

テクニカルノート

# ジェスチャ認識システム構築に向けた センサ配置およびデータ収集支援ソフトウェアの開発

齋藤 彩音<sup>1,a)</sup> 河合 航<sup>2,b)</sup> 杉浦 裕太<sup>1,3,c)</sup>

受付日 2019年5月30日, 採録日 2019年11月29日

**概要:** 実世界センサでの計測と機械学習を組み合わせることでジェスチャ識別を行う研究は数多く存在する。実世界センサは配置や個数, 計測したい動作によって測定結果が変化することからセンサ配置を検討することが重要である。しかし, 識別率の良いセンサ配置を検討することや実世界での学習データの蓄積には手間がかかる。本研究では, 実世界センサの配置をデザインできるソフトウェアを開発した。ソフトウェアでは, Kinect を用いて記録した実世界の変形と自由に配置されたセンサとの距離を計測しジェスチャ識別器を生成する。また, ソフトウェア上のデータで生成された識別器を用いて実世界に配置したセンサでのジェスチャ識別を行った。

**キーワード:** センサ配置, 機械学習, 測距センサ

## Software to Support Layout and Data Collection for Gesture Recognition System Development

AYANE SAITO<sup>1,a)</sup> WATARU KAWAI<sup>2,b)</sup> YUTA SUGIURA<sup>1,3,c)</sup>

Received: May 30, 2019, Accepted: November 29, 2019

**Abstract:** There have been many studies of gesture recognition by combining real-world sensor and machine learning. It is important to consider the sensor layout because the measurement result varies depending on the layout and the motion. However, it takes time and effort to find a sensor layout that has high identification accuracy and to acquire learning data. In this study, we developed software that can arrange real-world sensors. The software measures the distance between the sensors and a mesh created from measurements of real-world deformation and generates the classifier. We created physical devices and recognize gestures by using generated classifier.

**Keywords:** sensor layout, machine learning, distance-measuring sensor

### 1. 序論

日常生活の中に実世界センサを組み込むことによって実世界の物理的な変形を計測できる。実世界計測を行う場合は, システムでとらえたい実世界の変化を考えたいうえで, コストやデバイスのサイズなどを考慮しながらセンサの個

数, 配置位置を検討する。しかし様々な制約を考慮しながら最適なセンサ配置を設計するのは難しい。また, センサと機械学習を組み合わせることでジェスチャ識別や姿勢推定ができる。たとえば, Touché [9] はセンサに触れたときの周波数応答を機械学習することでタッチジェスチャを識別している。このように事前に学習過程を経ることで複雑な事象を識別する研究が多く行われているが, 識別したい状態や個人差に応じた特徴的なセンサデータを用いて識別器の生成を行う必要がある。しかし, 特徴的なセンサデータを取得できるセンサの個数, 位置や角度は試行錯誤によって決めていることが多い。また, センサの個数や配置を変更するたびに実世界において学習データを蓄積することは労力がかかる。

<sup>1</sup> 慶應義塾大学  
Keio University, Yokohama, Kanagawa 223-8522, Japan  
<sup>2</sup> 東京大学  
The University of Tokyo, Bunkyo, Tokyo 113-8656, Japan  
<sup>3</sup> 科学技術振興機構さきかけ  
JST PRESTO, Kawaguchi, Saitama 332-0012, Japan  
a) ayane-3110@keio.jp  
b) w-kawai@mi.t.u-tokyo.ac.jp  
c) sugiura@keio.jp

本研究では実世界での学習データ蓄積の手間を解消するため、実世界センサの配置をデザインできるソフトウェアを開発した。本研究では測距センサを対象にソフトウェアを開発し、手のジェスチャ識別を行った。このソフトウェアでは Kinect を用いて実世界の変形を 3D 深度情報として記録している。Kinect は Kinect Studio といったアプリケーションを使用することによって 3D 深度情報を録画することができる。この機能を利用することで、実世界でセンサ配置変更のたびに学習データとなるジェスチャを行う代わりに、ソフトウェア上で録画したジェスチャデータを利用して複数のセンサ配置を試すことができる。ユーザはソフトウェアを用いてバーチャル環境上で自由にセンサを配置し、Kinect を用いて 3D 深度情報としてソフトウェア上に記録された実世界の変形と配置したセンサ間の距離を計測する。得られた距離データを時系列データとして記録しジェスチャ識別を行う。また、ソフトウェア上で得られたデータで生成された識別器を用いて実世界に配置したセンサで得られたデータを識別することで、ソフトウェア上で学習データの蓄積ができることを確認した。

なお、この論文は文献 [8] の内容に実験を追加したものである。

## 2. 関連研究

### 2.1 計算機上での実世界システムのシミュレーション

本研究のように計算機上で実世界の形状や状況を復元し、シミュレーションを行う研究がされている。

Ino らは、Dhaibaworks と呼ばれるデジタルヒューマンモデルを生成できるソフトウェアを使用して、モーションキャプチャによって計測されたデータから手の CG モデルを計算機上で再現し、再現された CG モデルの撮影画像と手の関節などの 3 次元位置情報を利用して畳み込みニューラルネットワークで学習を行うことで、手の把持姿勢推定を行っている [2]。Kanaya らは、現実世界の地図データ入力、実データに基づく歩行者モビリティの再現、センサモデルを導入した HumanS シミュレータを開発し、地図上でセンサの配置や種類を変化させた場合のシミュレーションをした [3]。Yuan らは、バーチャル空間上でバーチャルキャラクターを歩行させたときの一人称カメラ画像と姿勢を学習し、実世界上でユーザの頭部に固定したカメラ画像とバーチャル空間での学習データを組み合わせて歩行時の姿勢を推定している [10]。

これらの研究は計算機上でのシミュレーションを行うことで、実世界での学習を行う必要がなく、費用や時間がかからないという利点がある。本研究ではシミュレーションの対象を光センサとし、実機での試行錯誤をしなくてもソフトウェア上で自由なセンサ配置をすることができ、学習データの蓄積もすることができるシステムを開発した。

### 2.2 光センサによる人間の身体動作計測

本研究では、シミュレーションする実世界センサとして光センサの一種である測距センサを想定している。光センサによって人間の体の動きを計測し、推定や識別をする研究が行われている。

AffectiveWear [4] は眼鏡フレーム上に配置された反射型光センサを用いて、顔の皮膚表面とセンサの間の距離を計測している。人間は表情によって顔の皮膚変形が異なることを利用して、距離データから機械学習を行い表情を識別している。Nakamura らは眉間の皮膚の動きを反射型光センサを用いて計測し、その寄り具合を推定することで情報操作に応用している [6]。Nakatsuma らはリストバンド型のデバイスに反射型光センサを複数配置することで手の甲に触れている指先の位置を推定している [7]。Miyata らは測距センサを用いたバンド型デバイスを物体に取り付けることで、センサと物体を持つ手との距離から手の把持姿勢推定を行っている [5]。若林らは測距センサを直線状に並べ、その上にかざした手の位置と手のひらの角度を推定し、非接触でタッチパネルのように座標と角度情報を入力するデバイスを作成した [11]。

本研究では光センサの一種である測距センサを想定したシミュレーションを行っている。測距センサの前で手によるジェスチャを行うとセンサと手の間の距離が変化することを利用して、ジェスチャ検出および識別をしている。

## 3. 提案手法

### 3.1 全体のシステム構成

本研究では実世界センサとしてよく用いられている測距センサを対象にしてソフトウェアを開発した。図 1 に本システムの流れを示す。本システムではソフトウェアにおける学習データを蓄積するフェーズと実世界に配置した測距センサで得られたデータを入力としてジェスチャ識別を行

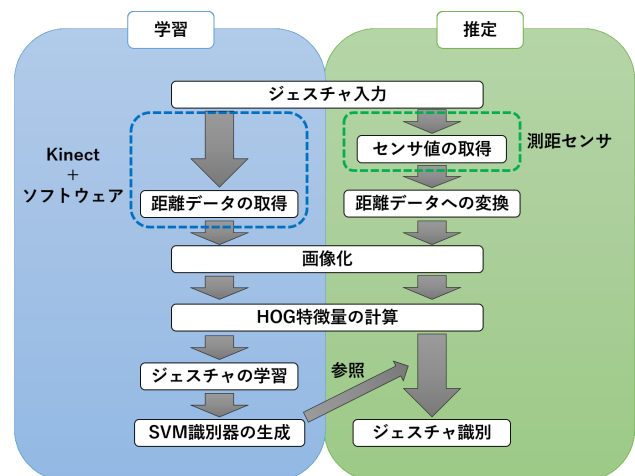


図 1 全体のシステム構成

Fig. 1 Flow of our system.

うフェーズに分かれている。学習データを蓄積するフェーズでは、Kinect の前でジェスチャを行いソフトウェア上で時系列の距離データを取得する。取得されたデータは画像に変換し、Histograms of Oriented Gradients (HOG) 特徴量を算出する。得られた特徴量のデータを入力として Support Vector Machine (SVM) で学習を行い識別器を生成する。

実世界にソフトウェア上と同様に配置された測距センサの前でも Kinect の前で行ったジェスチャと同じ種類のジェスチャを行い、センサ値を取得する。取得されたセンサ値は距離の値に変換され、画像にしてから HOG 特徴量を計算する。得られた特徴量は、ソフトウェア上で取得したデータで生成された識別器を参照することによりジェスチャ識別を行う。

### 3.2 ソフトウェアの構成

本研究では実世界の変化を Kinect v2 で 3D 深度情報として取得し、Unity を用いてバーチャル環境内で実世界の形状を復元した。

まず Kinect の Depth 座標系を実世界の空間座標系 (Unity の座標系) に変換した。変換された 3D 深度情報を Unity 上で 3D メッシュとして復元した。この復元は Unity 上でより高精細なメッシュを構成するため、Kinect で撮影された映像の中央の  $1/4 \times 1/4$  の範囲で行った。図 2 にソフトウェアのキャプチャ画面を示す。背面のテキストチャが、Kinect を用いて記録された 3D 深度情報を 3D メッシュとして形状復元したものであり、この図では手のジェスチャをキャプチャしている。

作成されたメッシュと同じバーチャル環境内にセンサとなるオブジェクトをユーザが適当な位置に配置できるようになっており、そのセンサとメッシュの間の距離を計測することによって距離の値をソフトウェア上で取得した。距離の値はコライダとの衝突判定ができる Raycast 関数を使った二分探索アルゴリズムで求めた。二分探索は Kinect

の Depth の分解能である 1 mm 単位になるまで続け、センサの測定可能な範囲に 3D メッシュが存在しない場合はセンサの測定可能な最大距離を距離の値とした。取得された距離の値は Unity の画面上でのグラフ表示 (図 2) と CSV ファイルでの保存をした。ソフトウェア上に配置する個々のセンサには 0 から順にセンサ番号を付けることができ、グラフ表示の際は上からセンサ番号順に表示される。

### 3.3 実世界でのセンサデータ取得

実世界上で配置する測距センサは SHARP 社製 GP2Y0E02A を用いた。測距センサは赤外線 LED と Position Sensitive Detector (PSD) によって構成されている。このセンサは赤外光を照射し物体に反射した光を PSD で受光する。受光した位置をもとに三角測量の原理でセンサから物体までの距離を計測する。測距センサの計測可能距離はセンサを基準にして 4 cm から 50 cm である。このセンサはセンサ値と距離の関係が線形であるため、センサ値から距離を  $Distance = 4.0 + (2.2 - SensorValue \times 3.3 / 1,023) / 0.036$  の式を用いて算出し、時系列データとして記録している。測距センサはマイクロコントローラ (Arduino Pro Mini 3.3V) に接続されパソコンにセンサデータを送信する。センサ値取得の時間間隔がソフトウェアと測距センサで異なるため、時間間隔が長いソフトウェアに合わせて測距センサのデータを抜き出して CSV ファイルに保存した。

### 3.4 距離データから画像への変換とジェスチャ識別

ソフトウェアと測距センサでそれぞれデータを取得する際、ジェスチャの開始タイミングは設置したセンサのうちいずれか 2 個のセンサの距離データが 30 cm より小さくなったときとした。また、得られた距離データに対し RC フィルタをかけた。使用したフィルタの式は  $y[i] = 0.4x[i] + 0.6y[i - 1]$  (ただし、 $x$ : 取得データ、 $y$ : 時系列データフレームとする) となっている。

ジェスチャ識別を行うために距離データを画像化した。Inaba らは足圧中心座標を推定するため、画像化したセンサデータから HOG 特徴量を抽出し機械学習を行っている [1]。本研究では同様の方法を採用した。フィルタリングした距離データを画像化するため、各フレームにおいて距離データを 1 列に並べ時間軸にそって結合していき、1 つのジェスチャを行った場合の距離データを格納した行列を生成した。距離データを正規化し、0~255 の画素値へと変換することでグレースケール画像を生成した。ソフトウェアから得られたデータで生成されたグレースケール画像の HOG 特徴量を入力として学習を行い、SVM 識別器を生成した。生成した識別器を用いて測距センサから得られたデータで生成されたグレースケール画像の HOG 特徴量を入力としてジェスチャ識別をした。

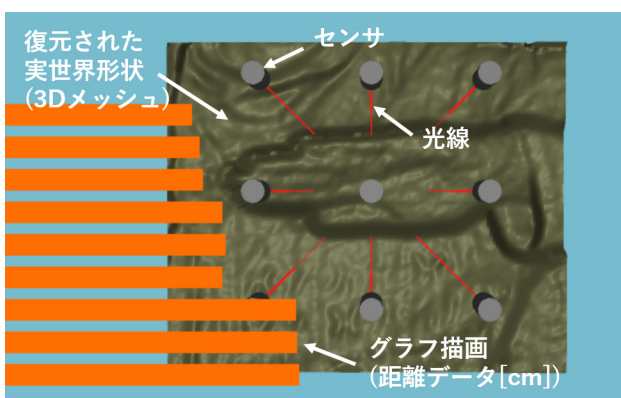


図 2 ソフトウェアのキャプチャ画面  
Fig. 2 Overview of developed software.

## 4. 評価実験

### 4.1 概要

本研究で実装したソフトウェア上でセンサを配置し、蓄積した学習データを実世界上に配置したセンサで計測した結果の識別に用いることができるかを検証するために精度評価実験をした。本実験は平面に取り付けた3×3のセンサ配置(図3左)とPhilips社のHue Goライト(直径15cmの半球状)の周囲を囲むような17個のセンサ配置(図4左)に対して行われた。ソフトウェア上でも実世界上に設置した測距センサと同じ配置になるようにセンサを並べた(図3右, 図4右)。次に, Kinectの前でジェスチャを数種類行い距離データを取得した。また, 同じジェスチャを実世界上に設置した測距センサの前でも行い, センサデータから距離データを取得した。識別したジェスチャの種類と生成画像の例を図5(平面), 図6(ライト)に示す。各ジェスチャはKinectと測距センサの前で10回ずつ行われ, 1回のジェスチャにつき平面の場合は10フレーム, ライトの場合は15フレーム取得し, 画像化した。ソフトウェア上で取得した各データセットは1つをテストデータ, 残りの9つを学習データとするLeave-one-out交差検証を行い識別率を算出した。また, ソフトウェア上で取得したデータで生成された識別器を用いて測距センサで取得した各ジェスチャ10回分のデータがどのジェスチャに分類されるかを検証した。

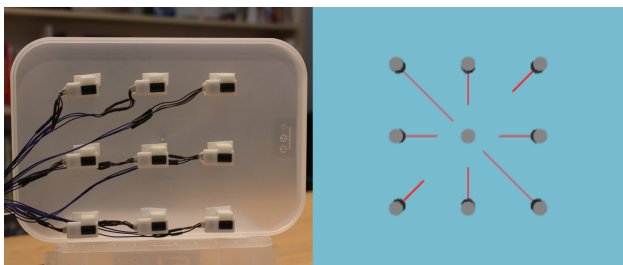


図3 平面にセンサを取り付けた場合(左:実世界, 右:ソフトウェア)

Fig. 3 Sensors attached to the board (Left: in the real world, Right: in the software).

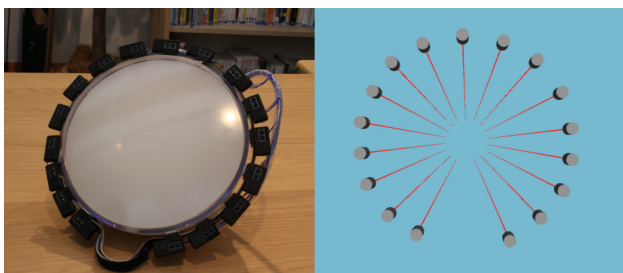


図4 ライトにセンサを取り付けた場合(左:実世界, 右:ソフトウェア)

Fig. 4 Sensors attached to the light (Left: in the real world, Right: in the software).

### 4.2 結果

ソフトウェア上で取得したデータを用いて学習と識別をした場合の識別率を図7左(平面), 図8左(ライト)に示す。このときの識別精度は平面にセンサを取り付けた場合平均で96.0%, ライトにセンサを取り付けた場合平均で100.0%となった。ソフトウェアのデータで生成された識別器を用いて測距センサで取得した各ジェスチャデータを識別した結果を図7右(平面), 図8右(ライト)に示す。このときの識別精度は平面にセンサを取り付けた場合平均で80.0%, ライトにセンサを取り付けた場合平均で81.7%となり, どちらも8割以上の識別率が得られた。図7右の「上」から「下」のジェスチャが「下」から「上」のジェスチャに誤識別されたり, 図8右の「右」から「左」のジェスチャが「左」から「右」のジェスチャに誤識別されたこ

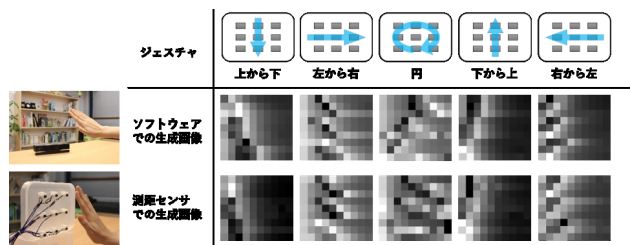


図5 識別したジェスチャの種類と生成画像の例(平面)

Fig. 5 Types of identified gesture and examples of generated images (board).

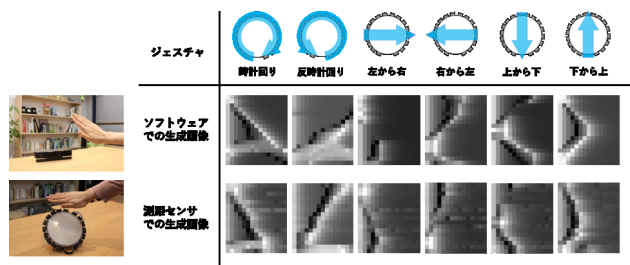


図6 識別したジェスチャの種類と生成画像の例(ライト)

Fig. 6 Types of identified gesture and examples of generated images (light).

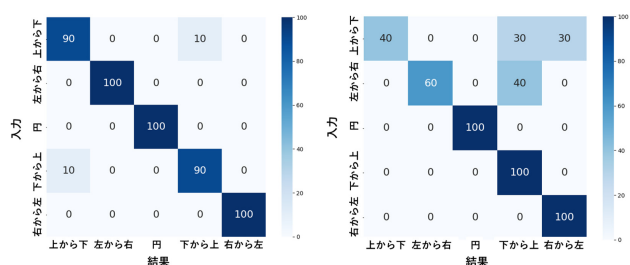


図7 平面にセンサを取り付けた場合の識別結果(左:ソフトウェア, 右:実世界に設置した測距センサ)

Fig. 7 Identification results when sensors attached to the board (Left: using data acquired with the software, Right: using data acquired by the distance-measuring sensors).

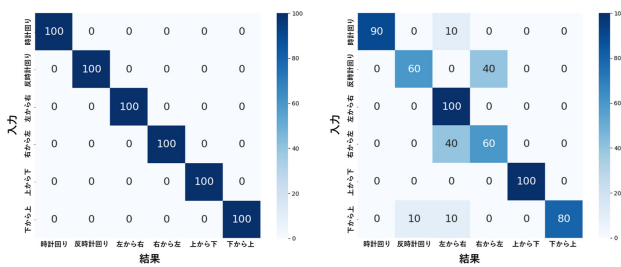


図 8 ライトにセンサを取り付けた場合の識別結果 (左: ソフトウェア, 右: 実世界に設置した測距センサ)

Fig. 8 Identification results when sensors attached to the light (Left: using data acquired with the software, Right: using data acquired by the distance-measuring sensors).

とは、図 5 や図 6 を見るとフレーム数よりも少ないフレームでジェスチャが終えられていることが影響していると考えられる。そのため、ソフトウェアの処理速度を上げることで 1 回のジェスチャに対して取得できるフレーム数が増加し、識別率も上昇すると考えられる。

### 5. 制約と今後の課題

ソフトウェア上では外乱光による影響を受けないが、実世界上に配置した測距センサは外乱光による影響を受けセンサにノイズが乗るため、影響を与えないようなフィルタシステムを組み込むことで解決できると考えられる。本実験ではソフトウェア側の処理速度が遅く約 0.30 秒間隔で距離データを取得していることから、比較的ゆっくりとしたジェスチャで識別をした。測距センサ側はより速い速度でデータを得ることができるため、ソフトウェアの処理速度を上げることにより、より動きの速いジェスチャの識別ができると考えられる。本研究では、ソフトウェア上で取得したデータを学習データとして、実世界上に配置したセンサで取得したデータの識別ができることを確認した。そこで今後の課題として、ユーザがセンサ数やセンサを置く範囲を指定することで、センサ配置を探索し、識別率が高くなるセンサの最適配置を提示するシステムの開発を考えている。

### 6. 結論

実世界センサの配置をデザインできるソフトウェアの開発を行い、ソフトウェア上で取得したデータで生成した識別器を用いて実世界上に配置したセンサで取得したデータの識別可能性を検証した。ソフトウェア上では Kinect の深度情報をもとにバーチャル世界に復元された実世界の変形と、同じバーチャル世界に配置されたセンサとの間の距離を計測することで距離データを取得した。ソフトウェア上で計測したデータで生成された識別器を用いて実世界上に配置したセンサで取得したデータを識別した結果、8 割以上の識別率を得られた。今後の課題として、ソフトウェ

アの処理速度を上げることや、識別率が高くなるセンサ配置を提示するシステムの開発を考えている。

謝辞 本研究は、JST AIP-PRISM JPMJCR18Y2 および JST さきがけ JPMJPR17J4 の支援を受けた。

### 参考文献

- [1] Inaba, K., Murai, A. and Sugiura, Y.: Center of Pressure Estimation and Gait Pattern Recognition Using Shoes with Photo-reflective Sensors, *Proc. 30th Australian Conference on Computer-Human Interaction, OzCHI '18*, pp.224-228, ACM (online), DOI: 10.1145/3292147.3292189 (2018).
- [2] Ino, K., Ienaga, N., Sugiura, Y., Saito, H., Miyata, N. and Tada, M.: Grasping Hand Pose Estimation from RGB Images Using Digital Human Model by Convolutional Neural Network, *Proc. 9th Int. Conf. and Exh. 3D Body Scanning and Processing Technologies*, pp.154-160 (online), DOI: 10.15221/18.154 (2018).
- [3] Kanaya, T., Hiromori, A., Yamaguchi, H. and Higashino, T.: HumanS: A Human Mobility Sensing Simulator, *2012 5th International Conference on New Technologies, Mobility and Security (NTMS)*, pp.1-4 (online), DOI: 10.1109/NTMS.2012.6208740 (2012).
- [4] Masai, K., Sugiura, Y., Ogata, M., Kunze, K., Inami, M. and Sugimoto, M.: Facial Expression Recognition in Daily Life by Embedded Photo Reflective Sensors on Smart Eyewear, *Proc. 21st International Conference on Intelligent User Interfaces, IUI '16*, pp.317-326, ACM (online), DOI: 10.1145/2856767.2856770 (2016).
- [5] Miyata, N., Honoki, T., Maeda, Y., Endo, Y., Tada, M. and Sugiura, Y.: Wrap & Sense: Grasp Capture by a Band Sensor, *Proc. 29th Annual Symposium on User Interface Software and Technology, UIST '16 Adjunct*, pp.87-89, ACM (online), DOI: 10.1145/2984751.2985713 (2016).
- [6] Nakamura, H. and Miyashita, H.: Control of Augmented Reality Information Volume by Glabellar Fader, *Proc. 1st Augmented Human International Conference, AH '10*, pp.20:1-20:3, ACM (online), DOI: 10.1145/1785455.1785475 (2010).
- [7] Nakatsuma, K., Shinoda, H., Makino, Y., Sato, K. and Maeno, T.: Touch Interface on Back of the Hand, *ACM SIGGRAPH 2011 Posters, SIGGRAPH '11*, p.39:1, ACM (online), DOI: 10.1145/2037715.2037760 (2011).
- [8] Saito, A., Kawai, W. and Sugiura, Y.: Software to Support Layout and Data Collection for Machine-Learning-Based Real-World Sensors, *HCI International 2019 - Posters*, Stephanidis, C. (Ed.), pp.198-205, Cham, Springer International Publishing (2019).
- [9] Sato, M., Poupyrev, I. and Harrison, C.: Touché: Enhancing Touch Interaction on Humans, Screens, Liquids, and Everyday Objects, *Proc. SIGCHI Conference on Human Factors in Computing Systems, CHI '12*, pp.483-492, ACM (online), DOI: 10.1145/2207676.2207743 (2012).
- [10] Yuan, Y. and Kitani, K.: 3D Ego-Pose Estimation via Imitation Learning, *Computer Vision - ECCV 2018*, Ferrari, V., Hebert, M., Sminchisescu, C. and Weiss, Y. (Eds.), pp.763-778, Cham, Springer International Publishing (2018).
- [11] 若林航佑, 河合 純, 金田重郎: 赤外線測距センサを用いた座標と角度による感覚的入力装置の提案, 研究報告コンシューマ・デバイス&システム (CDS), Vol.2011,

No.5, pp.1-7 (オンライン), 入手先 (<https://ci.nii.ac.jp/naid/170000065748/>) (2011).



齋藤 彩音 (学生会員)

2019年慶應義塾大学理工学部情報工学科卒業。同年同大学大学院理工学研究科修士課程進学，在籍中。



河合 航

2019年東京大学工学部機械情報工学科卒業。同年同大学大学院情報理工学系研究科知能機械情報学専攻修士課程進学，在籍中。



杉浦 裕太 (正会員)

2013年慶應義塾大学大学院メディアデザイン研究科博士後期課程修了。博士(メディアデザイン学)。慶應義塾大学大学院メディアデザイン研究科特任助教，産業技術総合研究所特別研究員，慶應義塾大学情報工学科助教，

2018年4月より同専任講師。ヒューマンコンピュータインタラクションに関する研究に従事。