction. In Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '18). Association for Computing Machinery, New York, NY, USA, Article 30, 1–8.
2. Kenji Suzuki, Kazumasa Okabe, Ryuuki Sakamoto, and Daisuke Sakamoto. 2016. Fix and slide: caret navigation with movable background. In Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '16). Association for Computing Machinery, New York, NY, USA, 478–482.

国際会議デモ・ポスター

1. Kenji Suzuki, Daisuke Sakamoto, Sakiko Nishi, and Tetsuo Ono. 2019. SCAN: Indoor Navigation Interface on a User-Scanned Indoor Map. In Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '19). Association for Computing Machinery,

New York, NY, USA, Article 66, 1–6.

2. Kenji Suzuki, Kazumasa Okabe, Ryuuki Sakamoto, and Daisuke Sakamoto. 2015. Fix and Slide: Caret Navigation with Movable Background. In Adjunct Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15 Adjunct). Association for Computing Machinery, New York, NY, USA, 79–80.

その他発表論文

1. 鈴木健司, 岡部和昌, 坂本竜基, 坂本大介: Fix and Slide: 指による遮蔽を回避する相対的なポインティング手法, 研究報告グループウェアとネットワークサービス (GN), 2017-GN-100, 21, pp.1-8, (2017).
2. 鈴木 健司, 岡部 和昌, 坂本 竜基, 坂本 大介: Fix and Slide: 文字列全体を操作することによるキャレットの操作手法, インタラクティブシステムとソフトウェアに関するワークショップ (WISS2015), 日本ソフトウェア科学会, pp.73-78, (2015).
3. 鈴木 健司, 岡部 和昌, 坂本 竜基, 坂本 大介, テキスト全体の移動によりキャレットの相対位置を変化させるポインティング手法の提案, インタラクシオン 2015, 情報処理学会, N55, pp.638-640, (2015).

表彰等

1. 令和元年10月 論文「SCAN: Indoor Navigation Interface on a User-Scanned Indoor Map」により, アメリカ Association for Computing Machinery から「MobileHCI '19 Best Demo Award」を受賞
2. 平成27年11月 論文「Fix and Slide: Caret Navigation with Movable Background」により, アメリカ Association for Computing Machinery から「UIST '15 Best Poster Award Honorable Mention」を受賞
3. 平成27年3月 論文「テキスト全体の移動によりキャレットの相対位置を変化させるポインティング手法の提案」により, 情報処理学会より「インタラクシオン 2015 インタラクティブ発表賞」を受賞

目 次

2.1	アジャイルソフトウェア開発宣言 (アジャイルマニフェスト) [1].	7
2.2	XP (eXtream Programing) の概念図. (この画像は, クリエイティブ・コ モンズ・表示・継承ライセンス 3.0 のもとで公表されたウィキメディアの 画像「File:Extreme Programming.svg」を素材として二次利用しています.)	8
2.3	スクラムの概念図.	10
2.4	ユーザ体験の時間構造 [3].	12
2.5	ユーザ体験の階層構造 [4].	13
2.6	マイクロ UX 法の手順.	19
2.7	アジャイル開発と並走する UX 改善のイメージ.	21
3.1	一般的なキャレット操作手法 (左) と提案手法 (右): ユーザは文字列全体 を背景としてドラッグすることで, 任意の場所にキャレットを移動するこ とができる.	24
3.2	提案手法の概要. キャレット移動 (上段) と文字列選択 (下段). タッチを起点 に, キャレットを固定 (Fix) し, 背景文字列を移動 (Slide) することで, キャ レット位置を操作できる.	25
3.3	実験で使用するフォントサイズ 3 種: 12 ポイント (上段), 14 ポイント (中 段), 16 ポイント (下段). 中段の図は文字列の部分選択条件の例を示し, 下 段は外側選択条件の例を示している.	28
3.4	3 要因分散分析の結果と標準誤差	31
3.5	操作手法×文字列選択の位置の結果と標準誤差	32
3.6	SUS の結果と標準誤差	33
3.7	文字列選択範囲を指示する紙面の例	35
3.8	実験参加者と実験の様子	36
4.1	Force Zoom: 指が画面を押す力を計測し, 押す力の強さに比例した速度で ズームアウトする.	41
4.2	Bounce Back (上段) と Force Zoom (下段) の概要. 画面を押す圧力を用 いズームアウト操作を行う.	43

4.3	GraspZoom の概要.	47
4.4	作業実行時間の結果	49
4.5	操作回数の結果	50
4.6	SUS の結果	51
4.7	ズームアウト単独セッションにおける SUS の個別の質問項目の結果	52
4.8	ズームアウトとズームイン複合セッションにおける SUS の個別の質問項目の結果	53
5.1	SCAN：ユーザが撮影した屋内地図の画像上に自身の現在地と向いている方向を表示する.	59
5.2	ユーザの移動は IMU によって計測され、撮影した屋内地図の画像上の現在地を更新する.	60
5.3	地図上の現在位置の再補正方法. ランドマークを撮影し現在地の再計算を行う.	60
5.4	SCAN の概観	62
5.5	事前データの収集方法	64
5.6	ユーザ位置算出の仕組み	65
5.7	予備実験のスタート地点. この屋内地図を撮影した SCAN を使用しながら、指定された部屋へ移動を開始する.	67
5.8	SCAN の発展手法の概要.	69
5.9	発展手法における現在地測定手法. ランドマークを撮影し現在地を算出する.	70
5.10	発展手法における現在地補正手法. 位置補正モードに移行し、指で現在地を正しい位置に移動させる.	71
5.11	SCAN を拡張し実装した Yahoo!地図アプリの画面. 赤丸の部分が SCAN が提供する UI パーツ.	72

表 目 次

3.1	文字列選択の位置	29
4.1	作業実行時間の結果と標準誤差. 小さい値ほど良い.	49
4.2	操作回数の結果と標準誤差. 小さい値ほど良い.	50
4.3	SUS の結果と標準誤差. 大きい値ほど良い.	51