

データ駆動型実世界計測におけるセンサ配置とデータ収集を支援するシステムの開発

齋藤 彩音^{1,a)} 河合 航² 杉浦 裕太^{1,b)}

概要：エンタテインメントシステムを構築する上で重要となる実世界センサデバイスは配置や個数，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，センサ配置の検討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シミュレータを開発した。シミュレータ上で，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した。その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

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，システムで捉えたい実世界の変化を考えた上で，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。例えば，Touché [8] はセンサに触れたときの周波数応答を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での学習データを蓄積するのは労力がかかる。

エンタテインメントシステムにおいては，ジェスチャをはじめとした身体の動きを計測することでゲームコントローラとして利用できるため，ジェスチャを識別しやすいセンサ配置を検討する必要がある。

本研究は実世界センサの配置をデザインできるシミュレータソフトウェアの開発とシミュレータにおける学習

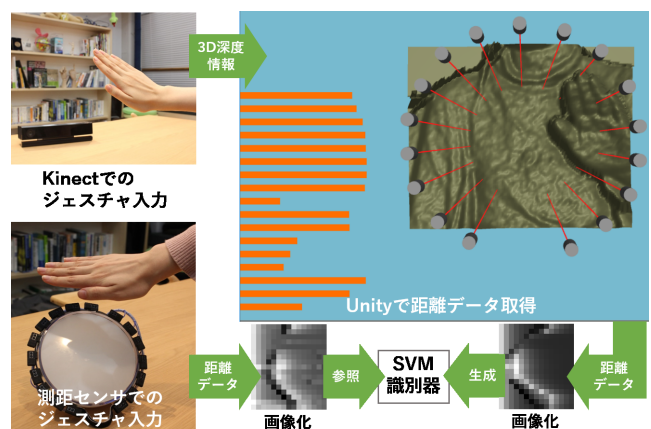


図 1 システムの流れ

データの蓄積の支援を目的とする。本研究では測距センサを対象にしてシミュレータを開発した。ユーザはシミュレータ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 深度情報としてシミュレータ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。本研究ではジェスチャ識別を行うため，得られた距離データを時系列データとして記録する。また，シミュレータ上で学習データの蓄積ができることを確認するために，シミュレータ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。

¹ 慶應義塾大学

² 東京大学

a) ayane-3110@keio.jp

b) sugiura@keio.jp

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た [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、センサモデルの導入をした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行った [3]。Yuanらは、バーチャル空間上でバーチャルキャラクタ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た [9]。

これらの研究は計算機上で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測距センサとし、実機での試行錯誤をしなくても、シミュレータ上で自由なセンサ配置と学習データの蓄積が可能な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

光センサを用い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する。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た。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、眉間の寄せ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た [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た [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 [10]。

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

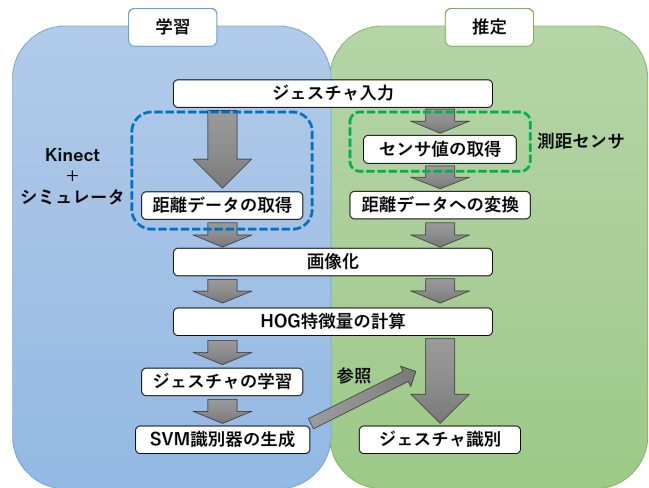


図2 全体のシステム構成

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する。

3. センサ配置とデータ収集を支援するシステム

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シミュレータを開発した。図2に本システムの流れを示す。本システムではシミュレータ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シミュレータ上で時系列の距離データを取得する。取得されたデータを画像に変換し、HOG (Histograms of Oriented Gradients) 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SVM (Support Vector Machine) で学習を行い識別器を生成する。

シミュレータ上と同じセンサ配置になるように実世界上で測距センサを配置した。Kinectの前で行ったジェスチャと同じ種類のジェスチャを実世界上に配置された測距センサの前で行い、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、画像に変換してからHOG特徴量を計算する。得られた特徴量を入力として、シミュレータ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 シミュレータの構成

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

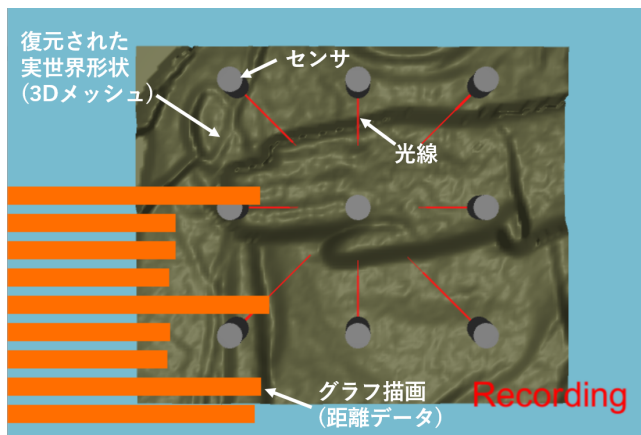


図 3 シミュレータのキャプチャ画面

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。

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シミュレータ上で取得した。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 の画面上でグラフとして表示 (図 3) し、CSV ファイルとして保存を行った。

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 PSD (Position Sensitive Detector) 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物体までの距離の関係が線形であるため、センサ値から距離を $Distance = 4.0 + (2.2 - SensorValue * 3.3/1023)/0.036$ 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 3.3 V) 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シミュレータと測距センサで異なるため、時間間隔が長いシミュレータに合わせて測距センサのデータを抜き出し 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

シミュレータ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 より小さくなった時とした。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。シミュレータ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。

4. 評価実験

4.1 概要

本研究で実装したシミュレータ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 4 左) と Philips 社の Hue Go ライト (直径 15 cm の半球状) の周囲を囲むような 17 個のセンサ配置 (図 5 左) に対して行われた。シミュレータ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 4 右, 図 5 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 6 (平面)、図 7 (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。シミュレータ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、シミュレータで取得したデータを用いて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識別されるかを検証した。

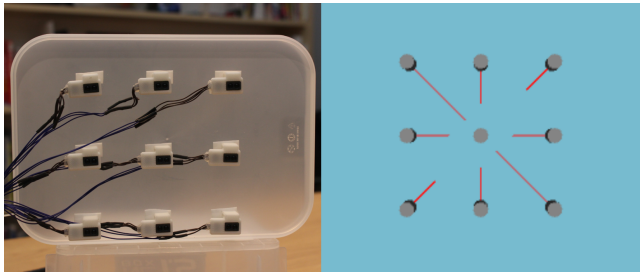


図 4 平面にセンサを取り付けた場合 (左: 実世界, 右: シミュレータ)

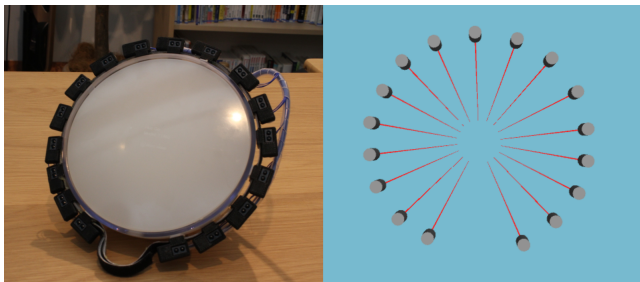


図 5 ライトにセンサを取り付けた場合 (左: 実世界, 右: シミュレータ)

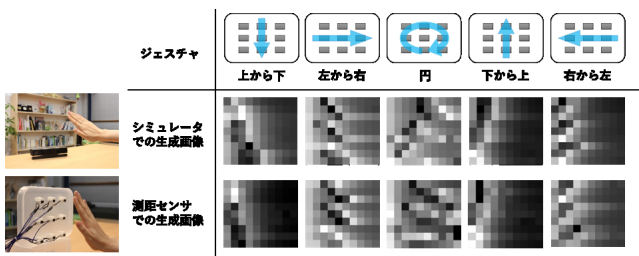


図 6 識別したジェスチャの種類と生成画像の例 (平面)

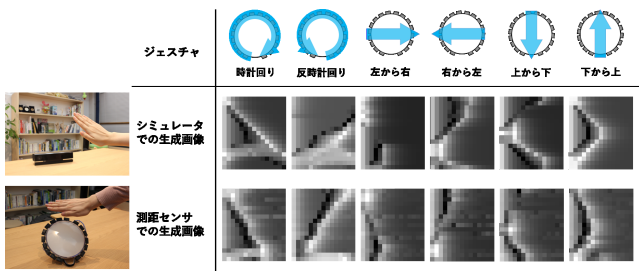


図 7 識別したジェスチャの種類と生成画像の例 (ライト)

4.2 結果と考察

シミュレータ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 8 左 (平面), 図 9 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. シミュレータ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 8 右 (平面), 図 9 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 8 右の「上」から「下」のジェスチャが「下」から「上」のジェ

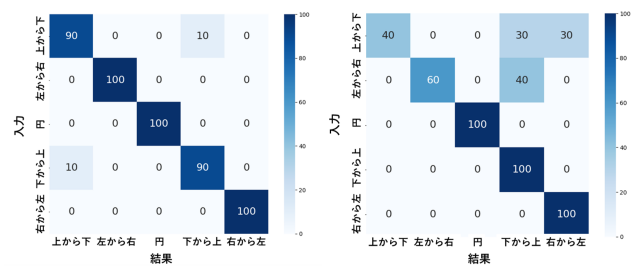


図 8 平面にセンサを取り付けた場合の識別結果 (左: シミュレータ, 右: 実世界に設置した測距センサ)

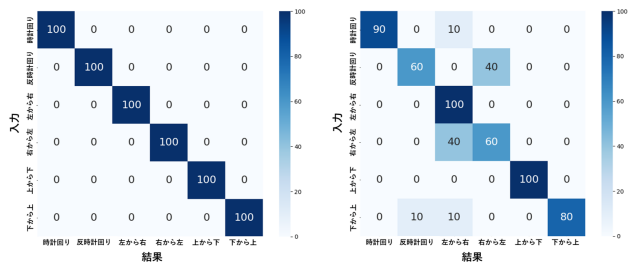


図 9 ライトにセンサを取り付けた場合の識別結果 (左: シミュレータ, 右: 実世界に設置した測距センサ)

スチャに誤識別されたり, 図 9 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とは, 図 6 や図 7 を参照す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, シミュレータ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

シミュレータ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ノイズの影響を低減できると考えられる. 本実験ではシミュレータ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, シミュレータ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. 今後の課題としてユーザがセンサ数を指定すると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.

6. 結論

実世界センサの配置をデザインできるシミュレータソフトウェアの開発を行い, シミュレータ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. シミュレータ上では 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。シミュレータ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、シミュレータ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 PRESTO JPMJPR17J4 の支援を受けたものです。

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, *Proceedings of the 30th Australian Conference on Computer-Human Interaction, OzCHI '18*, New York, NY, USA, ACM, pp. 224–228 (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, *Proceedings of 9th Int. Conf. and Exh. on 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, *Proceedings of the 21st International Conference on Intelligent User Interfaces, IUI '16*, New York, NY, USA, ACM, pp. 317–326 (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, *Proceedings of the 29th Annual Symposium on User Interface Software and Technology, UIST '16*, New York, NY, USA, ACM, pp. 87–89 (2016).
- [6] Nakamura, H. and Miyashita, H.: Control of Augmented Reality Information Volume by Glabellar Fader, *Proceedings of the 1st Augmented Human International Conference, AH '10*, New York, NY, USA, ACM, pp. 20:1–20:3 (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*, New York, NY, USA, ACM, pp. 39:1–39:1 (online), DOI: 10.1145/2037715.2037760 (2011).
- [8] Sato, M., Poupyrev, I. and Harrison, C.: Touché: Enhancing Touch Interaction on Humans, Screens, Liquids, and Everyday Objects, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '12*, New York, NY, USA, ACM, pp. 483–492 (online), DOI: 10.1145/2207676.2207743 (2012).
- [9] Yuan, Y. and Kitani, K.: 3D Ego-Pose Estimation via Imitation Learning, *Technical Paper at European Conference on Computer Vision, ECCV '18*, Munich, Germany, LNCS, pp. 763–778 (2018).
- [10] 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).