

映像に基づく入退室カウントセンサを活用した 多地点混雑情報配信サービスの開発

高槻 大貴[†] 佐伯 幸郎[†] 中村匡秀^{†††} 多田 幸生[†]

[†] 神戸大学 〒657-8501 神戸市灘区六甲台町 1-1

E-mail: [†]takatsuki@ws.cs.kobe-u.ac.jp, ^{†††}masa-n@cs.kobe-u.ac.jp

あらまし 我々は、先行研究において、部屋の入口の映像を解析して、移動物体の入退室をカウントする IoT センサを開発している。本研究では、このセンサを多地点に配備し、多地点の混雑情報を統合して配信するサービスを開発した。具体的には、複数のセンサで収集した在室人数をエッジサーバに集約し、混雑状況を解析して、twitter に配信する。

キーワード IoT, コンピュータビジョン, サービス, 映像

Developing information delivery service of multi-point crowds using video-based person counting sensors

Daiki TAKATSUKI[†], Sachio SAIKI[†], Masahide NAKAMURA^{†††}, and Yukio TADA[†]

[†] Kobe University Rokkodai-cho 1-1, Nada-ku, Kobe, Hyogo, 657-8501 Japan

E-mail: [†]takatsuki@ws.cs.kobe-u.ac.jp, ^{†††}masa-n@cs.kobe-u.ac.jp

Abstract We have analyzed a picture at the entrance of the room in the preceding study and have developed the IoT sensor which counts entering and leaving the room of a mobile. This sensor is deployed in numerous spots by this research, crowded information on numerous spots is integrated and delivered service has been developed. Specifically, the stay in the room number of people collected by more than one sensor is concentrated in an edge server, crowded circumstances are analyzed and it's delivered to twitter. Algorithm of an image analyses will be reconsidered by this research later, the one of more than one people, and others, a mis-count by a companion was improved.

Key words IoT, Computer vision, Service, Video

1. はじめに

近年、ICT 技術の飛躍的な進化に伴い、様々なセンサや IoT 機器でリアルタイムに周囲の環境情報を取得する技術開発が進んでいる。このような IoT センサを実現する技術の一つとして、カメラを用いた周辺環境を理解しようとするコンピュータビジョン技術がある。コンピュータビジョンでは静止画や動画から実世界の情報を取得する全ての過程を扱うため、検知、識別、理解など人間が視覚的に行っていた行為を自動化する様々な技術的な応用が広がり、これらの技術を用いた付加価値サービスが実現している。例えば、赤ちゃん見守り用のアプリ [1] は遠隔地から宅内の赤ちゃんの様子をリアルタイムで見ることができ、コンピュータビジョンに基づき動きを検知を実現すると同時に、音声処理技術を併用することで異常検知を実現し、何か問題が起きれば通知するといった機能がある。また、自動

車業界では、近年自動運転実現に向けた取り組みとして、自動車に搭載されている単眼カメラが様々な対象物をリアルタイムに検出・認識するというコンピュータビジョンの技術が利用されている [2] [3] [4]。このようなコンピュータビジョンの応用例として、カメラから得た取得画像をもとに、空間における人間の存在を把握する人流センサ [5] がある。人流センサでは人間の存在数や移動方向の検出をもとに特定空間内での人間の存在を認識できるため、駅の混雑解消や大きなイベント時の人間の動線の把握、集客数の確認などへの利活用が進んでいる。

本研究では、先行研究により開発された、コンピュータビジョンによる人流 IoT センサの一つである、人間の部屋への入退室を計測できる監視 IoT システム [6] をもとに、同一用途目的の複数の空間が存在するような場合に、その空間の混雑具合をリアルタイムに把握し、より利用がし易い空間を通知可能な多

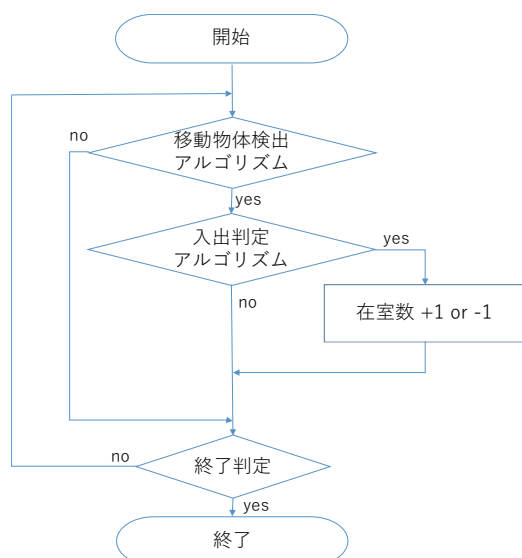


図 1 入退室カウントセンサのフローチャート

点混雑情報配信サービスの提案を行う。提案サービスを利用することで、ユーザはフリースペース、食堂、商業施設など同等の機能を持ち、かつ事前に混雑具合の比較が困難であるような空間を利用する際、予めどちらが空いているか知ることができる、もしくはどちらが人と多く出会えるかなどを知ることができるなど新たな付加価値を得ることができる。多地点混雑情報配信サービスは、複数の入退室カウントセンサを WebSocket を用いリアルタイムに情報取得する管理サービスの開発と、ユーザへの情報提供を行う通知サービスから構成される。

本論文の構成は以下の通りである。まず 2 章で先行研究で開発された監視 IoT システムである入退室カウントセンサについて述べ、3 章では多地点混雑情報配信サービスの仕様について示す。また 4 章では多地点混雑情報配信サービスの実装と実環境下での実験を行い、提案サービスの有効性について考察を行う。

2. 入退室カウントセンサ

2.1 概要

入退室カウントセンサとは、シングルコンピュータの Raspberry Pi3(Model B) を用い、接続された web カメラから得られる映像から人の動きを検出する移動物体検出アルゴリズムと入退室をカウントする入出判定アルゴリズムによって移動体の入退室をカウントするセンサである。移動物体検出アルゴリズムでは背景差分法によって移動物体を検出しており、これらの処理は OpenCV によって実装されている。

2.2 想定する監視環境

入退室カウントセンサが想定する使用環境は、照明量が一定である屋内である。この理由として、入退室カウントセンサが人間の移動検出に利用する背景差分法の適用可能範囲として、背景画像中の光量の変動が少ないことが前提としているためである。光量の変動が大きい場合には誤認識の原因となり得る。また、入退室カウントセンサでは、カメラの設置場所も重要な検討事項である。例えば部屋への入退室をカウントする場合、

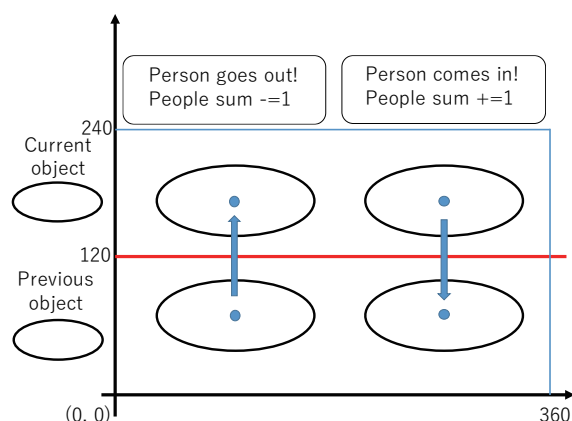


図 2 入出判定アルゴリズムの概要

出入りの邪魔にならないこと必須であり、必然的に出入口付近の壁面もしくは天井への設置となる。壁面に設置した場合、人が一人づつ間隔をあけて入ってくる場合には検出が可能である。しかし、人々が横に並ぶように入る、もしくは出入りですれ違う場合では、対象物が重なり、背景差分から連続的に人を検出するのは困難である。一方で天井から床を見下ろす形でカメラを設置する場合、人が複数列になってもすれ違って重なってしまうことが前者よりも少なく、より正確な検出が可能になると期待できる。よって設置位置は天井にし、かつドアの開閉が検出に影響しないよう、ドアが開けられても検出画面に映らない位置へのカメラ設置を想定する。

2.3 移動物体検出アルゴリズム

入退室カウントセンサにおける移動物体の検出アルゴリズムは以下の 6 ステップで行われる。

STEP1: Web カメラから Raspberr Pi に撮影した動画を送信する。

STEP2: ステップ 1 の動画から 1 フレームの画像を `ret, frame = cap.read()` によって取得する。

STEP3: Web カメラから送られるフレームサイズはデフォルトで 480×640 pixel であるが、これを 240×320 pixel に縮小する。この理由は、以後の画像認識の操作で 1 フレームのすべての pixel を操作する際、 480×640 pixel の解像度であると処理に時間がかかり、fps が大幅に落ちてしまうためである。

STEP4: BGR 空間のフレーム画像を `cvtColor` 関数でグレースケール化する。

STEP5: 背景画像と現在のフレームの差分の絶対値を `absdiff` 関数により取得する。この関数はグレースケール画像を比較するため、ステップ 4 の変換が必要である。

STEP6: ステップ 5 で得られた差分画像を二値化する。

2.4 入出判定アルゴリズム

入出判定アルゴリズムは、移動物体検出アルゴリズムで移動体が検出された際に、対象画像とその前フレーム画像との差分をもとに入出の判定を行う。図 2 に入出判定アルゴリズムの概要を示す。Web カメラの入力画像の Y 座標は最大で 240 であり、最小値は 0 である。入出判定を行うには、まず面積が 4000 pixel 以上の輪郭の現在のフレームにおける Y 座標の値と、

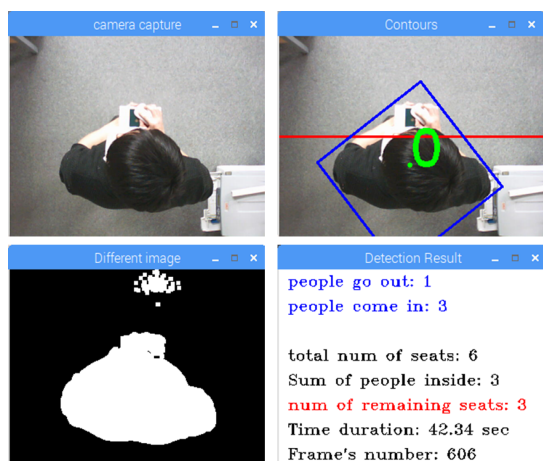


図 3 入退室カウントセンサの実行画面

1 つ前のフレームにおける Y 座標の値を配列に格納する．移動物体の入出のタイミングは中間線（Y 座標 120）に対して，図 2 のように現在と前のフレームの Y 座標が中間線を挟み反対に輪郭が検出されたときである．移動物体の Y 座標が奥側から中間線を越えて出入りに移動した場合，部屋にいる人数を減らす．逆に出入り口側から奥側に移動した場合，部屋にいる人数を増やす．

2.5 検出画面と結果の表示

入退室カウントセンサでは，入退室検出実験の検出画面と結果の表示を 4 つの Windows で表示する．図 3 に実験時の実行画面を示す．

それぞれの Window は以下の項目を表示する．

Camera capture: カメラの映像をリアルタイムに表示

Different image: 2 値化した背景差分画像

Contours: 4000pixel 以上のオブジェクトの輪郭を描写

Detection result: 検出された入退室の計算結果を出力

なお，Detection result に表示される項目は以下である．

- People go out: 退出者の合計
- People come in: 入室者の合計
- Total num of seats: 部屋の中の席の合計（プログラム内で定義する定数）

- Sum of people inside: 部屋の中にいる合計人数．（初期人数）+（部屋の外から中に入った人の合計）-（部屋の中から外に出て行った人の合計）で定義される．

- Time duration: 背景を固定してから経過時間
- Frame's number: 背景を固定してから送られたビデオフレームの総数

3. 多地点混雑情報配信サービス

多地点混雑情報配信サービスは，複数の空間で入退室カウントセンサを用い，それぞれの混雑状況を計測し，その結果をサーバで一元管理することで，比較した混雑状況を利用者に通知するサービスである．例えば，より空いている食堂，空いているショッピングセンターなどを利用者は事前に知ることがで

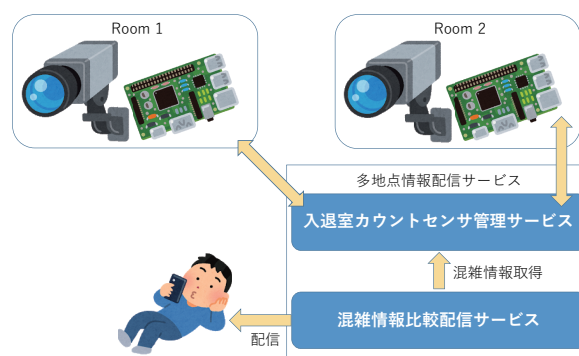


図 4 多地点混雑情報配信サービスの構成図

きる．

図 4 に多地点混雑情報配信サービスの構成を示す．多地点混雑情報配信サービスは管理サービスである入退室カウントセンサ管理サービス，ユーザへの情報配信を行う混雑情報比較配信サービスで構成される．以下にそれぞれのサービスについて述べる．

3.1 入退室カウントセンサ管理サービス

入退室カウントセンサ管理サービスは複数の入退室カウントセンサの検出状況を一元的に管理するサービスである．管理サービスは入退室カウントセンサそれぞれと 1 対 1 での通信を常に行い，それぞれのセンサの状況をリアルタイムに把握する．また，それぞれのセンサがもつ，設置場所，用途などの詳細情報についても管理することで，次に述べる混雑情報比較配信サービスで比較対象となり得るセンサ群を定義する．

3.2 混雑情報比較配信サービス

混雑情報比較配信サービスは，入退室カウントセンサ管理サービスで算出された混雑情報を元に，あらかじめ定義された比較対象となり得る入退室カウントセンサの比較結果をユーザに配信するサービスである．ユーザへの情報配信方法としては，プッシュ型での通知を行う．そのためユーザは受動的に情報を得ることができる．配信サービスで利用が可能な配信方法としては，twitter, Line などの SNS，メールでの配信などが想定される．

4. 多地点混雑情報配信サービスの実装

本章では多地点混雑情報配信サービスの具体的な実装を行い，実環境における実験を行った結果について述べる．

4.1 入退室カウントセンサ管理サービスの実装

入退室カウントセンサ管理サービスは Raspberry Pi3 で動かすことを想定している．そのためリアルタイム性の確保と同時に，軽量の双方向通信を行えることが必要となる．そこで本研究における実装ではサーバプログラムを Python で実装し，入退室カウントセンサから管理サービスに情報を送信する手段として通信には WebScket を利用する．WebSocket は以下の特徴をもっている．

- サーバ（本研究では管理サービス）とクライアント（入退室カウントセンサ）が 1 度コネクションを確立すると，その後の通信を専用プロトコルで行う．

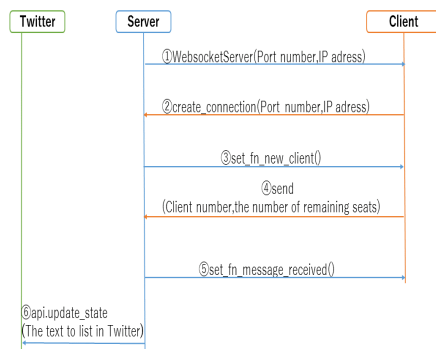


図 5 WebSocket によるサーバとクライアントのアルゴリズム

- ・ 接続が確立しているサーバと全てのクライアントは同じデータを共有し，リアルタイムで送受信できる．
- ・ サーバとクライアントの双方向通信を低コストで行うことができる．

WebSocket によるサーバとクライアントのアルゴリズムの流れを図 5 に示す．以下に図 5 の関数内で行われる処理をステップごとに説明を行う．

STEP1: WebsocketServer を使い，ポート番号とホストの IP アドレスを設定することでサーバを立ち上げる．

STEP2: ws = create_connection() 関数によって，接続したいポート番号と IP アドレスを設定しサーバと接続する．

STEP3: 新しいクライアントが接続したら，接続しているクライアントの数を 1 つ増やす．またクライアントとの接続が切れたら，接続しているクライアントの数を 1 つ減らす．

STEP4: ws.send() 関数で監視している部屋の現在の残席数とクライアント番号を文字列にしてサーバへ送信する．

STEP5: 接続したクライアントが 1 つ以上ある場合，クライアントからクライアント番号とその部屋の残席数の文字列を受け取り，文字列を整数に変換した後比較する．

STEP6: 規定の残席数になった場合，比較した結果を混雑情報比較配信サービスへ送信する．

4.2 混雑情報比較配信サービスの実装

混雑情報比較配信サービスは，ユーザにリアルタイムに混雑状況の比較結果を配信する必要がある．そのため，利用者が日常的に利用している情報を受信するサービスを利用し，混雑情報を送ることが望ましい．そこで，本研究では SNS である Twitter に混雑情報を通知するシステムを実装する．実際に Twitter による配信を実現するため，配信用 Twitter アカウントを作成し，Twitter API を利用してそのアカウントに Raspberry Pi3 から実際に情報を送信する．配信サービスの実装には Python を使い，Twython という Python 用ライブラリを Raspberry Pi3 にインストールする．次に，サーバ側のプログラムから Twitter アカウントにアクセスするために，<http://apps.twitter.com/> の Web ページで，デベロッパー登録とアプリケーション登録

を行う必要がある．そして，サーバ側のプログラムの中で，登録時に習得した情報を以下のように記述する．

```

import os from twyhon
import Twython
CONSUMER_KEY = '取得した' API Key
CONSUMER_SECRET = '取得した' API Secret
ACCESS_KEY = '取得した' Access Token
ACCESS_SECRET = '取得した' Access Token Secret
api = Twython(CONSUMER_KEY, CONSUMER_SECRET,
              ACCESS_KEY, ACCESS_SECRET)
  
```

そして，api.update_status(status= 記述したい本文) というアップデート関数を呼び出すことにより，Twitter アカウントに送信したいテキストメッセージを Bot することが可能である．

4.3 実験

実装した多地点混雑情報サービスの実験を行う．本実験は図 8 と図 9 の 2 つの部屋 (システム設計実験室 1 S207 とシステム設計研究室 1 S206) を使用し，あらかじめプログラム内で定義した部屋の総席数と入退室カウントセンサー (クライアント) により検出した入退室情報を基に求められる室内の残席数をそれぞれ統合管理プログラム (サーバ) に送信し，より空いている部屋を正しく Twitter に通知できているかを検証する．また入退室の際は 1 人だけでカメラの監視範囲を通り，他の人と接触しないようにする．

本実験では，クライアントは部屋の総席数を 7 席と定義し，それぞれの部屋の残席数をサーバに送信する．サーバでは，どちらかの部屋の残席数が変化した時にサーバのターミナルにそれぞれの部屋の残席数を表示する．どちらかの部屋の残席数が 5 席または 3 席になった時，その時点の残席数を比較し，それぞれの残席数と比較結果を Twitter に通知する．人の入退室は図 7 の順で行うとする．

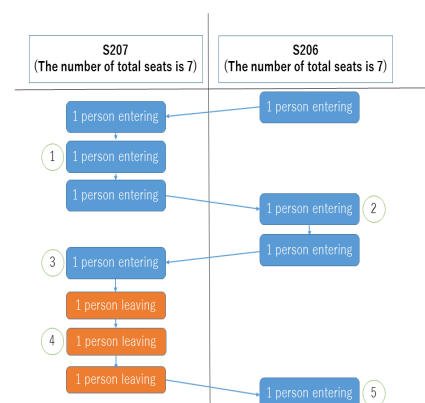


図 6 実験における入退室の過程

本研究の実験環境として，カメラの具体的な位置は図 7 のように地面から高さ 270cm に設置し，カメラが撮影できる範囲領域は縦 170cm，横 260cm であり人が数人収まるサイズである．また出入り口のドアが検出対象に入らないように，カメラをドアの半径分よりもさらに奥に設置するようにした．また部屋の

明るさを一定にして行う．また，もう1つ部屋（S206 システム研究室1）は図8のようにカメラと Raspberry Pi を設置する．カメラの具体的な位置は地面から 270cm で，カメラが撮影できる領域は縦 130cm，横 170cm である．実験ではこの2つの部屋の入退室状況を比較する．



図7 システム設計実験室 S207

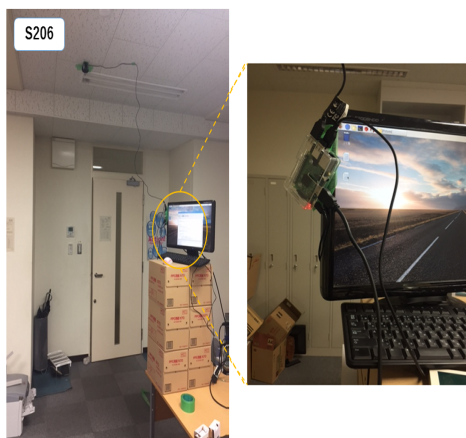


図8 システム設計研究室 S206

4.4 実験結果

図6の順で入退室を行った時，㊦で S207 の部屋の残席数が5席になり，その時点でのそれぞれの残席数と比較結果がツイートされる（図9）．㊦では S207 の残席数が3席となる（図10）．最後に㊦で S206 の残席数が3席となりツイートされる（図11）．以下はそれぞれの部屋の管理画面とツイート結果である．



図9 それぞれの部屋の状況とツイート結果（1）



図10 それぞれの部屋の状況とツイート結果（3）



図11 それぞれの部屋の状況とツイート結果（5）

これらの結果から，多地点混雑情報配信サービスは各部屋の残席数を正しく比較して Twitter に送信できていることが確認できる．また，図12と図13は各部屋の検出した移動物体の Y

座標と室内の人数の推移をプロットしたものである。入退室カウントセンサーによる移動物体検出も正しく検出されていることがわかる。

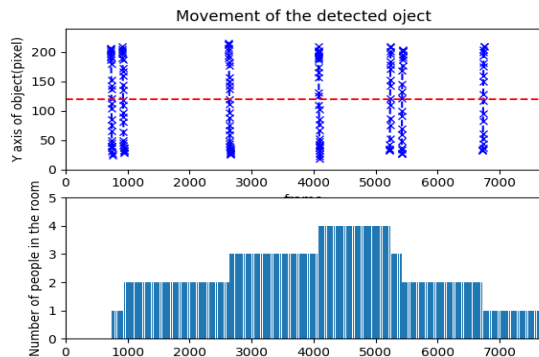


図 12 S207 で検知した物体の Y 座標と部屋の人数

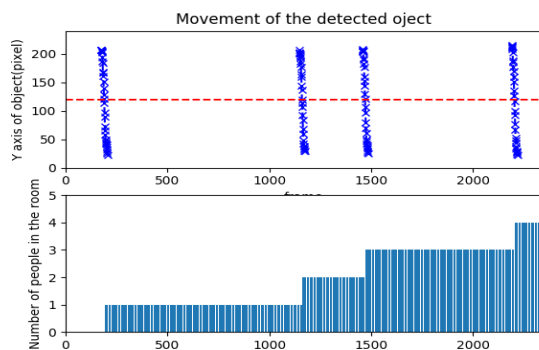


図 13 S206 で検知した物体の Y 座標と部屋の人数

5. おわりに

多地点混雑情報配信サービスの実験を通して、入退室カウントセンサーを用いて 2 つの部屋の残席数をサーバへ送り、比較することで、どちらがより空いているかを比較した後、Twitter に送信しているので様々な人が手軽に現在の部屋の状況を知ることができる。プログラムの改良により 3 つ以上の部屋の入退室管理をすることができ、より実用的なシステムになることが考えられる。しかし、本研究では部屋へ入退室する場所は 1 箇所に限定して検出を行い管理していたが、大学食堂など出入りする場所が複数ある場合、同じ部屋であることを認識して管理しなければならないという問題が残っている。統合管理プログラムでは部屋の出入りする場所が複数ある場合、同じ場所として認識し入退室情報を管理しなければならないといった課題があり、プログラムのさらなる改良が求められ、今後の課題としたい。これらのシステムやプログラム開発の考え方は、大学の食堂だけでなく様々な施設で応用ができ、より実用的なものに

なると考えられる。本研究で開発されたプログラムも、その発展の足がかりになることが十分に期待される。

謝辞 この研究の一部は、科学技術研究費（基盤研究 B 16H02908, 15H02701, 基盤研究 A 17H00731, 萌芽研究 15K12020）、および、立石科学技術振興財団の研究助成を受けて行われている。

文 献

- [1] 日経 BP, “赤ちゃん向け iot デバイス,” <http://business.nikkeibp.co.jp/atcl/report/15/061700004/112400163/>, Nov. 2016.
- [2] J. Choi, J. Lee, D. Kim, G. Soprani, P. Cerri, A. Broggi, and K. Yi, “Environment-detection-and-mapping algorithm for autonomous driving in rural or off-road environment,” *IEEE Transactions on Intelligent Transportation Systems*, vol.13, no.2, pp.974–982, June 2012.
- [3] S. Tsugawa, “Vision-based vehicles in japan: machine vision systems and driving control systems,” *IEEE Transactions on Industrial Electronics*, vol.41, no.4, pp.398–405, Aug. 1994.
- [4] A. Almagambetov, S. Velipasalar, and M. Casares, “Robust and computationally lightweight autonomous tracking of vehicle taillights and signal detection by embedded smart cameras,” *IEEE Transactions on Industrial Electronics*, vol.62, no.6, pp.3732–3741, June 2015.
- [5] T. Shimmura, H. Arai, and U. Inoue, “Estimating human-flow speed for video surveillance by probabilistic stands,” *IEEE Conference on Cybernetics and Intelligent Systems*, 2004., vol.1, pp.179–184vol.1, Dec. 2004.
- [6] 村山伶和, “移動物体検出を応用した監視 iot システムの開発,” 神戸大学工学部情報知能工学科卒業論文, pp. • • • • •, 2016.