

エレベータ利用に関する意思決定を支援する 状況センシングシステムの設計と開発

大平 祐大¹ 荒川 豊¹ 中村 優吾¹

概要：新型コロナウイルス感染症の影響により、公共交通機関や飲食店などの混雑可視化への関心が高まってきている。本研究では、エレベータの混雑を解消するために、エレベータ内の人数と各階の混雑度を利用者に可視化するシステムを開発した。エレベータ内外の人数を計測するために、新型コロナウイルス接触確認アプリからの BLE 信号を検出する方法を提案している。また、BLE 信号から得られる時系列 RSSI データを元にエレベータの乗降検出のアルゴリズムから乗降検出を行なったところ、乗降の検出率は 100%であることを確認した。また、推定待ち時間の算出を行なったところ、待ちの推定開始時刻と推定終了時刻に 5 秒から 10 秒ほどのタイムラグが発生することはあるが、推定待ち時間と実際の待ち時間がおよそ一致することを確認した。

1. はじめに

コロナ禍の影響により、人々が密にならないような環境づくりの重要性が増している。密になりやすい場所としては、電車やバス・飲食店・エレベータ内などが挙げられる。ここでは、エレベータに着目して話を進めていく。エレベータが密になりやすい場所であることは、厚生労働省の新型コロナウイルスの感染拡大防止を訴えるポスター内でも言及されている [1]。

エレベータ利用において発生しうる特有の問題として、エレベータ内は狭く密集しやすい閉鎖空間であるということが挙げられる。例えば、建築基準法施行令第 129 条によると、定員数 6 人のエレベータの床面積は、 $1.06m^2 \sim 1.23m^2$ となり、この面積内に 6 人全員が入るとかなり密集した状態になる。また、別の問題点として混雑を回避しづらい点が挙げられる。利用者がエレベータに乗ったタイミングでは混雑してなくても、途中階で多くの人が乗った場合、元から乗っていた利用者は目的階ではない限りまず降りることはない。また逆のパターンで、利用者は昇降ボタンを押してエレベータを待つが、そのエレベータ内が混雑していた場合、利用者は諦めて次のエレベータを待つ、もしくは無理矢理乗車するような状況が発生しうる。これは、昇降ボタンを押したタイミングでは、利用者は各階の状態とエレベータ内の人数を把握できないから起きる事案だと考えられる。上で述べたような利用者が各階にいた場合、エレベータは各階止まることになるため、エレベータの輸送効

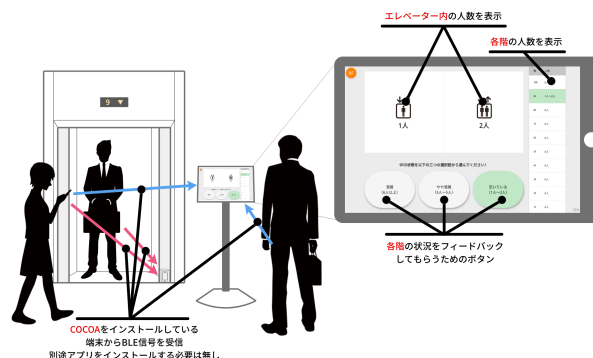


図 1 システム概要

率の悪化や待ち時間の増加にもつながる。

これらの問題を解決するために、スマートフォンアプリを通じてエレベータの混雑状況を利用者に伝える手法が提案されている [2]。スマートフォンアプリによる手法では利用者はアプリケーションをインストールする必要があるため、混雑情報を把握するためのハードルが高くなる問題点が存在する。また [2] では、エレベータ内の混雑状況を把握する手法として、カメラによって乗降者を検出する手法が提案されている。カメラによる検出は、高い精度で検出できるが、プライバシーの問題や天井設置が必要になるなど導入コストの点で問題が存在する。別の切り口の解決方法では、エレベータ前にゲートを設置し、ID カードを読み込ませることで、エレベータ利用者の人数や行き先を全てシステム側で把握し、最適な乗車タイミングやエレベータを選択させる手法が提案されている [3]。ゲートによってエレ

¹ 九州大学, Kyushu University

ベータの利用者を捌くので、エレベータに乗る人数を制御でき、エレベータ内の混雑な状態を解消することは安易だが、導入にコストがかかることやエレベータの台数が少ないほど有効性を示せない問題点が存在する。

本研究では、エレベータ前にタブレット端末を設置し、エレベータ内の人数や各階の混雑状況や 予測待ち時間等を利用者に提示するシステムを提案する (図 1)。このシステムによって、エレベータ特有の問題である昇降ボタンを押したタイミングでは他のフロアの状態やエレベータ内の人数を把握できない問題が解決すると考える。アプリケーションをインストールしてもらう方法と比較して、本システムでは、利用者は簡単に混雑情報を得ることができるので、昇降ボタンを押す前に階段利用もしくは次のエレベータを待つ等の意思決定を促すことができると考える。また、エレベータ内の人数を把握する手段としては BLE 信号を利用した手法を提案する。スマートフォンやタブレット端末でも BLE 信号を取得できるため、本システムではエレベータ内にスマートフォンを別途設置するだけでよく、比較的 low コストなシステムを実現している。また基礎実験の一環として、本システムから取得した BLE 信号のデータを元にエレベータ乗車検出や待ち時間の推定等を行った。

本稿の構成は以下の通りである。第 2 章で、関連研究について述べ、第 3 章で、エレベータ利用状況可視化システムの概要と要件について述べる。また、第 4 章では本システムを用いて利用者の行動パターンおよび待ち時間の推定について説明する。最後に第 5 章では、本稿の結論および今後の課題について述べる。

2. 関連研究

本章では、混雑度推定と混雑度可視化に関する研究と本研究の位置付けについて述べていく。

2.1 混雑度計測に関する研究

カメラを用いて画像処理を行うことにより群衆を検出する手法が提案されている [4]。カメラを用いる場合は、プライバシーを考慮する必要がある。また、エレベータ内は電源を供給することができない場合が多いため、カメラを用いる方法だと、消費電力の問題で一日中バッテリーで運用することは難しい。低消費電力な混雑度計測の手法として、WiFi CSI (Channel State Information) を用いた手法 [5]、BLE 信号の RSSI の強度分布から混雑を推定する手法 [6], [7] がある。ただし、いずれも測定環境を狭む形で送信機と受信機を設置し、電波の減衰量から対象環境の混雑度を計測するものであり、広い室内や屋外を対象としていたり、対向で装置を設置する必要がある。一方、高齢者施設に置いて BLE ビーコンを人に装着することでエリアごとの人数を確実にカウントする方法 [8], [9] などもあるが、大学において全員に BLE ビーコンを配布することは

現実的ではない。

そこで、本研究はコロナウイルス感染症に伴って開発された接触確認アプリ COCOA によって発信される信号を活用する手法を適用する。最近のスマートフォン OS では、バックグラウンド動作が厳しく制限されるため、BLE 信号を発信しつづけるようなアプリの作成はできなくなっていた。またできたとしても多人数の人にインストールしてもらうことは困難であった。それに対して、COCOA は、一定数の人がすでにインストール済みであり、iOS や Android において特別に継続的なバックグラウンド動作が認められている。

この COCOA を活用して混雑度を計測した事例としては、本研究室で 2020 年 6 月より実稼働しているキャンパス混雑度可視化システム itocon (いとこん) [10] や、バス車内の混雑度センシングに利用された例 [11] などがある。この手法では、環境に設置する装置は 1 つで済み、かつ、ユーザ側へのアプリインストールが不要である。欠点としては、必ずしも全員が COCOA をインストールしているわけではないという点である。

2.2 混雑度可視化に関するサービスや研究

混雑度可視化については、マクロとミクロという観点でまとめることができる。マクロな観点では、Google Maps や Yahoo! Map アプリが 2015 年頃から開始している。Yahoo! については 2020 年 1 月に提供終了したもののコロナ禍により再び復活している。これらのサービスは、Google や Yahoo! のサービス利用者の情報に基づいて計測されている。コロナ禍における外出自粛具合を可視化するために、ドコモのモバイル空間統計やソフトバンク系の Agoop など、携帯電話会社による混雑度可視化サービスも認知度が高まっている。これらのサービスでは 500m メッシュ単位での大まかな混雑度が可視化されている。

ミクロな観点では、コロナ禍によって、電車の混雑度が乗換案内で表示されるようになった^{*1}。店舗単位での混雑度については、VACAN^{*2}という会社が台頭している。

しかしながら、エレベータについてはビル内の利用者しかその情報を必要としないことから、エレベータ会社以外に混雑度の計測や可視化を行っていないと思われる。

2.3 本研究の位置付け

本研究に類似する研究として、カメラを用いてエレベータ乗り場の人数計測や車椅子の識別を行なっている研究 [12] や、エレベータの動作履歴から乗り場の人数を推定する研究 [13] などがある。またエレベータの大手である日立ビル

^{*1} 電車の混雑傾向がスグわかる【混雑予報】機能を公開 <https://blog-transit.yahoo.co.jp/info/20200601.html>

^{*2} VACAN <https://corp.vacan.com/>

システムでは、人流予測型エレベータ運行管理システム^{*3}が製品として販売されており、その中にフロアごとの混雑度予測などさまざまな機能が含まれている。しかしながら、このようなハイエンドなエレベータシステムは、極めて導入コストが高いと考えられ、日本最大のキャンパスである伊都キャンパスでも見かけたことがなく、広く普及しているものではないと考えられる。

本研究は、既存の普及型のエレベータシステムを対象とし、後付かつ簡易に設置可能な状況センシングおよび可視化システムを実現するものである。今回の発表には含まれないが、提案システムを用いることで、乗車待ちをせずに階段を利用することを選択するなど意思決定を支援することが目的としており、超高層型のビルに適用するものではなく、階段でも昇降可能な高さの建物に適用先を限定した研究となる。

3. エレベータ利用状況可視化システム

本章では、エレベータ内の人数と各階の人数を可視化するためのシステム図1について説明する。

3.1 システム要件

本システムを実現する上で、以下の要件が必要になる。現在は要件1、要件2のみ実装しており、要件3については今後実装していく予定である。

- 要件1: エレベータ内の人数と各階のエレベータホールの状態を取得
- 要件2: エレベータ利用者に対して要件1で取得した情報を可視化
- 要件3: 予測待ち時間を算出し、エレベータ利用者に対して、階段利用を促す

3.2 システム実装

本システムの構成図を図2に示す。本研究で開発したアプリケーションは、一つのアプリ内にスキャンモードとサイネージモードが存在し、この二つのモードを切り替えるだけで利用できる。スキャンモードは、エレベータ内に設置するスマートフォン(図4)用のモードであり、BLEスキャン機能のみを使用することできる。エレベータ内は基本電源供給を行うことができないため、機能を最低限に絞ることで、電源なしで1日ほど連続で稼働させることを可能とした。またサイネージモードでは、エレベータホールに設置するタブレット(図5)用のモードであり、BLEスキャン機能に加えて、3段階のボタンによるフィードバック機能、エレベータ内の人数と各階の混雑度を表示する機能を使用することができる(図3)。このモードは基本的に電源供給することができるかつ常時点灯することを前提

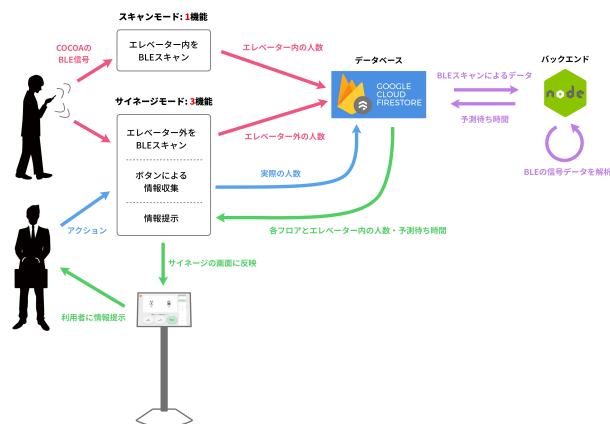


図2 システム構成図

としたモードとなっている。アプリケーションの実装のために、Flutter というフレームワークを使用し、BLE 信号を取得するために、flutter.blue というプラグインを導入している。

技術的な説明について述べる。まず要件1を実現するために、-70dBm 以上の BLE 信号を検出し、人数カウントを行なっている。しかし、全ての BLE 信号を人数のカウントに使用すると、Bluetooth イヤホンやスマートフォンなど複数の機器を所有している人を重複カウントする問題が発生するため、新型コロナウイルス接触確認アプリ(以下、COCOA と呼ぶ)の BLE 信号のみをフィルタリングすることで、スマートフォン端末のみをカウントするようにしている。COCOA の BLE 規格は世界的に決められており[14]、ServiceUUID の値から COCOA の信号のみをフィルタリングすることが可能となっている。ただ、この手法には問題点があり、COCOA のインストール率が 100%ではない点とそもそも Android 端末からの COCOA の信号強度が弱く、十分に捕捉できない点が挙げられる。データベースには、Cloud Firestore を使用しており、BLE の信号データを全て Cloud Firestore に保存している。

次に要件2を実現するために、Cloud Firestore からエレベータ内の人数と各階の混雑度を取得し、タブレット画面に表示している(図3)。また、現在のフロアの状態を利用者に入力してもらうために、混雑(6人以上)・やや混雑(3人~5人)・空いている(1人~2人)の3種類のボタンを表示している。これはエレベータ内と違って、エレベータホールは遮蔽された空間ではなく、意図しない信号までカウントすることがあるので、利用者自身にボタンを押してもらう手法を採用している。

要件3については未実装だが、BLE の信号データを NodeJS 側で解析し、予測待ち時間等をタブレット画面に表示させることを検討している。

^{*3} 日立ビルシステム FI-700 <https://www.hbs.co.jp/fi-700/index.html#9>

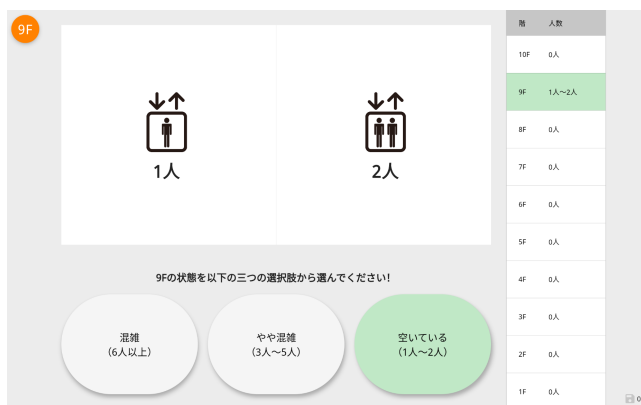


図 3 アプリケーションの画面



図 4 エレベータ内にスマートフォンを設置



図 5 エレベータホールにタブレットを設置

4. エレベータ利用者の行動パターンと待ち時間の推定

本章では、前章で述べたシステムより得られた BLE 信号の時系列 RSSI データを元に、エレベータ利用者の行動パターンと待ち時間の推定を行う手法について述べる。

4.1 対象とするエレベータ利用者の行動パターン

本研究において対象とするエレベータ利用者の行動パ

ターンは以下の 4 種類である。

- エレベータに乗車
- エレベータから降車
- エレベータホール前で待っていたが、諦めて階段を利用
- ただ通り過ぎただけ

4.2 行動パターンと待ち時間を推定するアルゴリズム

時系列 RSSI データから 4.1 で述べた行動パターンを推定するためのアルゴリズム 1 について説明する。エレベータホール前の設置されているタブレット、右側のエレベータ内に設置されているスマートフォン、左側のエレベータ内に設置されているスマートフォンから取得した時系列 RSSI データをそれぞれ A1, A2, A3 とする。A1 において、RSSI の値が -70dBm 以上になった時の時刻を startTime とする。RSSI の最大値から 10dBm 以上下がった時の時刻を endTime とする。 startTime から 10 秒前を $\text{startTime}'$, endTime から 10 秒後を $\text{endTime}'$ とし、 $\text{startTime}'$ から $\text{endTime}'$ までの A2, A3 のデータを A2', A3' とする。A2', A3' の RSSI の最大値がともに -70dBm を下回った場合は、エレベータホール前で待っていたが、「諦めて階段を利用した」もしくは「ただ通り過ぎた」と判定する。A2' の RSSI の最大値が A3' の RSSI の最大値よりも上回った場合、「右側のエレベータに乗車」もしくは「右側のエレベータから降車」と判定し、下回った場合、「左側のエレベータに乗車」もしくは「左側のエレベータから降車」と判定する。A1 の RSSI の値が最大となる時刻より A2' の RSSI の値が最大となる時刻の方が後の場合「右のエレベータに乗車」、先の場合は「右のエレベータから降車」と判定する。「左側のエレベータに乗車」もしくは「左側のエレベータから降車」も同様の手順で判定する。また、「エレベータに乗車」の場合、 startTime と endTime の差分を待ち時間とする。

4.3 実験方法

九州大学伊都キャンパスウエスト 2 号館 9F のエレベータ (図 5) で実験を行う。今回の実験では、COCOA をインストールした iPhone を所有した状態で「右側のエレベータに乗車」・「右側のエレベータから降車」の 2 種類の行動パターンを再現し、実験を行った。実験方法を以下に述べる。

まず、9F のエレベータホール前の降ボタンを押し、エレベータが来るまで右側のエレベータの前で待機する。エレベータに乗り、1F まで移動し、エレベータから降りる。1・2 分ほど待機したのちに、1F のエレベータホール前の昇ボタンを押し、同様の手順で、9F まで移動する。これを 3 セット繰り返す。昇降ボタンを押した時刻、エレベータに乗った時刻、エレベータから降りた時刻を記録する (表 1)。

Algorithm 1 行動パターンの推定アルゴリズム

Require: A_1, A_2, A_3
Ensure: Activity Pattern p .

```

1:  $i \leftarrow 0$ 
2:  $max \leftarrow null$ 
3:  $startTime \leftarrow null$ 
4:  $endTime \leftarrow null$ 
5: while  $i \leq A_1.length$  and  $p = null$  do
6:   if  $A_1[i].rssi \geq -70$  or  $startTime \neq null$  then
7:     if  $startTime = null$  then
8:        $startTime \leftarrow A_1[i].time$ 
9:     end if
10:    if  $A_1[i].rssi > max.rssi$  then
11:       $max \leftarrow A_1[i]$ 
12:    end if
13:    if  $max.rssi - A_1[i].rssi \geq 10$  or  $A_1[i].time \geq A_1[i-1].time + 30s$  then
14:       $endTime \leftarrow A_1[i].time$ 
15:       $startTime' \leftarrow startTime - 10s$ 
16:       $endTime' \leftarrow endTime + 10s$ 
17:       $max2 \leftarrow null$ 
18:       $max3 \leftarrow null$ 
19:       $k \leftarrow 0$ 
20:      while  $k \leq A_2.length$  do
21:        if  $startTime \leq A_2[k].time \leq endTime$  then
22:          if  $max2 = null$  or  $max2.rssi \leq A_2[k].rssi$  then
23:             $max2 \leftarrow A_2[k]$ 
24:          end if
25:        end if
26:      end while
27:       $k \leftarrow 0$ 
28:      while  $k \leq A_3.length$  do
29:        if  $startTime \leq A_3[k].time \leq endTime$  then
30:          if  $max3 = null$  or  $max3.rssi \leq A_3[k].rssi$  then
31:             $max3 \leftarrow A_3[k]$ 
32:          end if
33:        end if
34:      end while
35:      if  $max2.rssi \geq -70$  or  $max3.rssi \geq -70$  then
36:        if  $max2.rssi \geq max3.rssi$  then
37:          if  $max2.time \geq max.time$  then
38:             $p \leftarrow$  "Get on the elevator on the right."
39:          else
40:             $p \leftarrow$  "Get off the elevator on the right."
41:          end if
42:        else
43:          if  $max3.time \geq max.time$  then
44:             $p \leftarrow$  "Get on the elevator on the left."
45:          else
46:             $p \leftarrow$  "Get off the elevator on the left."
47:          end if
48:        end if
49:      else
50:         $p \leftarrow$  "Give up the ride or Just passing through."
51:      end if
52:    end if
53:  else
54:    if  $A_1[i].time \geq A_1[i-1].time + 30s$  then
55:       $p \leftarrow$  "Just passing through."
56:    end if
57:  end if
58:   $i \leftarrow i + 1$ 
59: end while

```

表 1 実験の記録

行動パターン	昇降ボタンを押した時の時刻	エレベータに乗った時刻	エレベータから降りた時刻	待ち時間
9F から 1F に移動	15 時 10 分 31 秒	15 時 11 分 01 秒	15 時 11 分 22 秒	30 秒
1F から 9F に移動	15 時 13 分 20 秒	15 時 13 分 45 秒	15 時 14 分 10 秒	25 秒
9F から 1F に移動	15 時 16 分 32 秒	15 時 16 分 50 秒	15 時 17 分 22 秒	18 秒
1F から 9F に移動	15 時 19 分 00 秒	15 時 19 分 12 秒	15 時 19 分 43 秒	12 秒
9F から 1F に移動	15 時 23 分 17 秒	15 時 23 分 58 秒	15 時 24 分 34 秒	41 秒
1F から 9F に移動	15 時 25 分 55 秒	15 時 26 分 42 秒	15 時 27 分 32 秒	47 秒

表 2 アルゴリズムによる推定待ち時間

行動パターン	startTime	endTime	推定待ち時間
右のエレベータに乗車	15 時 10 分 40 秒	15 時 11 分 10 秒	30 秒
右のエレベータに乗車	15 時 16 分 32 秒	15 時 16 分 52 秒	20 秒
右のエレベータに乗車	15 時 23 分 24 秒	15 時 24 分 04 秒	40 秒

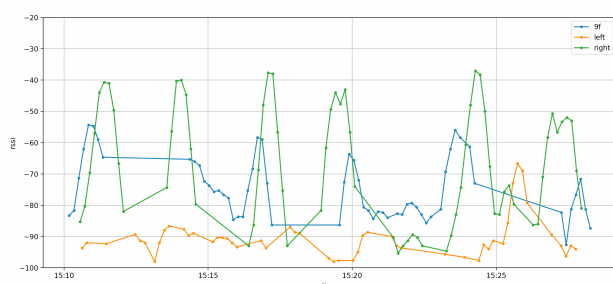


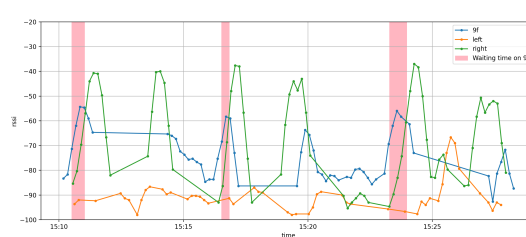
図 6 時系列 RSSI データ

4.4 実験結果

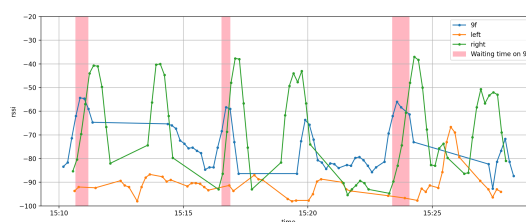
9F のエレベータホール前のタブレット端末, 右側・左側のエレベータ内に設置するスマートフォン端末から取得した 15 時 10 分 00 秒から 15 時 30 分 00 秒までの時系列 RSSI データは, 図 6 のようになった. アルゴリズム 1 を元に, 乗車・降車判定を行ったところ検出率は 100%であった. また, 乗車した時の 9F での待ち時間の推測結果を図 7 と表 2 に示す. 表 1 と比較して, 推定開始時間 (startTime) と推定終了時間 (endTime) に 5 秒から 10 秒ほどのタイムラグは生じることはあるが, 待ち時間の推定値はおおよそ一致した. 現在, BLE のスキャンのタイミングは約 10 秒に 1 回としているが, スキャンの間隔を短くすれば, 精度の向上が見込まれる.

5. おわりに

本稿では, エレベーター内の人数と各階の混雑度を利用者に可視化するシステムの提案を行なった. 今後の課題として, COCOA の BLE 信号数をエレベータ内の人数カウントに使用した手法だと, COCOA のインストール率が低い場合に正確な人数カウントができない, Android 端末を捕捉できない等の問題点が存在するため, 代替手法を考案する必要がある. また, 今回取得した BLE 信号のデータを元に予測待ち時間を算出するアルゴリズムについて考案し, 本システムによって, 混雑問題を解決したかどうかの評価



(a) 実際の待ち時間



(b) アルゴリズムによる推定待ち時間

図 7 時系列 RSSI データと 9F での待ち時間

実験を行う必要がある.

謝辞 本研究の一部は, JSPS 科研費 JP18H03233 および NEDO My-IoT 開発プラットフォームの研究開発 (JPNP18014) の支援のもと実施された.

参考文献

- [1] Prime Minister's Office Of Japan. 密 - 首相官邸. "https://www.kantei.go.jp/jp/content/000062771.pdf", (2021 年 8 月 23 日閲覧).
- [2] 鹿島建設株式会社. スマートビルがワーカーの健康行動をサポート. "https://www.kajima.co.jp/news/press/202107/6a1-j.htm", (2021 年 8 月 23 日閲覧).
- [3] 三菱電機. エレ・ナビ (ele-navi) でエレベーターの渋滞を解決. "https://www.mitsubishielectric.co.jp/elevator/nayami/002/", (2021 年 8 月 23 日閲覧).
- [4] 新井啓之, 伊藤直己, 谷口行信. 群衆をマクロにとらえる画像処理技術~人物・群衆の幾何モデルに基づいた人数推定とその応用~. Technical Report 13, NTT メディアインテリジェンス研究所, NTT メディアインテリジェンス研究所, NTT メディアインテリジェンス研究所, jan 2014.

- [5] 松本大生, 高橋遼, 石田繁巳, 荒川豊. 無線 lan を用いたデバイスフリー室内混雑度推定の初期評価. 情報処理学会第 82 回全国大会, 2020.
- [6] 鈴木優, 藤本まなと, 吉野幸一郎, 日高真人, Nguyen Quynh Mai, 永野一馬, 中村優吾, 大坪敦, 田中翔平, 安本慶一, 中村哲. 京都インバウンド観光における iot2h 観光情報アプリケーションの構築. 第 11 回データ工学と情報マネジメントに関するフォーラム (DEIM2019), 3 2019.
- [7] Kazuhito Umeki, Yugo Nakamura, Manato Fujimoto, Yutaka Arakawa, and Keiichi Yasumoto. Real-time congestion estimation in sightseeing spots with ble devices. In *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pp. 430–432. IEEE, 2018.
- [8] Kiyoaki Komai, Manato Fujimoto, Yutaka Arakawa, Hirohiko Suwa, Yukitoshi Kashimoto, and Keiichi Yasumoto. Beacon-based multi-person activity monitoring system for day care center. In *2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, pp. 1–6. IEEE, 2016.
- [9] Kiyoaki Komai, Manato Fujimoto, Yutaka Arakawa, Hirohiko Suwa, Yukitoshi Kashimoto, and Keiichi Yasumoto. Elderly person monitoring in day care center using bluetooth low energy. In *2016 10th International Symposium on Medical Information and Communication Technology (ISMICT)*, pp. 1–5. IEEE, 2016.
- [10] 橋遼, 林健太, 光来出優大, 二俣雅紀, 井上隼英, 松尾周汰, 石田繁巳, 荒川豊, 高野茂. バス停混雑度可視化システム itocon (いとこん). 第 28 回マルチメディア通信と分散処理ワークショップ論文集, pp. 227–230, nov 2020.
- [11] 立花 巧樹 中村 優吾 松田 裕貴 諏訪 博彦 安本 慶一 金光 勇慈. Ble を用いた路線バスの混雑度推定. 情報処理学会 DICO MO シンポジウム, 2021.
- [12] 内館光, 猪田良介, 辻俊明, 阿部茂. 実時間画像処理によるエレベータ乗場の人数計測と車椅子の識別. 電気学会論文誌 D (産業応用部門誌), Vol. 129, No. 6, pp. 578–584, 2009.
- [13] 田口浩, 鈴木直彦, 鹿井正博. エレベーター動作履歴に基づく乗場待ち客数の推定. In *IEICE Conferences Archives*. The Institute of Electronics, Information and Communication Engineers, 2015.
- [14] Google. Contact tracing - bluetooth specification v1.1. ”https://blog.google/documents/58/Contact_Tracing_-_Bluetooth_Specification_v1.1_RYGZbKW.pdf”, (2021 年 8 月 25 日閲覧) .