Vol.2019-CSEC-85 No.7
Vol.2019-IOT-45 No.7
2019/5/23

マーカーを用いたカメラと加速度センサによるセキュアデバイスペアリング手法の評価

長友 誠[1]　油田 健太郎[2]　岡崎 直宣[2]　朴 美娘[1]

概要：近年，Wi-Fi，Bluetooth などの無線技術を搭載したモバイルデバイスや IoT デバイスの増加に伴い，デバイス同士が無線で通信を行う機会が増加している．その一例として臨時的な会議においてプレゼンターの持つノート PC から出席している人の持つモバイルデバイスに電子資料を配布するシーンが考えられる．しかし，通信を暗号化していなければ，会議室の外にいるなりすまし者に資料が渡る可能性がある．そこで，以前我々はディスプレイにマーカーを表示したモバイルデバイスを動かしたとき，PC のカメラで取得したマーカーの変位データと，デバイスの加速度データとの類似度を算出することでペアリングを行う手法を提案し，実験を行った結果，マーカーの認識できる限界距離が 1.0m であり，また類似度のばらつきが大きく安定したペアリングができない結果となった．そこで本稿では，内蔵カメラより解像度の高い外付けカメラを用い，カメラと端末の距離，ディスプレイの大きさを変化させたときの類似度を複数の方法で算出した．その結果，データ値の差による類似度算出方法ではスマートフォンなどの小さいデバイスを用いると類似度が高くなり，データの相関による類似度算出方法ではタブレットなどの大きいデバイスを用いると類似度が高くなる結果となった．

キーワード：デバイスペアリング，マーカー，カメラ，加速度センサ

Investigation of Secure Device Pairing Method using Marker by Camera and Accelerometer

Abstract: Recently, mobile device and IoT device equipped with wireless technology such as Wi-Fi or Bluetooth is increasing. Along with this, it is also increasing to communicate wirelessly between these devices. For example, there is scene that a presenter distributes meeting material to participants' device. However, there is a possibility that impersonator outside the room obtain the materials. Hence, we proposed device pairing method, which calculates similarity between displacement data of marker displayed on a device from camera and acceleration data from the device. As a result of calculation of similarity, variation of similarity was large, so it is not possible to perform stable pairing. In this paper, we investigated correlation among distance from camera to device, size of device's display using external camera. As a result, the similarity of the small device such as a smartphone was high when the calculation method using the difference between data is used. Also, the similarity of the large device such as a tablet was high when the calculation method when the correlation between data is used.

Keywords: device pairiing, marker, camera, accelerometer

1. はじめに

近年，Wi-Fi，Bluetooth などの無線技術の発展により，それらを搭載したモバイルデバイスや IoT(Internet of Things) デバイスが増加している．それに伴って，これらのデバイスが無線通信を行い，情報を交換する機会が多くなっている．しかし，暗号化を行わない無線通信を行えば，盗聴や中間者攻撃などの攻撃を受ける可能性がある．よって，デバイス同士が無線通信を行う前に，通信の安全性を確保するため通信内容を暗号化する鍵をデバイス間で交換する必要がある．本稿では，そのデバイス間で鍵を交換する手順をデバイスペアリングと呼ぶことにする．

[1] 神奈川工科大学
[2] 宮崎大学

このデバイスペアリングは大きく「長期的なペアリング」と「アドホックなペアリング」の 2 つに分けられる．長期的なペアリングは，デバイス間の接続が長期的なペアリングである．例えば，スマートフォンとアクセスポイントのペアリングがあり，ユーザはスマートフォン上でアクセスポイントのキーを入力することでペアリングを行う．一度ペアリングを行なった後，アクセスポイントの通信範囲外から範囲内にスマートフォンが戻ってきても自動的に接続が行われるが，ユーザはキーを手入力で行う必要があるためペアリングに時間を要する．対照的に，アドホックなペアリングは，ある一定期間だけ接続を行う．例えば，臨時的な会議においてプレゼンターの持つノート PC が参加者の持つデバイスと無線で鍵を交換し，その鍵を使って暗号化した資料をそれらのデバイスに配布するシーンが考えられる．会議が終わればノート PC で鍵を破棄しペアリングを終了することができる．しかし，無線は壁を通り抜けるため，部屋の中に存在するデバイスと部屋の外に存在するデバイスの区別ができず，ノート PC と部屋の外のなりすまし者のデバイスが鍵を交換し，なりすまし者に資料が渡る可能性がある．

よって，本研究では，ペアリングを行う操作が簡単で，部屋などのパーティション区切りで行うことができるデバイス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が研究されている [1], [2]．Amigo[1] では，無線を発しているデバイス同士がお互いに受信する RSS の類似度でお互いが近接しているかを判別する．文献 [2] は，複数のアクセスポイントから受信する RSS を用いて 10〜15 m^2 四方の同じ部屋に PC が存在するか判定する手法である．これらの手法はユーザが意識することなくペアリングを行うことができるが，RSS は周りの物や時間で大きく変化するため，安定的なペアリングが難しい．

一方で，加速度センサを用いたペアリング手法として，2 つのデバイスを同時に振ったとき，同様の加速度データから共通鍵を生成するペアリング手法 [3] も研究されているが，第三者がデバイスの動きを真似すると同じ鍵が生成される可能性がある．また，モバイルデバイスを振動させ，もう片方のデバイスの加速度センサで振動のデータを取得することで PIN コードを送信する手法 [4] も提案されているが，ペアリング可能な距離はほぼ 0m である．

また，カメラを用いたペアリング手法として，デバイスのライトの点滅によってカメラを搭載した PC へ情報を渡すことでデバイスの確認をし，ペアリングを行う手法が研究されている [5]．しかし，この手法はペアリング可能な距離が数十センチと短い問題がある．

さらに，カメラと加速度センサを用いたペアリング手法として，モバイルデバイスを動かしたときに得られる加速度データと Kinect などの赤外線を搭載した PC から得られたユーザの腕の動きのデータを比較することでペアリングを行う手法が提案されている [7]．この手法は特殊なカメラを用いているため，アドホックなペアリングを行う場面には向いておらず，手の動きのデータを取得しているためデバイスの傾きを取得できない問題がある．そこで，通常のカメラを用いて加速度センサを搭載したデバイスのトラッキングを行う手法 [8], [9] も提案されているが，カメラの範囲外のなりすまし者が正規のデバイスの動きを真似することでペアリングができてしまう問題が考えられる．

そこで，以前我々は通常のカメラを搭載した PC と加速度センサを搭載したモバイルデバイス間のペアリング手法を提案した [10]．この手法はデバイスのディスプレイ上にマーカーを表示し，ユーザがデバイスでモーションを描いた際に，デバイスから得られる加速度データとカメラで取得したマーカーの動きのデータの類似度を算出しペアリングを行う．しかし，ノート PC 内蔵のカメラを用いて実験を行なった結果，ペアリング可能な距離が 1m ほどしかなく，また類似度のばらつきが大きい結果となった．

そこで本稿では，以前提案したペアリング手法 [10] において，ノート PC の内蔵カメラより解像度が高い外付けのカメラを用い，デバイスの種類，カメラとデバイスの距離を変化させたときの類似度の確認を複数の手法を用いて行う．

2. 関連研究

2.1 RSS(Received Signal Strength) を用いたペアリング手法

Amigo[1] は RSS を用いたデバイス間のペアリングであり，2 つのデバイスがお互いに近接しているかどうかを，双方の受信する RSS のユークリッド距離などを特徴として機械学習で判定を行う手法である．正規のデバイスとなりすましのデバイス間の距離が約 3m である場合，これらのデバイスの区別が可能となる．

文献 [2] では，2 つの PC が 10-15m 四方の部屋に存在しているかを，これらの PC が周囲の複数のアクセスポイントから得られる RSS を特徴とし，機械学習で判定を行う手法である．2.4G-Hz 帯と 5-GHz 帯の RSS を使用し，実験結果として 96%の精度を確認できている．

上記 2 つの手法は，ユーザが意識することなくペアリングが行われる．しかし，RSS は環境状況により大幅に変化するため，安定したペアリング手法ではない．例えば，同じ部屋内でもパーティションを 1 つ挟めば RSS が大きく変化する．

2.2 加速度センサを用いたペアリング手法

文献 [3] は，加速度センサを搭載した 2 つのデバイスを同じように振り，得られた加速度データからデータの送受信を暗号化するための共通鍵を生成するペアリング手法で

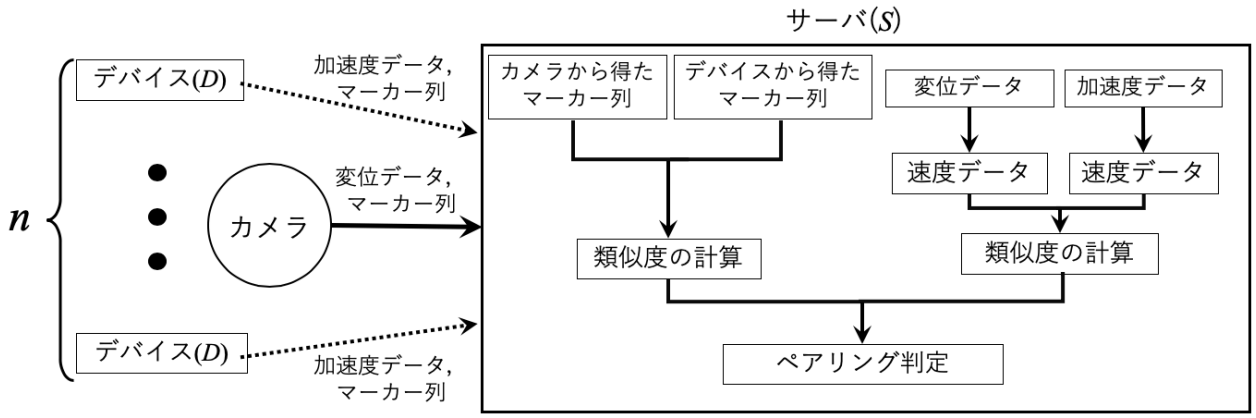

図 1 システムモデル

ある．実験の結果として，13bit の共通鍵を 70%の成功率で生成できている．しかし，この手法では加速度センサのみを用いているため，第三者がペアリングを行うデバイスの動きを真似することによって，不正にペアリングが行われる可能性がある．

Vibreaker[4] は振動機能を持つデバイスと加速度センサを搭載したデバイス間のペアリング手法である．この手法では，デバイス同士を振動が伝わる場所に置き，片方のデバイスが振動でもう片方のデバイスに PIN を送信する．受信側のデバイスは取得する加速度データを用いて PIN を受信する．実験では，17bit を約 3 秒で送ることができている．しかし，この手法はデバイスの振動機能を用いているため，ペアリング距離が極端に短い．

2.3 カメラを用いたペアリング手法

Saxena ら [5] は LED ライトを搭載したデバイスとカメラを搭載したデバイスのペアリング手法を提案している．DH(Diffie-Hellman) 鍵交換 [11] を用いてデバイス間で共通鍵を生成した後，デバイスの近接確認のために，LED ライトの点滅で鍵のハッシュ値をカメラを搭載したデバイスに送信し，そのデバイス内でハッシュ値を確かめることでペアリングを行う手法である．一方で，Alex ら [6] は LED ライトの 1 回の点滅を 1bit としてパケットを送信する手法を提案している．これらの方式は，可視光を用いているため，部屋区切りでペアリングが可能であるが，数十センチの短い距離でしかペアリングができない問題がある．

2.4 カメラと加速度センサを用いたペアリング手法

加速度センサを搭載したモバイルデバイスと Kinect などの赤外線カメラを搭載した PC とのペアリング手法が研究されている [7]．デバイスを持っている手を赤外線カメラで認識し，ユーザはデバイスを動かす．その際に赤外線カメラから得られる手の動きとデバイスから得られる加速度データを比較する．それによってカメラの前にあるデバイスの認証を行うペアリング手法である．しかし，赤外線カメラを使う必要があるためアドホックなシーンで用いることが難しく，また手の動きだけを認識しているため，デバイスの傾きを考慮していない問題がある．

また，加速度センサを搭載したデバイスを通常のカメラを搭載したサーバを用いてトラッキングを行う手法が提案されている [8], [9]．この手法は，カメラに映っている人が動いた際に，身につけているデバイスの加速度センサから得られるデータとカメラから得られる人の動きをマッチングすることでトラッキングを行う．しかし，これらの手法をペアリングに応用すれば，カメラの範囲外にいる第三者が範囲内にいるユーザの動きを真似することで，不正にペアリングが行われる問題が考えられる．

そこで，我々は加速度センサを搭載したモバイルデバイスと通常のカメラを搭載した PC 間のペアリング手法を提案した [10]．この手法はデバイスの画面上にマーカーを表示し，ユーザがデバイスをカメラの前で動かすことで，ペアリングが行われる手法である．しかし，ノート PC 内蔵のカメラを用いて実験を行なった結果，ペアリング距離が 1m ほどしかなく，また，類似度のばらつきが大きい結果となった．

3. マーカーを用いたカメラと加速度センサを使ったデバイスペアリング手法

この章では，以前我々が提案した通常のカメラを搭載した PC と加速度センサを搭載したデバイスのペアリング手法 [10] を紹介する．カメラの前でユーザがマーカーの表示されたデバイスを動かすことでペアリングを行う手法である．この方式は QR コードなどのマーカーより単純なマーカーを使用することで，ペアリング可能な距離を伸ばすことができる．またデバイスの動かし方は人によって特徴が異なるため，カメラの範囲外にいるなりすまし者が正規のユーザの動きを真似しても PC とペアリングすることが難しいと考えられる．以降，この方式のシステムモデルとペアリング手順について述べる．

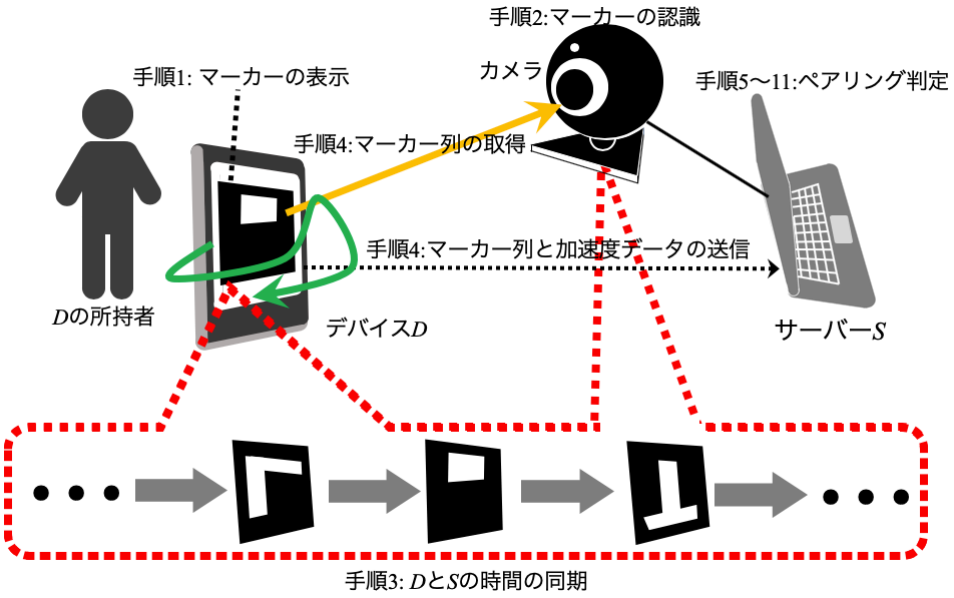

図 2　ペアリング手順

3.1　システムモデル

図 1 にシステムモデルを示す．このシステムはデバイス，認証サーバ，カメラの 3 つから成る．以下にそれらの説明を述べる．

- デバイス (D)
  スマートフォンなどの加速度センサとディスプレイが搭載されたデバイスである．認証時はディスプレイ上にカメラが認識できる単純なマーカーを表示し，一定時間ごとにマーカーの種類が変化する
- 認証サーバ (S)
  カメラが搭載された PC である．このサーバはデバイスから無線で送られてきた加速度データとカメラから得られるデバイス上に表示されたマーカーの動きのデータ (変位データ) の類似度を算出する．同様にデバイスから送られたマーカー列とカメラで取得したマーカー列の類似度を算出する．
- カメラ
  サーバに接続され，サーバに画像情報を送るカメラである．赤外線カメラなどの特殊でない通常のカメラである．

3.2　ペアリング手順

この方式では，ユーザがデバイスを動かしたときに取得できる加速度データとサーバが得るマーカーの変位データの類似度を算出する．その際に，加速度データを積分，変位データを微分することで同じ速度データに変換する．その後，それら 2 つの速度データの類似度を算出する．ペアリング手順は以下の通りである (図 2)．

(1) マーカーの表示
　デバイス D はディスプレイにマーカーを表示する．このマーカーはカメラが認識できる単純なマーカーであり，一定時間ごと (例えば 0.5 秒ごと) にマーカーの種類が変化する．

(2) マーカーの認識
　S は D のディスプレイ上に表示されたマーカーをカメラを通して認識する．

(3) デバイス D とサーバ S の時間の同期
　D のディスプレイ上でマーカーが変化した時間と，S のマーカーの変化を認識した時間を同じ時間とみなすことで双方の時刻を合わせる．

(4) 変位データと加速度データの取得
　ユーザは D のディスプレイ上に表示されたマーカーがカメラに映るように，D を任意に動かす．このとき，D は x, y, z 軸の加速度データと表示したマーカーの種類の列のデータ α を無線で S に送信する．同時に S 側ではカメラから取得した映像上のマーカーの座標とマーカー列のデータ β を取得する．α と β は以下のように表現される．

$$\alpha = \{(t_i^\alpha, a_i^x, a_i^y, a_i^z, m_i^\alpha) | i \in \{1, ..., m\}\} \tag{1}$$

$$\beta = \{(t_j^\beta, (x_j^1, y_j^1), ..., (x_j^4, y_j^4), m_j^\beta) | j \in \{1, ..., n\}\} \tag{2}$$

　ただし，α 内の $a_i^x, a_i^y, a_i^z, m_i^\alpha$ は時刻 t_i^α における，x, y, z 軸の加速度データと表示したマーカーの種類を表す．また，β 内の $(x_j^1, y_j^1), (x_j^2, y_j^2), (x_j^3, y_j^3), (x_j^4, y_j^4), m_j^\beta$ は時間 t_i^β に取得したマーカーの 4 つ角の座標と認識したマーカーの種類を表す．

(5) 加速度データと変位データの修正
　ユーザごとにデバイスを動かす際のデバイスの傾きが

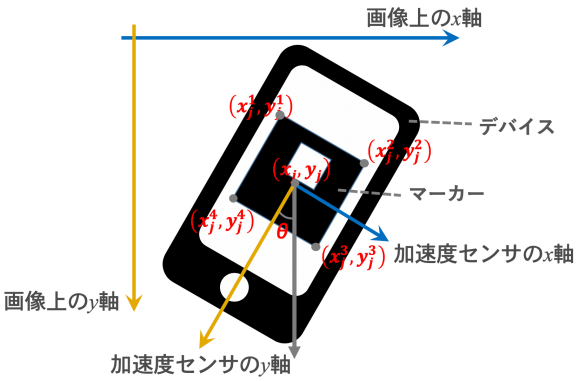

図 3 デバイスの傾きによる加速度データの修正

異なる可能性があるため，S はマーカーの変位データと加速度データをマーカーの傾きにより修正する．修正を行ったデータ α', β' は以下の式で表される．

$$\alpha' = \{((a_i'^x, a_i'^y), m_i^\alpha, t_i^\alpha) | i \in \{1, ..., m\}\} \quad (3)$$

$$\beta' = \{((x_j, y_j), m_j^\beta, t_j^\beta) | j \in \{1, ..., n\}\} \quad (4)$$

ただし，$a_i'^x, a_i'^y$ はマーカーの傾きを θ とした回転行列を用いて以下の式で計算される．

$$\begin{pmatrix} a_i'^x \\ a_i'^y \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} a_i^x \\ a_i^y \end{pmatrix} \quad (5)$$

マーカーの傾き θ の算出方法は，**図 3** のように，マーカーの 4 つ角を用いてマーカーの下方向を求め，その方向と画像上の y 軸方向との角度で算出する．また，β 内の (x_i, y_i) はマーカーの重心の座標であり，以下の式で表される．

$$x_j = \frac{x_j^1 + x_j^2 + x_j^3 + x_j^4}{4}, \quad y_j = \frac{y_j^1 + y_j^2 + y_j^3 + y_j^4}{4} \quad (6)$$

(6) ノイズの除去
サーバ S は FFT(Fast Fourier Transform) を使い，ハイパス，ローパスフィルタを用いて重力加速度，ノイズの除去を行う．

(7) 変位データの補間
変位データと加速度データの個数 n, m について，マーカーの認識に時間を要するため $n < m$ となる．そこで，3 次元スプライン補間を用いて変位データの補間を行い，データ数 m に揃える．

(8) 速度データへの変換
サーバ S は変位データ (x_j, y_j) を微分，加速度データ $(a_i'^x, a_i'^y)$ を積分することで 2 つのデータを速度データ (v_i^x, v_i^y), (x_j', y_j') に変換する．変換後の速度データ α'' と β'' は以下の式で表される．

$$\alpha'' = \{((v_i^x, v_i^y), m_i^\alpha, t_i^\alpha) | i \in \{1, ..., m\}\} \quad (7)$$

$$\beta'' = \{((x_i', y_i'), m_i^\beta, t_i^\beta) | i \in \{1, ..., m\}\} \quad (8)$$

(9) データの正規化
手順 (8) の α'' と β'' は，どちらも速度データであるが単位が異なるため，直接比較できない．そこでこれら 2 つのデータの正規化を行い，比較可能なデータに変換する．変換後のデータ $\tilde{\alpha}, \tilde{\beta}$ は以下の式で表される．

$$\tilde{\alpha} = \{((\tilde{v}_i^x, \tilde{v}_i^y), m_i^\alpha, t_i^\alpha) | i \in \{1, ..., m\}\} \quad (9)$$

$$\tilde{\beta} = \{((\tilde{x}_i, \tilde{y}_i), m_i^\beta, t_i^\beta) | i \in \{1, ..., m\}\} \quad (10)$$

ただし，$\tilde{v}_i^w, \tilde{u}_i (w \in \{x, y\}, u \in \{x, y\})$ を時系列順のベクトルとみなし，そのベクトルの大きさが 1 となるように正規化を行う．それらは以下の式で表される．

$$\tilde{v}_i^w = v_i^w / \sum_{k=1}^{m} (v_k^w)^2, \quad \tilde{u}_i = u_i' / \sum_{k=1}^{m} u_k'^2, \quad (11)$$

これらの値は-1 から 1 の値をとる．

(10) マーカー列の類似度計算
$\tilde{\alpha}$ と $\tilde{\beta}$ から，サーバ S はマーカー列の類似度を計算する．類似度がある閾値以上であれば，手順 (11) へ進む．閾値より小さければペアリング不成功となる．

(11) 速度データの類似度計算
$\tilde{\alpha}$ と $\tilde{\beta}$ から，サーバ S は速度データの類似度を計算する．手順 (10) と同様に，類似度がある閾値以上であればペアリングが成功し，デバイス D とサーバ S の間で DH 鍵交換を行う．その後に，交換した鍵を用いて暗号化通信を行う．

以上，1 台のデバイスとサーバのペアリングを行う手順を述べた．2 台以上のデバイスが同時にペアリングを行う場合は，複数のデバイスからサーバへ加速度データが送られ，カメラから複数の変位データが得られる．そこで手順 (11) におけるマーカーの類似度算出を複数の加速度データと変位データの組み合わせ全てで行う．その中でマーカーの類似度が最も大きい組み合わせを同一のデバイスから得られた加速度データと変位データとみなす．それによって複数デバイスとサーバのペアリングを行うことができる．

3.3 マーカー列の類似度計算

3.2 節の手順 (10) において，サーバ S は以下の式によりマーカー列の類似度 m_s の計算を行う．

$$m_s = \frac{\tilde{\alpha}, \tilde{\beta}\text{内の } m_i^\alpha = m_i^\beta \text{となる個数}}{m(\tilde{\alpha}\text{もしくは}\tilde{\beta}\text{の要素数})} \quad (12)$$

ただし，m_s は 0〜1 の値をとる．

図 4 にマーカー列の類似度計算の例を挙げる．この例では，表示されたマーカーの種類の順番は (1,2,3) であるが，マーカー列は m_7^α と m_7^β のマーカーの種類が異なるため類似度は 1 とならない．このような類似度計算方法を採用した理由は，マーカーの種類の順番を類似度とすれば，カメラがマーカーの種類を誤って認識した際に，マーカーの種

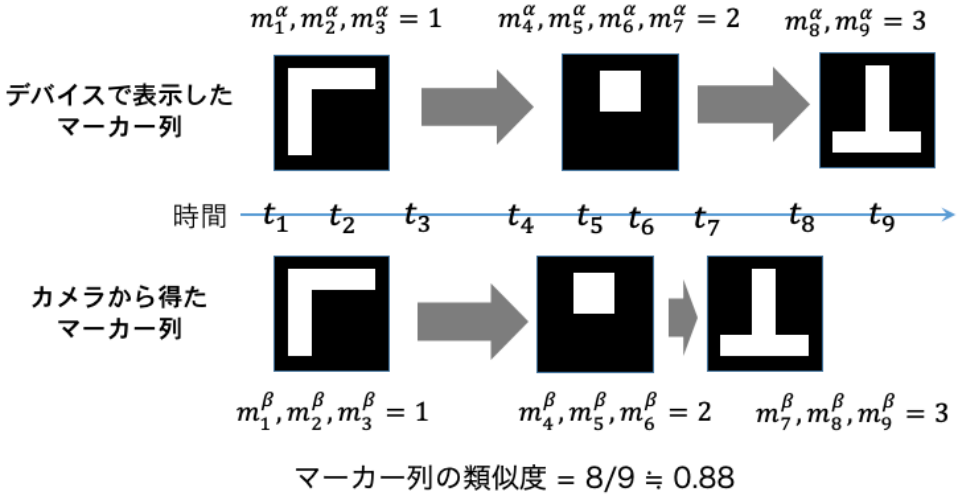

図 4 マーカー列の類似度計算の例

類の種類が 1 つ増加するからである．よって，それ以降のマーカーの種類がすべて異なる種類とみなされる．式 (12) の計算方法であれば誤って認識したマーカー列の部分のみマーカーの類似度に影響する．

3.4 速度データの類似度計算

3.2 節のペアリング手順 (11) において，x, y 軸の速度データの類似度を以下の 4 つの類似度算出方法で算出する．

- シンプルなマッチング
  各時刻におけるデータの値の差の合計を平均化する類似度算出方法である．類似度 s_w は以下のように計算される．

$$s_w = \frac{1}{m}\sum_{k=1}^{m}|\tilde{v}_k^w - \tilde{w}_k| \tag{13}$$

  ただし，$w \in \{x, y\}$ である．s_w の値が低ければ類似度が高いと判断される．
- DP マッチング
  類似度 $g(m, m)$ は以下の漸化式で計算される．$g(m, m)$ はシンプルなマッチングと同様に値が低ければ類似度が高いと判断される．

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

  ただし，コスト関数 c は $c(i,j) = |\tilde{v}_i^w - \tilde{w}_j|/m(w \in \{x,y\})$ とし，$g(0,0) = d(\tilde{v}_1^w, \tilde{w}_1) = c(0,0)$ となる．
- Jaccard 係数
  $\tilde{v}_i^w, \tilde{w}_i (i \in \{1,...,m\}, w \in \{x,y\})$ を $\tilde{v}_i'^w, \tilde{w}_i' (i \in 1,...,m)$ へ 0.05 間隔で量子化し，その後に集合 $\tilde{A} = \{(\tilde{v}_i'^w, i)|i \in 1,...,m\}, \tilde{B} = \{(\tilde{w}_i', i)|i \in 1,...,m\}$ を求める．そのとき，類似度として集合 $\tilde{A}$ と $\tilde{B}$ の差を以下の式で計算する．

$$similarity_w = \frac{|\tilde{A} \cap \tilde{B}|}{|\tilde{A} \cup \tilde{B}|} \tag{14}$$

  この $similarity_w$ は 0〜1 の値をとる．
- 相関係数
  時系列データをベクトルとみなし，その相関を類似度として計算する．類似度 r_w は以下の式で計算される．

$$r_w = \frac{\sum_{i=1}^{m}(\tilde{v}_i^w - \tilde{v}^w)(\tilde{w}_i - \tilde{w})}{\sqrt{(\sum_{i=1}^{m}(\tilde{v}_i^w - \tilde{v}^w)^2)(\sum_{i=1}^{m}(\tilde{w}_i - \tilde{w})^2)}} \tag{15}$$

  ただし，$w \in \{x, y\}, \tilde{v}^w = \sum_{k=1}^{m}\tilde{v}_k^w,$，$\tilde{w} = \sum_{k=1}^{m}\tilde{w}_k$ である．この r_w は-1〜1 の値を取り，正の値であれば正の相関，負の値であれば負の相関を取る．

3.5 実装

ここまで紹介してきたペアリング手法のプロトタイプの開発を行う．デバイス側のアプリケーションでは Android Studio 上で Java 言語を使い，サーバ側のアプリケーションでは Python 言語を用いた．マーカーの認識には OpenCV のライブラリである Aruco[12] を用いた．

図 5 と**図 6** にデバイスとサーバのアプリケーション画面を示す．デバイスは Nexus 5X，サーバはノート PC の DELL MI11C-6WHBR を用い，カメラは外付けのカメラ GROWFAST ASX001B を使用している．デバイスのアプリケーション上でペアリングボタンを押すと，サーバのアプリケーションとルータを介した Wi-Fi 通信を始め，マーカーの種類が 0.5 秒ごとに変化する．その後にユーザがデバイスを動かし，サーバが加速度データと変位データを受け取った後，マーカー列の類似度と速度データの類似度を算出する．

4. 評価実験

3 章で紹介したペアリング手法について，デバイスの大きさ，カメラとデバイスの距離，デバイスを動かすモー

Vol.2019-CSEC-85 No.7
Vol.2019-IOT-45 No.7
2019/5/23

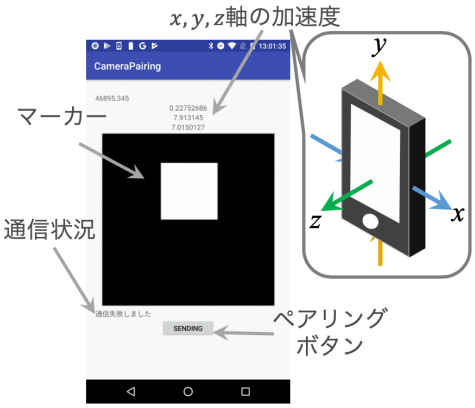

図 5　デバイス側のアプリケーション

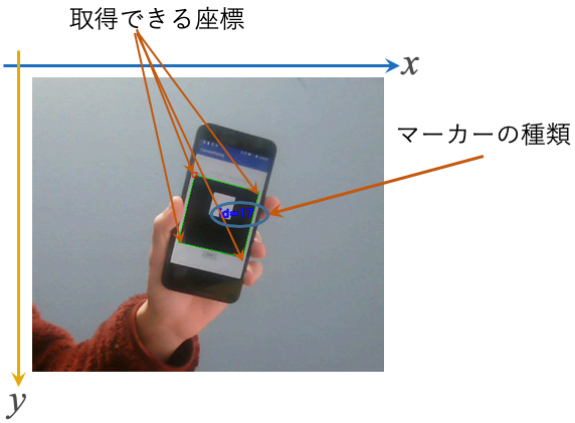

図 6　サーバ側のアプリケーション

ションを変化させた際の，複数の類似度算出方法で類似度を確認する実験を行う．デバイスは，スマートフォンとタブレット (Nexus 5X，Nexus 7) を用い，アプリケーション起動時には画面の大きさに合わせた大きさのマーカーが表示される．カメラとマーカーの距離は，デバイスを動かした際に，最長 3.0m の距離で安定してマーカーが認識できたため，1m 区切りの (1.0m, 2.0m, 3.0m) の距離で実験を行う．モーションは，シンプルなモーション (○) と複雑なモーション (∞) を用いる．

被験者は神奈川工科大学に所属する 20 代の 7 名であり，以下の手順で実験を行う．

(1) 被験者はデバイス (Nexus 5X, Nexus 7) を持ち，カメラとの距離を (1.0m, 2.0m, 3.0m) 離す．
(2) 被験者はデバイスのアプリケーションを起動し，ディスプレイがカメラに映るようにデバイスをカメラに向ける．
(3) 被験者はデバイスでモーション (○，∞) を約 5 秒間描く．
(4) 手順 (1)〜(3) を 5 回繰り返す．

表 1 に上記の手順で実験を行ったときのマーカー列の平均類似度を示す．結果として，マーカーの類似度のついては，Nexus 5X より Nexus 7 を用いた場合の類似度が高くなった．この理由として，Nexus 7 の方が Nexus 5X よりマーカーが大きく表示され，認識されやすかったと考えられる．また，距離による類似度の差は見られなかった．

表 1　マーカー列の平均類似度 (括弧内は標準偏差)

device	Nexus 5X		Nexus 7	
motion	○	∞	○	∞
1.0 m	0.79(0.28)	0.79(0.27)	0.85(0.19)	0.85(0.22)
2.0 m	0.79(0.25)	0.80(0.23)	0.87(0.20)	0.91(0.17)
3.0 m	0.83(0.16)	0.77(0.24)	0.85(0.25)	0.91(0.15)

表 2　速度データの平均類似度 (シンプルなマッチング)

device	Nexus 5X		Nexus 7	
motion	○	∞	○	∞
1.0 m	0.043(0.014)	0.045(0.014)	0.053(0.014)	0.051(0.015)
2.0 m	0.043(0.015)	0.046(0.021)	0.053(0.016)	0.057(0.015)
3.0 m	0.049(0.014)	0.043(0.015)	0.051(0.015)	0.062(0.020)

表 3　速度データの平均類似度 (DP マッチング)

device	Nexus 5X		Nexus 7	
motion	○	∞	○	∞
1.0 m	0.031(0.012)	0.031(0.011)	0.038(0.011)	0.035(0.011)
2.0 m	0.034(0.012)	0.035(0.016)	0.038(0.008)	0.040(0.012)
3.0 m	0.037(0.009)	0.032(0.009)	0.042(0.012)	0.045(0.016)

表 4　速度データの平均類似度 (Jaccard 係数)

device	Nexus 5X		Nexus 7	
motion	○	∞	○	∞
1.0 m	0.24(0.12)	0.26(0.12)	0.19(0.08)	0.22(0.09)
2.0 m	0.26(0.14)	0.27(0.14)	0.20(0.07)	0.19(0.09)
3.0 m	0.24(0.11)	0.27(0.11)	0.19(0.09)	0.20(0.08)

表 5　速度データの平均類似度 (相関係数)

device	Nexus 5X		Nexus 7	
motion	○	∞	○	∞
1.0 m	0.39(0.36)	0.40(0.24)	0.38(0.27)	0.40(0.26)
2.0 m	0.45(0.32)	0.48(0.17)	0.50(0.22)	0.48(0.24)
3.0 m	0.42(0.27)	0.42(0.22)	0.51(0.28)	0.49(0.20)

表 2 にシンプルなマッチングを用いたときの速度データの平均類似度を示す．結果として，マーカーの大きさが小さい Nexus 5X を用いたときの類似度の方が Nexus 7 を用いたときより高いことがわかった．この理由として，Nexus 5X の方が小さくユーザにとって動かしやすかったからであると考えられる．**表 3** に DP マッチングで速度データの類似度を算出した結果を示す．シンプルなマッチングと同様に Nexus 5X を用いた方が Nexus 7 より類似度が高くなり，シンプルなマッチングより類似度が全体的に高くなっていることがわかる．**表 4** に Jaccard 係数で速度データの類似度を算出した結果を示す．これも同様に小さいデバイスである Nexus 5X を用いた方が類似度が高く

なっていることがわかる．**表 5** に相関係数で速度データの類似度を算出した結果を示す．シンプルなマッチング，DP マッチング，Jaccard 係数とは逆に Nexus 7 を用いたときの類似度の方が Nexus 5X を用いたときより高くなった．これは，Nexus 7 の方が Nexus 5X より重量が大きく，ユーザがモーションを描いたときの加速度センサの揺れが少なくなり，高い相関が得られたからだと考えられる．また，Nexus 5X を用いたときは 2.0m，Nexus 7 を用いたときは 3.0m のとき類似度が最も高くなった．この結果から，デバイスとカメラの距離が近いほど類似度が上がることはなく，デバイスの大きさによって適切な距離が存在することが分かった．加えて，上記 4 つの類似度算出方法において，類似度に比べて標準偏差が大きくなり，安定したペアリングができない結果となった．

以上の結果より，デバイスとカメラの距離による類似度の差は相関係数以外では見られず，デバイスの種類については，Nexus 5X などのスマートフォンを用いた際にはシンプルなマッチングや DP マッチングなど，Jaccard 係数など，データの差による類似度算出方法を用いれば類似度が高くなることが分かった．また，Nexus 7 のようなスマートフォンより大きいタブレットを用いた場合は相関係数を用いたデータの相関による類似度算出方法を用いれば類似度が高くなることが分かった．

5. おわりに

本稿では，以前我々が提案した，加速度センサを搭載したモバイルデバイスとカメラを搭載したサーバをペアリングする手法について，外付けカメラを用い，デバイスの種類，デバイスとカメラの距離，モーションを変化させ，複数の類似度算出方法により類似度を確認する実験を行った．

実験の結果として，マーカーの類似度については 0.8〜0.9 ほどの値となり，デバイスとカメラの距離によって類似度が変化することはなかった．速度データの類似度については，スマートフォンなどの小さなデバイスにおいてデータ間の差によって高い類似度が得られ，タブレットなどの大きなデバイスではデータ間の相関によって高い類似度が得られた．今後の課題を以下に述べる．

- 任意のモーションでの実験
  今回の実験では，被験者にシンプルなモーション (○) と複雑なモーション (∞) の 2 つのモーションを描いてもらい，類似度を算出した．しかし，実際にユーザがペアリングを行うときはユーザがその場で任意にモーションを行うため，それを考慮した実験を行う必要がある．
- データの特徴量の導出
  今回の実験では，変位データと加速度データを微分・積分によって速度データに変換しそれらのデータの類似度を直接算出した．しかし，類似度の標準偏差が大きく安定したペアリングが難しい結果となった．そこで，ユーザが描くモーションの特徴を抽出し，それらの類似度を求めることで更に精度よく安定したペアリングができると考えられる．

参考文献

[1] 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 2007), pp. 253-270, 2007.
[2] Yugo Agata, Jihoon Hong, Tomoaki Ohtsuki: “Room-level proximity detection based on RSS of dual-band Wi-Fi signals,” Proceedings of 2016 IEEE international conference on communications (ICC), 2016.
[3] 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.
[4] 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.
[5] 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.
[6] 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.
[7] 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. Camera View,” Intelligent Robots and Systems, pp. 3872-3877 , 2008.
[8] N. Maruhashi, T. Terada, and M. Tsukamoto: “A method for identification of moving objects integrative use of a camera and accelerometers,” Proceedings of the 27th annual ACM symposium on applied computing, pp. 1-6, 2012.
[9] S. Osamu, S. Kagami, and K. Hashimoto: “Identifying a Moving Object with an Accelerometer in a Proceedings of the 16th International Conference on Multimodal Interaction, pp. 216-223, 2014.
[10] M. Nagatomo, K. Aburada, H. Yamaba, N. Okazaki, and M. Park: “Proposal of Ad-hoc Secure Device Pairing Method Using Similarity between Marker Movement and Acceleration,” Proceedings of the Workshops of the 33rd International Conference on Advanced Information Networking and Applications(WAINA2019), pp. 683-692, 2019.
[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(online), 入手先 ⟨https://docs.opencv.org/3.4.0/d5/dae/tutorial_aruco_detection.html⟩ (2019.4.10).