

愛知工業大学情報科学部情報科学科
コンピュータシステム専攻（メディア情報専攻）

令和2年度 卒業論文

独立したコミュニティにおける
滞在ウォッチの安定運用のための
システム拡張に関する研究

2020年2月

研究者 K19074 外山瑠起
K00011 八草花子

指導教員 情報一郎 教授

目 次

第1章 はじめに

本章では、研究背景や我々の先行研究について述べ、本研究の目的と概要を説明する。

1.1 背景

近年、社会的に重要な課題の一つとなっているのが、在室者管理の課題である。これは、環境保護やエネルギー効率化、セキュリティ強化などの目的から、家庭やオフィスなどの建物内での生活やビジネスに関する様々なアプリケーションに応用されるためである。例えば、エネルギー管理は、居住者がいる場合といない場合でエネルギー消費量が大きく異なるため、在室者管理を使用して、建物内でのエネルギー消費を最適化できる。また、セキュリティや照明などのシステムの自動化にも応用され、これにより、環境保護やエネルギー効率化を図れる。在室者管理は、災害時や緊急事態においても重要な役割を持つ。災害時には、避難した居住者を確認が重要であり、在室者管理を使用すれば、確認作業をスムーズに行える。

また在室者管理ができれば、コミュニティにおいても様々なメリットがある。その一つに、コミュニケーション促進がある。在室者管理ができれば、コミュニティ内で誰がいるのかを確認できる。これにより、コミュニティ内での交流や活動がスムーズになり、コミュニケーションの促進を行える。例えば、共同生活をする人たちが同じ時間にいる場合には、共同での食事や過ごし方を提案でき、コミュニケーションを促進できる。また、コミュニティ内でのイベントやミーティングの開催タイミングを調整できる。これにより、参加者が多い時間帯に開催できるため、参加者が集まりやすくなり、コミュニケーションの促進を行える。

また近年は新型コロナウイルスの影響により、感染拡大を防ぐ上で有効な手段と考えられている。在室者管理を使用すれば建物内における人流管理が可能になる。これにより、感染者がいる可能性が高い時間帯や場所を特定し、それに対して対策を行える。例えば、感染者がいる可能性のある時間帯には、入退室を制限できる。また、感染者がいた場所を特定ができる。これにより、感染者が訪れた場所の消毒を行えるため、感染拡大を防げる。さらに、建物内における感染者との接触者を特定ができる。これにより、感染者との接触者に対して、隔離や検査を行えるため、早期発見・早期対応ができるようになる。そのため、病院や医療機関などでは、在室者管理システムを導入し、患者やスタッフの感染リスクを低減するために活用されている。

在室者管理の研究は、学术界や産業界においても注目を集めており、20世紀後半から様々な方法が提案されてきた。近年では、様々なセンサを使用した方法が提案されている。例えば、照度センサや温度センサ、音声センサ、カメラなどが使用される。これらセンサを使用して得られたデータを処理し、居住者がいるかどうかを判定する。ただし、これらのセンサを使用した在室者管理は、環境条件や居住者の生活スタイルなどにより精度が異なるため、正確な在室者管理ができない場合がある。また、プライバシー保護の観点から、カメラを使用は避けられる場合が多い。このように、在室者管理は重要な課題であり、様々なアプリケーションに応用されるが、環境やプライバシーにより、正確な検出が難しいという問題もある。他の在室者管理に関する方法として深層学習を使用した在室者管理がある。深層学習は、大量のデータを学習し、高い精度を得られるため、在室者管理においても有効であると考えられている。また、在室者管理においては、プライバシー保護も重要な観点である。深層学習を使用すれば、画像や音声などのデータの特徴量として抽出し、学習を行うため、プライバシーに敏感な情報を取り込まない。しかしながら、深層学習を使用した在室者管理には、大量のデータが必要であり、データ収集や学習には多くのリソースが必要なのが問題点である。

我々の先行研究として BLE(Bluetooth Low Energy) ビーコン (以下, ビーコン) を用いた在室管理プラットフォーム「滞在ウォッチ」を提案しているビーコンは低コストであり, 普及しやすいと考えられる。これにより, 建物全体に導入が容易であり, 在室者検出のカバー範囲を広げられる。また, ビーコンは低消費電力であり, 長期間にわたって使用できる。これにより, 在室者検出を 24 時間行えるため, 建物内での生活やビジネスに対して常時モニタリングを行える。さらに, ビーコンはプライバシー保護にも優れている。ビーコンは個人を特定できないため, 個人情報の保護にも適している。滞在ウォッチは, ビーコンを持ち歩き在室情報を記録するメンバ, 現在状況や履歴を閲覧したり API を通して利用する利用者, メンバ管理, メンバへのビーコン配布, 利用者の登録を行う管理者, システムを開発する開発者が存在する。メンバの負担軽減のため, 室者情報をビーコンで受動的に記録する方法を採用している。ビーコンを持った利用者が部屋に訪れると受信機が検知し, サーバに在室者情報を送信し, データベースに記録する。データベースに保存された情報は, 独自に作成した Web API によって外部からの利用が可能である。先行研究として, Web API を使用した退勤管理システムや在室情報可視化システム, 部屋利用者の来訪促進システム, コミュニケーションプラットフォームなどがある。

1.2 滞在ウォッチの安定運用のためのシステム拡張

本研究の目的は, 滞在ウォッチによる複数コミュニティ間連携の実現である。複数コミュニティ間とは空間的な距離が近いコミュニティスペース間と定義する。例として, 大学の研究室間やビルのオフィス間などが挙げられる。滞在ウォッチにおける複数コミュニティ間連携によって得られる利点として, まずコミュニティ間でのコミュニケーションの増加が挙げられる。普段コミュニケーションを取らない別コミュニティの人々とのコミュニケーションが促進される。次に共有や, 生産性の向上が挙げられる。他のコミュニティから新しい知見やアイデアを共有することができるため, 想像量や創造性が向上する, スムーズなスムーズなコミュニケーションやリソースを共有できるため, 生産性が向上する可能性がある。またコミュニティ間のコラボレーションが容易になるため, 新しいプロジェクトやイニシアティブが生まれる可能性がある。

しかし滞在ウォッチを複数コミュニティ間で連携するには大きく分けて 2 つの問題点が存在する。まずコミュニティで独立した運用ができていない点が挙げられる。「独立した運用」とは, 各コミュニティで自らのシステムや装置を運用できることを指している。複数コミュニティ間で連携するためには, 各コミュニティで滞在ウォッチを運用するために必要な設備やシステムを持つ必要がある。それぞれのコミュニティが独立したシステムを持っているいれば, 各コミュニティが独自に滞在情報を収集し, 管理することができるため, それらを連携させられる。しかし既存の滞在ウォッチは単一コミュニティを前提として設計されているため各コミュニティが独立した運用を行えない。次に在室情報を長期に渡り継続的に記録できない点が挙げられる。在室情報の長期的な記録は, コミュニティ間でのコミュニケーションを促進するために必要である。既存の滞在ウォッチで使用される BLE ビーコンのみを使用して運用しているため電池がなくなると在室情報を記録できない。これらの問題を解決した滞在ウォッチの運用を安定運用と定義する。本研究では滞在ウォッチを複数コミュニティ間で連携するために, 独立したコミュニティにおける滞在ウォッチの安定運用のためのシステム拡張について提案する

1.3 論文構成

第 2 章では, 本研究と関連した研究との違いを比較する。第 3 章では, 在室管理プラットフォーム「滞在ウォッチ」について述べる。第 4 章では, コミュニケーションの機会損失を軽減するシステム「きょうの滞在」について述べる。第 5 章では本研究に対するまとめと今後の課題について述べる。

第2章 関連研究

本章では，本研究と関連した研究との違いを比較する．2.1 節では部屋における在室者検出方法，無線通信技術による検出手法の比較を述べる．2.2 節では，部屋におけるスマートフォンやビーコン，IC カードを用いた在室者の検出方法の関連研究について述べる．2.3 節では記録された在室者情報の提示方法の関連研究について述べる．2.4 節ではコミュニケーション促進の関連研究について述べる．

表 2.1: 測位技術の比較

測位技術	屋外測位	屋内測位	消費電力
GPS	○	×	高い
携帯電話基地局	△	×	普通
Wi-Fi	△	○	普通
BLE	×	◎	低い

2.1 屋内位置推定における在室者の検出方法

以前から在室管理は自動化されれば便利なシステムになると言われていた [?]. 部屋における屋内位置推定にはいくつかの在室者の検出方法があり，用途によって人の在否のみと個人を特定する方法がある．滞在ウォッチでは，利用者が目的とする人の居場所を把握できるように，個人を特定する方法に着目する．具体的には IC カードやライブカメラを用いた検出方法があり，これらには導入時に配線工事の手間や高価な機材を必要とするため困難であるのとプライバシーへの配慮が必要である．そこで，自動で在室者情報を記録する無線通信技術による検出方法に着目する．志毛らの BLE を用いた位置情報共有システムの開発 [?] では，表 2.1 の無線通信による測位技術の比較を行った．消費電力が低く，屋内位置推定に向いている BLE を用いた検出方法が有効だと考えた．

2.2 在室者の検出方法に関する研究

大学や会社では在室者を検出する手法を用いて，講義の出欠 [?][?][?][?] や勤怠管理 [?] が行われている．在室者を検出する手法として IC カードを用いた検出方法 [?] [?] [?] [?] [?] [?] や，ビーコンの受信電波強度を利用した検出方法 [?][?] がある．

IC カードを用いた在室者検出方法では利用者に IC カードを携帯してもらい，専用の機器などを用いて在室者を検出する手法である．必要となる機器を導入した後は利用者は IC カードを用いるだけで在室者検出ができる．しかし新たに導入する場合は必要となる機器や，IC カードを登録する手間などコストは高くなってしまう．

ビーコンの受信電波強度を用いた在室者検出方法には二つある．1 つ目は利用者にビーコンを携帯してもらい，在室者を検出したい部屋に受信機を置く方法である．部屋利用者の在室者検出はビーコンの受信電波を受信機が取得するだけで行えるので，自動で行える．また，ビーコンはサイズが小さいものが多く，利用者が携帯する負担もかからない．在室者検出する部屋に受信機を置き，利用者はビーコンを携帯するだけなのでコストも抑えられる．2 つ目に利用者にはスマホを携帯してもらい，在室者を検出したい

部屋にビーコンを置く方法である。スマホの普及が進んでいる [?] ので、導入コストはビーコンのみであるため、1つ目の方法よりもコストを抑えられる。しかし、全員がスマホを所持しているわけではないので、スマホを所持していない利用者には別の検出方法を導入する必要がある。本研究では利用者が自発的に在室者情報を記録する手間を必要としない方法として、部屋ごとに受信機を設置し、個人がビーコンを携帯し自動で検出する方法を採用する。

在室者を検出し、在室者情報を管理するシステムに関する研究がある [?][?][?]。スマホを用いて在室者を検出し、在室者情報を管理するシステムは、スマホを所持している人が多いため、新たに必要となる機器の数が少なく、コストを抑えられる [?]。またビーコンを用いて在室者を検出し、在室者情報を管理するシステムは利用者にビーコンを携帯してもらい、在室者を検出したい部屋に受信機を設置すれば在室者を検出できるので、コストを抑えられる [?][?]。

本研究ではスマートフォンを所持していない利用者も想定し、利用者一人一人にビーコンを携帯してもらい、在室者を検出したい部屋に受信機を設置する方法を採用した。

2.3 在室者状況の提示方法に関する研究

在室者を検出した後に在室者情報を用いた在室者状況の提示方法は様々ある [?] [?] [?] [?] [?]。利用者全員が見られるサイネージに在室者情報を提示するものや、個人が所有しているスマートフォンやタブレットから在室状況を確認できるものなどがある。サイネージに提示する手法を用いた研究を図??[?]，図??[?] に示す。利用者全員が見られるサイネージに提示する手法では、一目で在室情報を確認できる必要がある。スマートフォンやタブレットに提示する手法を用いたものを図?? に示す。また個人が所有しているスマートフォンやタブレットから在室状況を確認できる手法ではスマートフォンやタブレットに適するレイアウトを考える必要がある。本研究ではその場にいる人のコミュニケーション促進するために、その場にいる人が見られるサイネージに提示する手法を採用した。

2.4 コミュニケーション促進に関する研究

第三者間のコミュニケーションや知っている人同士のコミュニケーションを支援、促進する研究がある。まず見知らぬ他人や顔だけは知っているがコミュニケーションを取らない人とのコミュニケーションを支援促進するものがある [?][?][?][?]。気軽に他者とのコミュニケーションができると示唆している。しかし知っている人同士のコミュニケーションを促進するものではなく、あくまで見知らぬ他者やコミュニケーションを取らない人とのコミュニケーションを支援、促進するものである。また公共空間でのコミュニケーションを支援、促進する研究がある [?][?][?][?]。見知らぬ人や顔だけは知っている関係ではなく、同じ空間を共有している人同士のコミュニケーションを支援、促進するものである。コミュニケーションのきっかけや共有できる情報を提示するものとして、本研究でも同じ空間を共有している人同士のコミュニケーションを促進する方法として参考にする。

第3章 4.3 独立したコミュニティにおける滞在ウォッチの安定運用のためのシステム拡張

滞在ウォッチを複数コミュニティ間で運用するには、独立したコミュニティで運用できない、長期に渡り在室情報を継続的に記録できない問題を解決した安定運用を行う必要があり、それに向けた取り組みについて述べる。4.1章で既存の滞在ウォッチに独立運用するためのシステムの再構築について述べる。4.2章で利用者の管理とアクセス制御システムの整備について述べる。4.3章でスマホアプリによるビーコンと実デバイスの併用について述べる。

4.1 安定運用に向けた既存システムの再構築

既存の滞在ウォッチは長期に渡り継続的に保守・開発していくのが困難だったため安定運用を目指す上でシステムの再構築を行った。

4.1.1 サーバの設計と構築

サーバ側には独立したバックエンドシステムとの連携が容易であり Web アプリケーションがより高い可用性とスケーラビリティを実現できる REST API を採用した。REST API は、複数のクライアントからアクセスができるため、様々なデバイスやプラットフォームからアクセスできる。これにより、より広いユーザー層からアクセスができる。既存の滞在ウォッチのサーバ側のシステムは Python を用いて構築されていた。Web API は大量の処理をする必要があり、Python の実行速度の遅さが、高負荷環境でパフォーマンスに影響を与える可能性があった。またデフォルトで非同期プログラミングをサポートしていないため、非同期処理に対応するために追加のライブラリを使用する必要がある。また動的型付け言語なため、型の不明確さによってバグが見つけづらく保守性が低い。そこで静的型付けであり、高速な処理能力と小さなメモリフットプリントを持つため、Web API の開発に適している Golang を採用した。Golang は高いパフォーマンスを持つ。C 言語のような低レベルな言語と同等のパフォーマンスを持ちながら、高水準の言語のような簡単な構文を持っているのに加え、並列処理を容易に実現できるため、高負荷な環境での Web API の開発にも適している。そのため WebAPI の要求に対応するために必要なパフォーマンスを提供することができる。また構造体とインターフェース型を備えており、WebAPI の開発に必要な柔軟性を持っている。これにより、開発者は、WebAPI を実装するために適した方法を選択することができる。さらに、標準パッケージによる HTTP サーバのサポートを持つため、Web API の開発に必要な機能を簡単に実装できる。サーバで使用するデータベースには無料で利用することができ、高いスケーラビリティを持つため、WebAPI の要求に対応することができる MySQL を採用した。MySQL は多くのプラットフォームでサポートされており、多くの言語に対応しており、開発者にとって選択肢が広がる。また MySQL は優れたセキュリティ機能を持つため、WebAPI のセキュリティの確保ができる。Golang で Mysql の操作を行う際 Golang の ORM ライブラリである Gorm を採用した。ORM はデータベースを操作を行うための手法の一つである。これにより、データベース操作を行う際に、SQL 文を直接記述する必要がなくなり、コードがすっきりし、可読性が高くなる。SQL 文を直接記述しないため SQL インジェクション対策になる。SQL インジェクションとは不正な SQL 文の挿入を行い、データベースに対して攻撃を行う手法である。ORM は、プログラマが入力したデータを自動的にエスケープし、サイバー攻撃に

つながるような文字を無効化するため SQL インジェクション攻撃を防止できる。Gorm は Golang 向け ORM ライブラリの一つである。Gorm はデータベースの挿入、更新、削除、検索などの基本的な機能操作をサポートしている。また自動的なマイグレーション機能を持っている。これにより開発者はデータベーススキーマを手動で管理せずに、スキーマの変更を自動的に反映できる。Gorm の他に Xorm などの選択肢もあったが複雑なクエリを簡単に作成するための豊富なクエリビルダー API を提供しているため Gorm を選択した。

サーバサイドプログラムのデプロイする際にはプログラムを実行するために必要な全ての環境をコンテナ内にパッケージ行い、開発環境と本番環境での環境値を最小限に抑えられる Docker を採用した。Docker は、コンテナを水平にスケールアウトでき、これによりアクセスが集中した場合にも、スムーズな処理を行える。デプロイ先は、研究室内のサーバを利用した。研究室内のサーバの利用はクラウドサービスに比べて、長期的にコストを削減することができる。デプロイ先を外部から叩けるように

4.1.2 クライアントサイドの実装

クライアントサイドでは、WebAPI を利用して、Web アプリケーションを実装した。Web アプリケーションのフレームワークとして、Next.js を採用した。Next.js は React.js をベースにしており、React.js の開発を効率化に加え、サーバサイドによるレンダリングをサポートによる SEO 対策やパフォーマンスの向上を実現できる。React.js は、JavaScript のライブラリであり、コンポーネントベースのアーキテクチャを採用している。コンポーネントを独立した単位として開発すると、開発者はそれぞれのコンポーネントに集中できるため、開発効率が向上する。コンポーネントは再利用可能な単位であり、複数のページで同じコンポーネントを利用することができ、これにより、開発コストを削減し、保守性を向上させる。また React は、仮想 DOM とは、実際の DOM (Document Object Model) を操作する代わりに、JavaScript のオブジェクトとして扱う仕組みを採用している。仮想 DOM は、実際の DOM と比較して、更新が必要な部分だけを計算を行うため、パフォーマンスの向上が図れる。

次に実際の画面の構成を説明する。

4.2 利用者の管理とアクセス制御システムの整備

独立した運用を行うためには利用者の管理とアクセス制御システムの整備が必要である。既存の滞在ウォッチは単独コミュニティでのみの運用を前提としており開発者と管理者が同一であった。Web 上からユーザの登録をするシステムが存在せずデータベースに対して直接変更を行う SQL を発行していた。また利用者の権限に関する情報とそれに基づいた Web ページログインの仕組みが存在せず、在室情報はプライバシーに関わるものだが誰でも閲覧可能な状態であった。しかし複数コミュニティ間で運用を行う場合、コミュニティの数が増えるに連れてシステム開発者の利用者管理の負担が大きくなり運用するのは難しくなる。これを解決するには各コミュニティごとに管理者を作り、各コミュニティで独立した運用を行う必要がある。各コミュニティごとに管理者が存在すればコミュニティの利用者の管理を全て行う必要がないため開発者の負担が軽減される。

そこで Firebase Authentication を使った Google プロバイダーによる OAuth 認証を実装した。OAuth (Open Authorization) は、ユーザーがサービスプロバイダー (例えば Google, FaceBook など) に対して、別のサービスにアクセスするための権限を与えるためのオープンスタンダードの認証プロトコルである。OAuth 以外の認証方式として Basic 認証などがある。Basic 認証はユーザ名とパスワードをベーシック認証ヘッダーに埋め込んで送信する方式である。この方式はセキュリティ上の問題が存在する。パスワードが暗号化されていないため、パスワードが盗まれる危険がある。また、Basic 認証はバックエンド側でユーザ名とパスワードを保存する必要があり、管理を行うにはパスワードのハッシュ化など適切な処理を施す必要がある。それと比較して OAuth 認証では、ユーザのアカウント情報を第三者にアプリケーションに渡さずにアクセス権を付与できるため、セキュリティ上の利点がある。OAuth 認証では、アク

アクセス権を持つアプリケーションにのみ有効で、期限切れになると、使用できなくなる。これにより、アクセス権を付与したアプリケーションが不正にアクセスするのを防止できる。これらの理由により、OAuth 認証を採用した。

OAuth プロバイダーには Apple, FaceBook など複数の選択肢がある中で Google を採用した。Google アカウントは世界中で広く利用されており、多くの人がアカウントを持っているおり、多数の利用者に対応可能である。Google アカウントを用いた OAuth 認証は、Google アカウントを使って、サービスプロバイダー（Google）に対して、利用者が滞在ウォッチシステムにアクセスするための権限を与える。利用者は Google アカウントにログインし、Google が提供する OAuth 認証プロセスを通じて、滞在ウォッチシステムにアクセスできる。OAuth 認証により、利用者は自分の Google アカウントの情報（パスワードなど）を滞在ウォッチシステムに入力をせずに、安全にアクセスできる。しかし Google アカウントによる OAuth 認証だけでは意図しない外部のユーザのアクセスを防ぐことはできない。なぜなら Google アカウントを持っているユーザなら誰でも認証を行えるためである。そこで Google アカウントによる OAuth 認証に加えて滞在ウォッチサーバ側で独自のユーザ認証を行った。具体的には滞在ウォッチ管理者が登録したユーザのみを認証するというものである。滞在ウォッチサーバのデータベースには Google のアカウントのユーザ名であるメールアドレスが保存してある。この保存されたメールアドレスは滞在ウォッチの管理が登録したものであるため、メールアドレスが存在する場合は管理者の許可ある正規のユーザと判別が可能である。

以下で具体的なフローを説明するまず初めに利用者は管理者に対して Google アカウントを報告する。報告後管理者が Web の管理者ページのユーザ登録画面から Google アカウントを登録する。

ユーザには 2 種類のパターンが想定される。まず 1 つ目は BLE ビーコンが既に登録済みのユーザ場合である。図 3.1 に示す。これは以前から滞在ウォッチユーザとして登録がされておりビーコンに関する情

メンバーの招待

BLEビーコン登録済み

BLEビーコン未登録

ユーザ選択 *

ユーザを選択

Gmailアドレス *

your@gmail.com

ユーザロール *

一般ユーザ

登録する

図 3.1: BLE ビーコン登録済み

報とユーザ情報が既に存在しており Google アカウントの情報とユーザロールのみがないユーザを想定し

ている。ユーザネームはデータベースに存在しているためバックエンドからデータを取得できる。取得したデータをユーザが選択できるようにセレクトボックスを使用した。ユーザロールのところでユーザの権限レベルの指定ができる。一般ユーザと管理者ユーザの2つがあり、一般ユーザは滞在ウォッチの情報が室情報の閲覧などを行え、管理者ユーザはユーザの登録を行える。登録されるとデータベースに存在するユーザにメールアドレスが紐づき、サーバ側は対象のメールアドレスに対して登録完了のメールを送信する。メールアドレスに対して通知を行うのはユーザがいつ登録が完了したのか気づけるようにするためである。2つ目に想定されるのが BLE ビーコンが登録されていない新規ユーザである。対象ユーザは4.3章で説明するスマホビーコンを使用する新規ユーザである。新規ユーザにはユーザネームがないため管理者はユーザネームを入力する必要がある。その他は1つ目と同じように入力して登録ボタンを押しサーバに対してリクエストを送信する。1つめとの違いサーバ側は新規の UUID の生成を行う。これにより Google アカウントを用いてスマホビーコンユーザの初期登録の際に生成された UUID を返すことができる。

メンバーの招待

BLEビーコン登録済み BLEビーコン未登録

ユーザネーム *

your name

Gmailアドレス *

your@gmail.com

ユーザロール *

一般ユーザ

登録する

図 3.2: BLE ビーコン登録済み

次にユーザ側に戻る Firebase Auth で認証が成功すると JWT(JSON Web TOken) トークンが発行される。Firebase Auth の JWT トークン は Firebase Auth が認証済みのユーザを確認するために使用するトークンである。JWT トークンは、JSON 形式の文字列では発行者、トークンの有効期限、トークンの使用目的、サブジェクトが含まれておりユーザを一意に識別可能である。JWT トークンは、フロントエンドから滞在ウォッチのバックエンドに送信される。滞在ウォッチのバックエンドは Firebase Auth の API を使用して、JWT トークンの検証を行い、トークンが有効であるか確認する。有効な JWT トークンであった場合、アプリケーションのバックエンドは、そのトークンに含まれる情報を使用して、Firebase Auth が認証済みであることを確認する。認証済みであった場合は次にメールアドレス情報を確認する。データベースにメールアドレスが存在する場合フロントエンドに対してアクセスの許可を行う Response Code 200 を返す。

このプロセスによって不正なユーザの在室情報の閲覧を防いでいる．よって適切な範囲での在室情報の共有が可能である．

謝辞の例

本研究を進めるにあたり，多くの御指導，御鞭撻を賜りました愛知工大教授に深く感謝致します。

また，御討論、御助言していただきました，〇×大学工学部電子情報工学科の山谷川介教授，および山谷研究室のみなさんに深く感謝致します。

最後に，日頃から熱心に討論，助言してくださいました愛知研究室のみなさんに深く感謝致します。