

# カメラと加速度センサを用いたデバイスペアリング方式 の提案とその評価

長友 誠<sup>1</sup> 油田 健太郎<sup>2</sup> 岡崎 直宣<sup>2</sup> 朴 美娘<sup>1</sup>

**概要：**無線技術を用いたデバイスペアリングを行う手法には、受信信号強度(RSS: Received Signal Strength)を用いた手法がある。しかし、RSSは些細な環境の変化で大きく変化するため、安定した精度が得られない。一方で、赤外線カメラでデバイスを持つ人の手の動きを認識するペアリング手法がある。これは、端末の移動は認識できるが、傾きを認識することができない課題がある。本論文では、パーテーションや部屋区切りでカメラを搭載したサーバと加速度センサを搭載したデバイスをペアリングする手法を提案する。カメラでデバイスの画面上に表示したマーカーを認識し、端末の動きをマーカーの動きに代替したデータとデバイスから取得した加速度データの類似度をサーバで計算する。その類似度が閾値以上であればペアリングが成功する手法である。加えて、提案した手法の実証実験としてカメラからデバイスを1.5mと2.0m離し、丸の字と∞の字で動かした際の類似度を調べる実験と、カメラに映っていない第3者が不正にペアリングを行うことができるかどうかの実験を行なった。実験の結果、カメラからの距離が1.5m、2.0mと離れれば、類似度にばらつきが出てくることが分かり、端末の動かし方によっても変化することがわかった。また、第2の実験では、平均的に見れば不正なペアリングが行われない類似度の閾値を設定できるが、類似度の標準偏差が大きくなり、今後は類似度の標準偏差を小さくするように改良を行う必要が出てきた。

## Proposal and Evaluation of Pairing Method with Camera and Accelerator

MAKOTO NAGATOMO<sup>1</sup> KENTARO ABURADA<sup>2</sup> NAONOBU OKAZAKI<sup>2</sup>  
MIRANG PARK<sup>1</sup>

### 1. はじめに

近年、IoT機器やスマートフォン、タブレット端末の増加に伴い、無線を用いた近接情報に基づくデバイス同士のペアリングを行う機会が増えている。本論文ではそのペアリングを、長期間のペアリングを持続することが想定された長期的なペアリングと、ある限定された期間のみペアリング可能なアドホックなペアリングに分類する。

長期的なペアリングの例として、アクセスポイント(AP)とスマートフォンのペアリングが挙げられる。これは、一度ペアリングが完了すればスマートフォンが接続範囲外出た後に再び範囲内に戻ると自動的にAPと接続されるため、長期間ペアリングが持続する。しかし、ペアリングを行う際にキーとなる値をスマートフォンに入力する必要があるため、ペアリングに時間がかかる。さらに、中間者攻撃(MITM: Middle-In-The-Middle attack)がある。上記のAPの例では、攻撃者が正規のAPのSSIDを入手し、正規のAPが送出するチャンネルの妨害電波を出し続け、正規のAPが接続できなくなる。その後に正規のAPと同じSSIDを持つ偽のAPを攻撃者が用意する。それを別のチャンネルで送出し、ペアリングを行うデバイスを偽のAPに誘導し、盗聴を行う[1]。

一方、アドホックな会議で会議に参加している人のモバイル端末などに無線で会議資料を配布する場面が挙げられる。この場合は会議資料を持つPCと会議に参加している

人のモバイル端末のみで一時的なペアリングを行う必要がある。また、会議が終われば自動的にそのPCとモバイル端末のペアリングを解除する必要がある。問題点として、第3者によるなりすまし[1]が挙げられる。部屋外の第3者が部屋内にいる人になりすまし、会議資料を持つPCと無線で通信を行うことで不正に会議資料を入手することができる。

本研究では、上記の例のようなアドホックなペアリングにおいて、パーテーションや部屋単位でペアリングができ、同時になりすましに強いペアリング手法を提案することを目的とする。

現在、受信信号強度(RSS: Received Signal Strength)を用いたペアリング手法[2], [3]の研究が盛んであるが、RSSは少しの環境の変化で大きく変化するため、部屋単位で安定したペアリングができない。また、2つのデバイスを同時に動かすことでそれらのペアリングを行う方式[4], [5]がある。これは、各デバイスが加速度センサから得られる加速度データからデータの送受信の暗号化に必要な共通鍵を直接生成することでペアリングを行う。しかし、鍵の生成成功率が70%ほどと不確実である。端末の加速度センサとそのバイブルーション機能を持つデバイスをペアリングする手法[6]では、デバイス同士を密着させて振動でbit情報を伝えるため、ペアリング距離が短い欠点がある。さらに、カメラを搭載したデバイスとLEDなどの可視光を発する機能を搭載したデバイスをペアリングする手法

1 神奈川工科大学  
Kanagawa Institute of Technology

2 宮崎大学  
University of Miyazaki

[7], [8]では、LED の点滅によってカメラが取得する画像の明るさを変化させ、bit 情報を送信することでペアリングを行う。しかし、ペアリング可能な距離が数十センチほどしかない。その他に、赤外線カメラを搭載したサーバと加速度センサを搭載したデバイスをペアリングする方式[9], [10]がある。これらは、赤外線カメラで人の手の動きのデータを取得すると同時に、その手に持っているデバイスの加速度データを取得し、その 2 つのデータの類似度を算出することでペアリングを行う手法である。しかし、これらは人の動きを検出できる特殊なカメラを必要とすると共に、端末の傾きを検出しにくいという課題がある。

そこで本研究では、通常のカメラを備え付けた PC と加速度センサ搭載したモバイル端末をペアリングする手法を提案する。本手法では、モバイル端末の画面上に表示したマーカーをカメラで認識し、モバイル端末の動きとみなす手法である。カメラが認識しやすいマーカーを読み取ることで端末の動きや傾きを検出しやすくなり、ペアリング精度の向上が期待できる。

以降、2 章で関連研究を紹介し、3 章で提案手法のシステムモデルとペアリング手順について述べる。4 章で提案手法の実装と実験を行い、5 章でその実験結果と考察を述べる。最後に 6 章で全体のまとめと今後の課題について述べる。

## 2. 関連研究

この章では、無線通信における RSS やカメラ、加速度センサを用いたペアリングについて紹介する。

### 2.1 受信信号強度(RSS)を用いた近接検出

Amigo[2]では、デバイス同士の無線通信の RSS の差の絶対値の平均、指數の平均、RSS ベクトルのユークリッド距離を特徴量とし、機械学習を用いてデバイス同士の近接検出をすることでペアリングを行う方式を提案している。複数の学習アルゴリズムを組み合わせることにより識別を向上させている。結果として、デバイス間が 5cm である場合、攻撃者（盗聴者）が 3m 以上離れている時に攻撃を検出できている。

縣ら[3]は、複数の AP からの RSS を用いた部屋単位でのデバイスペアリングを提案している。AP からデバイスに送られるビーコンフレームの 2.4 GHz 帯と 5 GHz 帯の RSS に加えて、ビーコンフレームを得ることができたアクセスポイントの集合を特徴量とする。それを *k*-Nearest Neighbor (KNN) 法を用いて同一の 10m 四方の部屋にデバイスが存在するかどうかの識別を行なっている。結果として平均識別率が 99.3 % の精度が得られたが、シナリオによっては 88 % に低下している。これは RSS が周りの少しの環境の変化で値が変化するためである。

よって、RSS のみを用いたペアリング手法では部屋やパーテーションで区切られた空間内のみでペアリングを行うことは難しい。

### 2.2 加速度センサを用いたペアリング手法

Smart-Its Friends[4] や Daniel ら[5]は、ペアリングを行いたい 2 つのデバイスを同時に振ることでペアリングを行う手法を提案している。ただし、両方のデバイスとも加速度センサを搭載していると仮定する。特に、[5]では、得られた加速度データそのものからペアリング後のデータ送受信の暗号化に必要な共通鍵を生成する手法である。各々のデバイスについて、加速度データを複数に分割し、部分鍵を生成した後にそれらを合成することで共通鍵を生成する。同じ鍵が生成されればペアリング成功となる。結果として、平均 13 bit の共通鍵が約 70% の成功率で生成できたが、関係の無い第 3 者が同時にデバイスを振った際に同じ鍵が生成される可能性がある。

Vibreaker[6]は、加速度センサに加えて、デバイスのバイブレーション機能も用いたペアリング手法である。まず 2 つのデバイスを密着させる。次に、片方のデバイスが 200 ms の振動を行なったか行なってないかで 1 か 0 の 1bit の情報をもう片方のデバイスに送る手法である。振動を受けたデバイスは加速度センサの変化により 1bit の情報をエンコードし、PIN コードを受け取る。それによってデバイス同士のペアリングが完了する。この手法では、通常の PIN に必要な 14 bit の情報量を送る際に、情報を送る合図の 3 bit を追加して送信する。よってペアリングに必要な時間は合計 3.4 s となる。問題点は、デバイス同士が振動を感じる距離に存在する必要があるため、ペアリング距離が極端に短い点である。

### 2.3 カメラを用いたペアリング手法

部屋やパーテーションでペアリング可能範囲を区切る際に、電磁波の中でも壁を貫通しない可視光を用いて区切る手法が有効である[7], [8]。Nitesh ら[7]は、LED ライトなどを搭載したデバイスとカメラを搭載したデバイスをペアリングする手法を提案している。まず、ペアリングを行う 2 つのデバイスが無線を用いて DH 鍵交換方式[11]を使い、両方のデバイスに共通鍵を生成する。その後に片方のデバイスが視覚情報 (LED ライトの点滅など) を用いてカメラを通じてもう片方のデバイスに DH 鍵のハッシュ値を送信する。受信デバイスは DH 鍵のハッシュ値を視覚情報で受け取った値と照合することでペアリングを行う。また、Alexis ら[8]による LED ライトの点滅で直接パケットの bit 情報を送る手法もある。しかし、どちらもペアリング可能距離が数十センチと短い制限がある。

また、赤外線カメラ (Kinect) を用いたデバイスペアリング手法も研究されている[9], [10]。山口ら[9]は、デバイス自体から得られた加速度データを無線で Kinect を搭載したサーバに送り、サーバがそれと手の動きのデータと照合することによりペアリングを実現している。この手法は、人が手にデバイスを持っていると仮定し、デバイスの動きを手

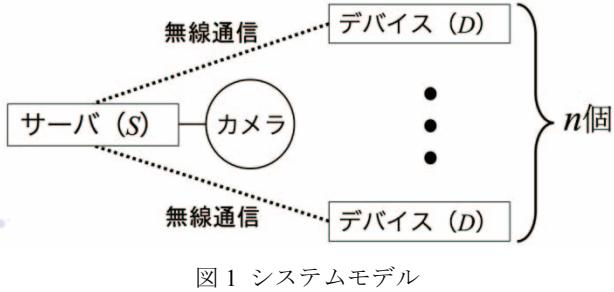


図 1 システムモデル

の動きに代替している手法である。Mahsan ら[10]は、スマートフォンを持っている人とタッチパネルを操作する人を Kinect で認識する手法を提案している。しかし、[9], [10]どちらもデバイスの傾きを検出しておらず、さらに特殊なカメラである Kinect を用いる必要がある。

そこで 本研究では、加速度センサを搭載したモバイル端末と、赤外線カメラなどの特殊なカメラを搭載していない、通常のカメラを搭載した PC をペアリングする手法について検討する。デバイスの検出には、デバイスの画面上に映したマーカーをカメラを通じて認識させることにより実現する。

### 3. カメラと加速度センサを用いたペアリング方式の提案

#### 3.1 システムモデル

本論文で提案する方式のシステムモデルを図 1 に示す。以下そのシステムモデルの構成要素を示す。

##### (1) 認証サーバ (S)

カメラを搭載している、正規のデバイスを認証するサーバである。デバイスから無線通信で受信した加速度データと、カメラから得た画像データからマーカーの変位データを抽出し、それらの類似度を算出する。また、ペアリング結果をデバイスに送信する。

##### (2) カメラ

サーバと接続しており、デバイスから取得したマーカー画像情報をサーバに送信する。

##### (3) デバイス (D)

加速度センサを内蔵したデバイスで、カメラで認識しやすいマーカーを画面に表示する。ペアリング時にデバイスの 3 軸の加速度データを計測し、サーバに送る。システムモデルでは、複数のデバイスがサーバと同時にペアリングを行うことを考え、 $n$  個のデバイスがあるとした。

#### 3.2 ペアリング手順

提案方式のペアリング手順を図 2 に示す。想定するサーバ  $S$  は、例えば WEB カメラを備え付けたノート PC で、デバイス  $D$  はモバイル端末とする。以下にペアリング手順を示す。

##### [step 1] マーカーの表示

$D$  はマーカーを表示する。このマーカーはカメラが数メ

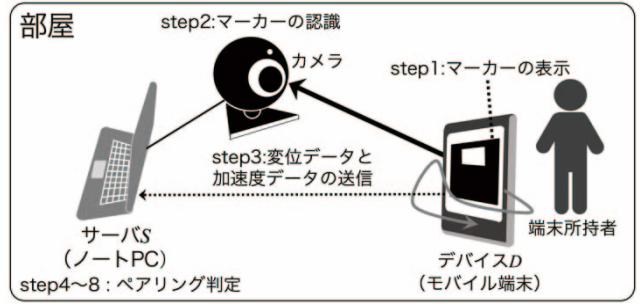


図 2 ペアリング手順

ートル先でも認識可能な単純なマーカーとする。

##### [step 2] マーカーの認識

$S$  はカメラを通じて  $D$  の画面上のマーカーを認識する。

##### [step 3] 変位データと加速度データの取得

$D$  の所有者はマーカーをカメラに映したままデバイスを任意のモーションで動かし、その後に端末から得られた  $x$ ,  $y$ ,  $z$  軸の加速度データとその取得時刻の組みの集合  $\alpha$  を無線で  $S$  に送信する。 $S$  はデバイスが動いている間、カメラ画像上のマーカーの変位データ  $x$ ,  $y$  とその取得時間の組みの集合  $\beta$  を取得する。 $\alpha$  と  $\beta$  はそれぞれ次のように表される。

$$\alpha = \{(\alpha_1^x, \alpha_1^y, \alpha_1^z, t_1^\alpha), \dots, (\alpha_m^x, \alpha_m^y, \alpha_m^z, t_m^\alpha)\} \quad (1)$$

$$\beta = \{(x_1, y_1, t_1^\beta), \dots, (x_n, y_n, t_n^\beta)\} \quad (2)$$

ここで、各  $\alpha_i^x, \alpha_i^y, \alpha_i^z (i \in \{1, \dots, m\})$  は時刻  $t_i^\alpha$  で取得した  $x$ ,  $y$ ,  $z$  軸の加速度を表し、 $x_j, y_j (j \in \{1, \dots, n\})$  は時刻  $t_j^\beta$  の画像上のマーカーの座標を表す。

##### [step 4] ノイズの除去

加速度データの重力加速度とノイズの除去を、高速フーリエ変換 (FFT: Fast Fourier Transform) を用いてハイパス、ローパスフィルタで行う。

##### [step 5] データの補間

カメラデータの各  $x$  軸,  $y$  軸のデータに対し、3 次元スプライン補間を用いて以下の関数で近似する。

$$S_j(t) = a_j + b_j(t - t_j^\beta) + c_j(t - t_j^\beta)^2 + d_j(t - t_j^\beta)^3 \quad (3)$$

また、 $a_j, b_j, c_j, d_j$  を決めるために、次の 5 つの条件が必要となる。

$$1. \quad S_j(t_j^\beta) = w_j$$

$$2. \quad S_j(t_{j+1}^\beta) = S_{j+1}(t_{j+1}^\beta) = w_{j+1}$$

$$3. \quad S'_j(t_{j+1}^\beta) = S'_{j+1}(t_{j+1}^\beta)$$

$$4. \quad S''_j(t_{j+1}^\beta) = S''_{j+1}(t_{j+1}^\beta)$$

$$5. \quad S''_0(0) = S''_{n-1}(t_j^\beta) = 0$$

ただし、 $w \in \{x, y\}, j \in \{1, \dots, n-1\}$  である。

### [step 6] 速度データへの変換

$S$  は[step 5]で得られたマーカーのデータを微分し, 速度データ  $\beta'$  に変換する. また, [step4]で得られた加速度データを積分し, 速度データ  $\alpha'$  に変換する. 各速度データ  $\alpha', \beta'$  を以下のように表す.

$$\alpha' = \{(v_1^x, v_1^y, v_1^z, t_1^\alpha), \dots, (v_m^x, v_m^y, v_m^z, t_m^\alpha)\} \quad (4)$$

$$\beta' = \{(x_1', y_1', t_1^\beta), \dots, (x_m', y_m', t_m^\beta)\} \quad (5)$$

ただし,  $\beta'$ においては, [step 5]行なったスプライン補間の関数から加速度データの時刻と同じデータを補間し, データの個数を加速度データの個数と同じ  $m$  個にした後に微分を行う.

### [step 7] データの正規化

カメラから取得されたデータ  $\alpha'$  の単位は pixel/s であり, 加速度センサから得られた速度データ  $\beta'$  の単位は m/s であるため, そのまま類似度を算出しても妥当な結果を得ることができない. よって, 各軸のデータ列に対して, 取得時刻個数の次元 (加速度データの取得個数:  $m$  次元) のベクトルとみなし, その大きさを 1 にすることで正規化を行う. 各  $\alpha', \beta'$  を正規化したデータ  $\tilde{\alpha}, \tilde{\beta}$  を以下の式で表す.

$$\tilde{\alpha} = \{(\tilde{v}_1^x, \tilde{v}_1^y, \tilde{v}_1^z, t_1^\alpha), \dots, (\tilde{v}_m^x, \tilde{v}_m^y, \tilde{v}_m^z, t_m^\alpha)\} \quad (6)$$

$$\tilde{\beta} = \{(\tilde{x}_1, \tilde{y}_1, t_1^\beta), \dots, (\tilde{x}_m, \tilde{y}_m, t_m^\beta)\} \quad (7)$$

ただし, 各  $\tilde{v}_i^w, \tilde{u}_i$  ( $w \in \{x, y, z\}, u \in \{x, y\}, i \in \{1, \dots, m\}$ ) を -1 ~ 1 の値を取るように以下の式で算出する.

$$\tilde{v}_w^i = v_w^i / \sum_{k=1}^m v_k^2 \quad (8)$$

$$\tilde{u}_i = u'_i / \sum_{k=1}^m u'_k^2 \quad (9)$$

### [step 8] 類似度の算出

$\tilde{\alpha}, \tilde{\beta}$  から類似度を算出する. 類似度の算出については次の節で述べる. 閾値  $\theta$  以上であればペアリングが成立し, DH の鍵交換方式で暗号化通信のための共通鍵を生成する. そうでなければペアリング不成立とする.

上記の手順において, カメラで複数のデバイス上のマーカーを読み取ることで一対複数のデバイスペアリングが可能となる.

### 3.3 類似度算出方法

本論文では, 3.2 節の[step 8]で類似度を計算する際に,  $\tilde{\alpha}$  の  $x, y$  軸のデータと  $\tilde{\beta}$  の  $x, y$  軸のデータの類似度を算出する. その後に  $x$  軸,  $y$  軸それぞれの類似度の平均を計算し, 全体の類似度とする. 類似度の計算には, 以下の 4 種類を用いて類似度を算出した. なお, 加速度データの  $z$  軸データや, マーカーの傾きを考慮した類似度算出については, 今後検討する予定である.

#### (1) 単純なマッチング

$x, y$  軸のデータの各取得時刻における値の差を全て足したのちに, その個数で割った平均を算出する. 値が小さい

ほど類似度が高いことを意味する. 類似度は以下の式で表す.

$$s_w = \frac{1}{m} \sum_{k=1}^m |\tilde{v}_k^w - \tilde{w}_k| \quad (10)$$

ただし,  $w \in \{x, y\}$  である.

#### (2) DP マッチング

類似度を以下の漸化式から求める.  $g(m, m)$  の値を算出し, それを類似度とする. 算出された値は単純なマッチングと同様に値が小さいほど類似度が高いことを意味する.

$$g(i, j) = \min \begin{cases} g(i-1) + c(i, j) \\ g(i-1, j-1) + 2c(i, j) \\ g(i, j-1) + c(i, j) \end{cases} \quad (11)$$

ただし,  $w \in \{x, y\}$  で, コスト関数  $c$  を  $c(i, j) = |\tilde{v}_i^w - \tilde{w}_j|/m$ ,  $g(0, 0) = d(\tilde{v}_1^w, \tilde{w}_1) = c(0, 0)$  とする.

#### (3) 相関係数

各  $x$  軸,  $y$  軸での相関係数は, 以下の式で算出する.

$$r_w = \frac{\sum_{i=1}^m (\tilde{v}_i^w - \bar{v}_w)(\tilde{w}_i - \bar{w})}{\sqrt{(\sum_{i=1}^m (\tilde{v}_i^w - \bar{v}_w)^2)(\sum_{i=1}^m (\tilde{w}_i - \bar{w})^2)}} \quad (12)$$

ただし,  $w \in \{x, y\}$ ,  $\bar{v}_w = \sum_{k=1}^m \tilde{v}_k^w$ ,  $\bar{w} = \sum_{k=1}^m \tilde{w}_k$  とする.  $r_w$  は -1 から 1 の値をとり, 正の値が大きければ正の相関, 負の値が大きければ負の相関があると判断できる.

#### (4) Jaccard 係数

まず, 各時刻のデータ  $\tilde{v}_i^w, \tilde{w}_i$  ( $i \in \{1, \dots, m\}$ ) を 0.05 間隔で  $\tilde{v}_i^w, \tilde{w}_i$  と量子化し, その後に量子化した値とその時間の組みの集合  $\tilde{A} = \{(\tilde{v}_i^w, t_i^\alpha) | i \in \{1, \dots, m\}\}$ ,  $\tilde{B} = \{(\tilde{w}_i, t_i^\beta) | i \in \{1, \dots, m\}\}$  を考え, 次式で集合の距離を定義する. ただし,  $w \in \{x, y\}$  である.

$$distance = \frac{|\tilde{A} \cap \tilde{B}|}{|\tilde{A} \cup \tilde{B}|} \quad (13)$$

$distance$  は 0 から 1 の値をとり, 大きければ大きいほど類似度が高いことを示す.

## 4. 評価実験と考察

### 4.1 提案手法の実装

3 章で提案したペアリング手法の評価実験用のプロトタイプを実装した. 開発環境として, サーバ側の開発言語は Python3 を使って開発した. マーカーの認識には OpenCV のライブラリ ArUco[12]を用い, デバイス側は Java 言語を用いて Android Studio 上で開発を行なった. 実験機器はサーバをノート PC の MacBook Pro 15 inch 2017, デバイス側はモバイル端末 Nexus 5X をそれぞれ用いた. また, カメラと加速度データを取得する機器はそれぞれ内蔵されている機器を用いた.

図 3 に端末側のアプリケーションを示す. アプリケーションを立ち上げると画面に認識用のマーカーが表示され, PC 側のアプリケーションと, ルータを介した Wi-Fi の無線通信を開始する. また, アプリケーション起動中は, 画面

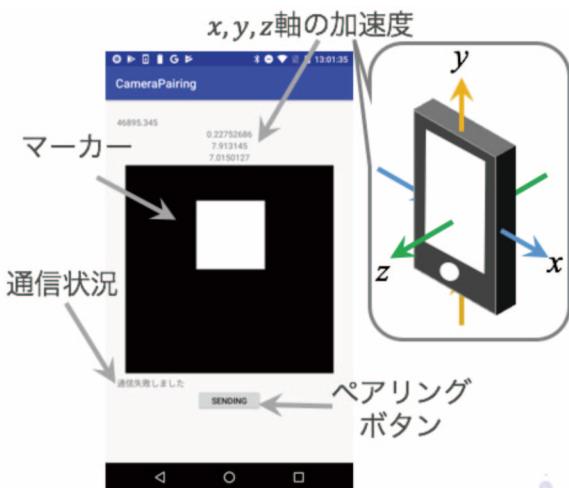


図3 端末側のアプリケーション

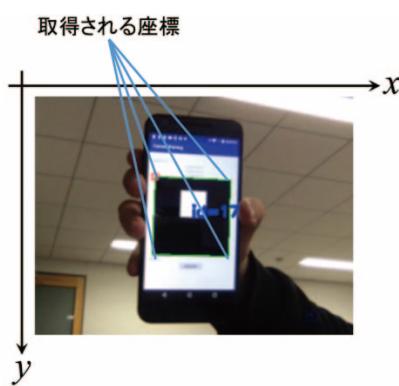


図4 PC側のアプリケーション

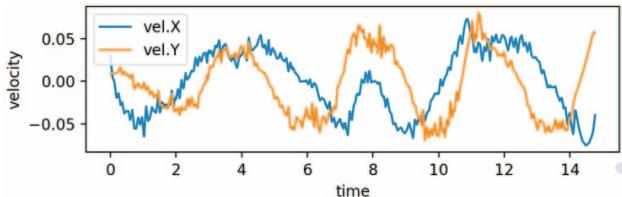


図5 カメラの変位データから変換した速度データ

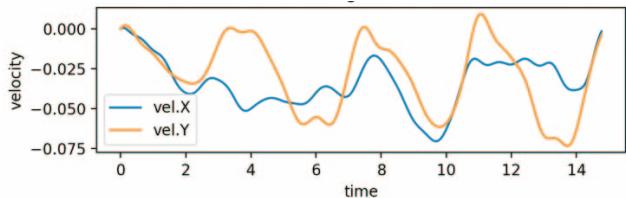


図6 加速度データから変換した速度データ

上に  $5.5\text{ cm} \times 5.5\text{ cm}$  のマーカーを表示する。ペアリングボタンを押すと  $x, y, z$  軸の加速度データとその取得時刻の取得が始まり、カメラに向けて端末を動かした後に再びペアリングボタンを押すとPC側にそのデータを全て送信する。図4にPC側のアプリケーションを示す。アプリケーションを起動すると、カメラが取得した画像が表示される。カメラがマーカーを読み取ると、画像上に映っているマーカーの4つの角の座標が取得できる。本実験では、 $640 \times 360$  ピクセルの大きさの画像をカメラから取得するように設定した。また、端末側から加速度とその取得時刻のデータを取得すると、カメラから得た変位とその取得時刻のデータとの類似度を3.3節に従い算出する。

図5, 6にペアリング時のデータの例を示す。横軸はペアリング処理開始からの時間を表しており、vel.X, vel.Yは  $x$  軸と  $y$  軸で取得されたデータを表す。速度データに変換する前のデータ個数はそれぞれ、変位データが239個、加速度データが419個であった。2つのグラフを見比べると、外形が一致している箇所が複数見ることができるため、どちらも同じ速度データに変換できていることがわかる。しかし、 $x$  軸の2~6秒の間で形が大きく異なっている。これは、加速度センサデータの方に大きくノイズが入ってしまったからであると考える。

#### 4.2 実験1：条件による類似度変化の確認実験

3章で提案した手法において、デバイス（本実験ではモバイル端末であるNexus 5X）のカメラからの距離、モーションによって類似度が変化する可能性がある。よって、実験1として、その確認実験を行なった。被験者は神奈川工科大学に在籍する8名（全員20代）で、以下の手順で行なった。

- (1) 端末をカメラから（1.5 m, 2.0 m）離す。
- (2) 端末側のアプリケーションを起動し、端末側の画面上にマーカーを表示する。
- (3) 端末を床に垂直な方向に立てるように持ち、カメラに平行にした状態で端末を（円、∞の字）の形に沿って約15秒動かす。
- (4) 3.2節のstep3～step8の手順により(3)で取得したデータから類似度を計算する。
- (5) (1)～(4)を5回行う。

今回の実装環境では、PCがマーカーを読み取ることができる限界距離が約2.0mであった。よって、今回の実験は、読み取りができる限界距離から50cm刻みの1.5mと2.0mの距離で行なった。

表1に被験者8名の類似度の平均を示す。単純なマッチング、DPマッチングは数値が低いほど類似性があり、相関係数とJacard係数は数値が高いほど類似性があると判断される。結果として、1.5m地点での円と∞の字のモーションについては、∞の字のモーションの類似度が高いと算出された。しかし、2.0m地点ではJacard係数のみ円のモーションの類似度が高いと算出された。円のモーションについて、相関係数のみ2.0m方が類似性が低いと判断された。また、∞の字でDPマッチングのみが2.0mの方の類似度が低い結果となった。この結果から、近い距離では単純なモーションほど類似度が低く、マーカーが認識できる限界

表1 類似度算出結果

(モーション, カメラからの距離)	単純なマッチング	DP マッチング	相関係数	Jacard 係数
(円, 1.5m)	0.038	0.022	0.412	0.265
(∞の字, 1.5m)	0.037	0.019	0.449	0.281
(円, 2.0m)	0.036	0.020	0.379	0.296
(∞の字, 2.0m)	0.031	0.020	0.502	0.282

距離に近くなるに連れて類似度にばらつきが出てきたことが分かった。

#### 4.3 実験2：不正なペアリングが可能かどうかの実験

提案手法において、カメラの範囲外にいるなりすまし者がペアリングを行う正規の端末の動きを真似して不正にペアリングが成功する可能性がある。もし、類似度にばらつきがあれば、正規の人よりなりすまし者の類似度が高くなる可能性がある。その不正なペアリングが可能かどうかを確かめるための実験を行った。今回は神奈川工科大学に在籍する3名（全員20代）を被験者とした。なお、PC側をAlice、正規の端末を持つ人をBob、カメラに映る範囲外のなりすまし者をEveとし、以下の手順で実験を行った（図7）。

- (1) Bobの端末をAliceの持つカメラから1.5m離す。また、Eveはカメラに映る範囲外に移動する。
- (2) BobとEveは端末を床に垂直な方向に立てるように持つ。Bobは端末の画面上のマーカーがカメラに映るようにし、カメラに平行にする。その後、端末(円、∞の字)のモーションで約15秒動かす。動かしている間、EveはBobの動きを真似して端末を動かす。
- (3) Aliceは(2)で得た変位データと、それぞれ本人と盗聴者の加速度データの類似度を3.2節のstep3～step8の手順により計算する。
- (4) (1)～(3)を5回繰り返す。

以上の手順（2つのモーションで10回）を、被験者3名が本人と盗聴者の両方を行える組み合わせ:  $3 \times 3 - 3 = 6$ 通りで行い、合計60回実験を行なった。

表2になりすまし者より正規にペアリングを行う人の方が類似度が高いと判定された割合を示す。結果として、相関係数を用いれば、円、∞の字、両方どちらも含めて84%、80%と他の類似度算出方法と比べて高い結果となった。表3に本人と盗聴者の各類似度の平均を示す。どの類似度算出方法も平均を見れば閾値となる値を、0.39, 0.23, 0.65, 0.29と決めれば本人と盗聴者を識別できることが分かる。しかし、全体の標準偏差が0.022, 0.016, 0.175, 0.103であったため、類似度が大きくぶれることも分かった。

表2 盗聴者より正規の人の類似度が高かった割合

モーション	単純なマッチング	DP マッチング	相関係数	Jacard 係数
円	76%	36%	84%	68%
∞の字	56%	44%	80%	68%

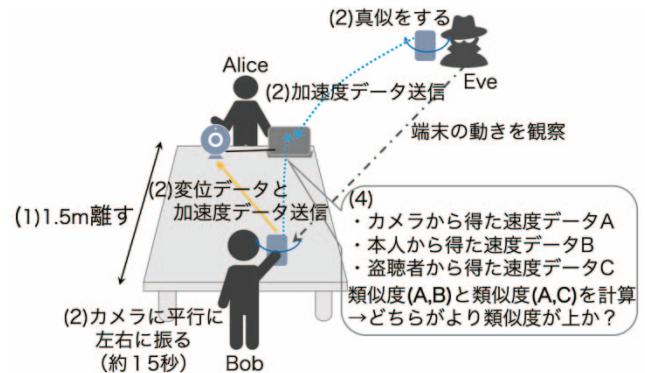


図7 不正なペアリングが可能かどうかの実験

#### 5. おわりに

本論文では、カメラを備え付けたサーバとマーカーが表示できる画面と加速度センサを備えたデバイスのペアリング方式の提案を行った。また、基礎的な実験として、x,y軸方向のデータのみを用いた実証実験を行った。具体的には、距離とモーションによって類似度が変化するかどうかの実験と、カメラ範囲外のなりすまし者が正規な人のモーションを真似して不正なペアリングが出来るかどうかの実験を行った。

1つ目の実験の結果から、カメラとマーカーの距離がマーカー認識限界距離（今回の実験では約2.0m）に近いほど類似度にばらつきが出ることが分かった。また、2つの目の実験の結果から類似度の平均で本人と盗聴者を分けることができる閾値を確認できたが、類似度のばらつき（標準偏差）が大きくなることが分かったため、安定したペアリングができない結果となった。

今後は、3次元のモーションやマーカーの傾きを類似度算出に用いることによって、類似度の標準偏差を減らす方法の検討をすることと、マーカーを複数同時に読み取ることによる一対複数のデバイスペアリングが可能か検証する必要がある。

#### 参考文献

- [1] 公衆無線 LAN 利用に係る脅威と対策  
入手先<<https://www.ipa.go.jp/files/000051453.pdf>>(2018.5.12参照)
- [2] A. Varshavsky, A. Scannell, A. LaMarca, and E. de Lara, “Amigo: Proximity-based authentication of mobile devices,” Proceedings of the 9th international conference on Ubiquitous computing (UbiComp ’07), pp. 253-270, 2007.

表 3 本人と盗聴者の各類似度の平均

モーション	単純なマッチング		DP マッチング		相関係数		Jacard 係数	
	正規な人	なりすまし者	正規な人	なりすまし者	正規な人	なりすまし者	正規な人	なりすまし者
円	0.039	0.045	0.023	0.028	0.67	0.53	0.31	0.26
∞の字	0.034	0.039	0.021	0.024	0.72	0.62	0.30	0.28
円, ∞の字	0.037	0.042	0.023	0.026	0.70	0.58	0.30	0.27

- [3] 縣侑吾, 洪志勲, 大槻知明, “複数アクセスポイントからのデュアルバンド信号の受信信号強度に基づく部屋レベルの Proximity 検出,” 電子情報通信学会技術研究報告, vol.115, no. 437, pp. 15-20, 2016.
- [4] L. E. Holmquist, F. Mattern, and B. Schiele, P. Alahuhta, M. Beigl, H. W. Gellersen, “Smart-Its Friends: A Technique for Users to Easily Establish Connections between Smart Artefacts,” Proceedings of the 3rd international conference on Ubiquitous Computing (UbiComp '01), pp. 116-122, 2001.
- [5] D. Bichler, G. Stromberg, and M. Huemer, “Innovative Key Generation Approach to Encrypt Wireless Communication in Personal Area Networks,” Proceedings of the 50th International Global Communications Conference, 2007.
- [6] S. A. Anand and N. Saxena. “Vibreaker: Securing Vibrational Pairing with Deliberate Acoustic Noise,” Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks (WiSec '16), pp. 103-108, 2016.
- [7] N. Saxena, J. Erik. Ekberg, and K. Kostianien, “Secure Device Pairing Based on a Visual Channel: Design and Usability Study,” vol. 6, issue. 1, pp. 28-38, 2010.
- [8] A. Duque, R. Stanica H. Rivano, and A. Desportes, “Unleashing the power of LED-to-camera communications for IoT devices,” Proceedings of the 3rd Workshop on Visible Light Communication Systems, pp. 55-60, 2016.
- [9] 山口徳郎, 立澤茂, 野中雅人, “モバイル端末センサと環境カメラを活用した端末ペアリング方式の提案,” 電子情報通信学会技術研究報告, vol. 112, no. 106, pp. 29-33, 2012.
- [10] M. Rofouei, A. D. Wilson, A. J. B. Brush, and S. Tansley, “Your Phone or Mine? Fusing Body, Touch and Device Sensing for Multi-User Device-Display Interaction,” Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 149-158, 2012.
- [11] W. Diffie and M. E. Hellman, “New directions in Cryptography,” IEEE Transactions on Information Theory, vol. 22, no. 6, pp. 644-654, 1976.
- [12] ArUco  
入手先<[https://docs.opencv.org/3.2.0/d5/dae/tutorial\\_aruco\\_detecton.html](https://docs.opencv.org/3.2.0/d5/dae/tutorial_aruco_detecton.html)>(2018.5.12 参照)