振る動作による携帯機器間のマルチユーザ転送の検討

金岡諒† 尾崎凌介† THEVILOJANAPONG Niwat‡ 狐崎直文† 戸辺義人†

青山学院大学理工学部情報テクノロジー学科† 三重大学工学研究科情報工学専攻‡

近年，スマートフォンの普及，および近距離無線通信の多様化が急速に進んでおり，近距離無線通信を用いたデータの送受信も増加している．近距離無線通信において，目的のデバイスを指定するには MAC アドレス等の ID に依存している．また，一つのサーバから一つのクライアントに情報を転送する仕組みは複数存在するが，一つのサーバが複数のクライアントに向けて情報を転送する仕組みは実装例がない．そこで，ID など機器の識別子に依存せずに，ジェスチャによって通信相手を特定して，携帯電話間で情報をマルチユーザ転送する仕組み EriCC-M を提案する．本稿では，EriCC-M の概要，設計，Android スマートフォン実装について述べる。

1. はじめに

現在，急速にスマートフォンの普及が広まっており，推定では 2015 年には過半数の人がスマートフォンを所持することになると言われている．スマートフォンの OS は様々な会社が制作しているが，その中で最も多いのが，Google 社が開発，提供している AndroidOS である．AndroidOS はオープンソースであり，アプリケーションソフト開発用の AndroidSDK(Software Development Kit)，ランタイムとライブラリ開発用の AndroidNDK(Native Development Kit)と合わせて無償で提供されている．

ビジネスを行う上で必要になることの１つに名刺交換がある．しかし，多くの人と交換しなければならないため，紙の名刺の場合は管理やコスト面でも欠点は多い．そこで，電子名刺が近年用いられるようになった．データ１つで何人でも名刺を渡すことができ，管理も自動で行ってくれる．しかし現状では電子名刺の形式は定まっていないため，電子名刺の受け渡しを行うには特定のアプリケーションやデバイスが必要である．

スマートフォンが普及するにつれて，自分のメールアドレスや電話番号などの個人情報を交換する際，その手段には様々な方法がある．赤外線通信や QR コード読み取りなどがある中，情報交換専用のアプリケーションが一般的になってきている．しかし，そのアプリケーションにも用途によっては使用するのが困難な場面が存在する．

また，近距離無線通信が日常生活のあらゆる場面で使用されるようになってきた．IC カード非接触型無線通信技術や，マウスやヘッドセットなど，主にパソコンや携帯電話の周辺機器に用いられる Bluetooth などが挙げられる．

近年，Bluetooth 等の近距離無線通信機能を備える携帯機器の利用が増えている．近距離無線通信により，近くにいる特定の人との小容量の情報を交換するの が便利になる．しかし，通信相手として機器の識別子を指定することが必要とされ，その識別子である MAC (Media Access Control)アドレスを，無線通信範囲内にいるデバイスの中から目的のデバイスを探し出さなければならない．ほんの「目の前」にいる人に対して，ほか に必要とされる情報が出てくる．

それに対して，NFC (Near-Field Communication)技術が注目されている．NFC は無線通信というより，スマートカードの延長技術として，接触もしくは接触に近い距離で情報を伝達することが基本としており，容易に通信ペアを特定することができる．しかし，逆に接触がないが近隣にある範囲のデバイスに対して情報を伝えることができなくなる．

そこで我々は，複数のユーザ間でジェスチャによって通信相手を特定して携帯電話間で情報交換する仕組み，EriCC-M を提案する．EriCC-M では，振る動作の特徴をとらえて通信相手を決定することを基本とする．

本稿では，以下，2 章で EriCC-M の関連研究について述べる．3 章では，EriCC-M を実装するにあたっての設計について述べる．4 章では，EriCC-M の実装について述べる．5 章では，EriCC-M の評価実験の内容，結果について述べる．6 章では，5 章で行った評価実験を踏まえての考察を

述べる．7 章では，本稿のまとめを述べる．

2. 関連研究

本章では，EriCC-M の関連研究について述べる．

Smart-Its Friends[1] は，Holmquist らによって提案された．2 つのセンサノードを持って，人が振ることにより発生する，加速度信号時系列パターンの類似する 2 つのセンサノードをグループ化するシステムである．中核ユニットからセンサユニットを分け，2 つのボードとのモジュール設計に基づいている．デバイス制御，特有のアプリケーションの処理および他の Smart-Its とのコミュニケーションは中核ユニットで割り付けられている．Smart-Its Friends に組み込まれた動きは，ユーザに Smart-Its での交友関係を課するための非常に使いやすいインタフェースを提供している．ユーザは 2 つのデバイスを手に取り，簡単に振ったり揺らしたりすることで，2 つのデバイスで接続を可能にしている．

Touch-and-Connect[2] は，名古屋大学の岩崎陽平氏らによって提案された無線ネットワーク環境においての機器間接続用フレームワークである．接続元と接続先の 2 種類のボタンを設置し，接続したい両機器のボタンを押すことで機器間の無線接続を可能にするフレームワークである．ユーザが直接機器のボタンを押すことにより機器間接続を行うため，IP アドレスや機器名の指定のようなユーザにとって分かりにくい作業をすることなく機器の連携を可能にした．

直観的インタフェースを提供する点ではこれらは本稿に関連するが，本稿では不特定多数ノードの中から通信相手ノードを特定することを対象にしている点が異なる．

u-Photo[3] は，慶應義塾大学の鈴木源太氏らによって提案された，写真を用いることで機器操作を可能にするシステムである．u-Photo では，ARToolKit[4] を利用し，各聴きに取り付けられているビジュアルマーカを認識することで機器の特定を行っている．機器の連携は，u-Photo のディスプレイに GUI を表示し，ユーザがコマンドをタッチすることで行っている．

Snappy[5] は，慶應義塾大学の伊藤友隆氏らによって提案された，振る動作を用いた聴き連携手法である．各機器に固有の SR(Swing Reference)コードを対応づけておき，携帯機器を振り SR コードの矢印を再現することで機器の特定を行っている．機器特定後，聴き情報を取得する場合は携帯端末を手前に振り，他の機器と連携させる場合は携帯端末を押し出すように振る．

Voice Twitter System Based on Gesture Operation of Smart Phone[6] は，慶應義塾大学の上野大樹氏，安村通晃氏らによって提案された，スマートフォンのジェスチャ操作と音声入出力を利用することにより，リアルタイム性やコミュニケーション性を重視しつつ，手軽で疲れないような新しいマイクロブログのシステムである．

BUMP[7] は，2 つの端末に一定範囲内で同時に加速度を与えることで，無線接続を可能にするアプリケーションである．GPS 機能によって位置情報を獲得し，一定範囲内で加速度センサによる振動を確認すると，同時に振動が与えられた端末同士の通信を確立するシステムである．これは，振る動作を与えることによって通信を確立する点で本稿に関連するが，本稿とは GPS 機能や回線を使用せずに通信を確立するという点で異なる．

3. EriCC-M の設計

本章では，EriCC-M システムの全体の設計の流れを述べる．

EriCC-M は，電子名刺などの小容量データを複数ユーザに受け渡すことを想定し，データを送信する側の Provider と受信する側の Receiver で構成する．Provider か Receiver かはアプリ起動後に選択する．また，EriCC-M は近距離無線通信として Bluetooth を用いる．通信する際，ソケット通信を採用するのだが，Provider 側にはサーバソケットを作成しない．なぜなら，ソケット通信は 1 対 1 の通信のみ可能で，1 つのサーバソケットでは複数の Receiver と通信することができないからである．そこで，各 Receiver にサーバソケットを作成し，Provider が順次 Receiver に接続していくという手法を考えた．Provider がどのように Receiver と通信を行っていくか図 3.1 に示し，その手順を下に記す．

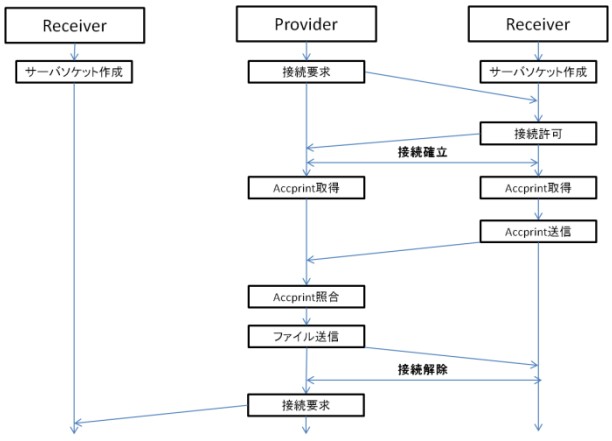

図 3.1 EriCC-M の通信手順

手順 1： Receiver がサーバソケットを作成する．
手順 2： Provider が目的の Receiver に接続要求を出す．
手順 3： Receiver が Provider に接続許可を出し，接続を確立する．
手順 4：Provider，Receiver ともに Accprint を計算する．Accprint の計算法については後述する．
手順 5：Receiver が計算した Accprint を Provide に送る．
手順 6： Provider が Receiver から送られてきた Accprint

と自分の Accprint との照合を行う. 照合方法については後述する.

手順 7： 照合に成功した場合，Provider は Receiver にファイルを送信する.

手順 8： Receiver がファイルを受信すると接続を解除して，Provider は別の目的の Receiver に接続要求を出す.

以下, 手順 3 から手順 8 までを目的の端末台数分繰り返す.

また EriCC-M では Accprint を取得する際，Android 端末に搭載されている加速度センサを用いる.

EriCC-M では取得した加速度を認証に用いるために, ジェスチャ中に取得した 3 軸の加速度信号，$a_x(t)$，$a_y(t)$，$a_z(t)$から，$\sqrt{a_x^2+a_y^2+a_z^2}$の時系列を求める．これをF[n]とし，離散フーリエ変換を適用し，得られた数値F[k]の中で 1 番目と 2 番目に強い成分を持つ周波数を Accprint(f1， f2)として扱う.

Provider は，ソケット通信上では Slave の役割を担う.

Provider は送られてきた Accprint と自分の Accprint の照合を行い，その結果によってその後の処理を変える．照合は Provider の Accprint を（f_1^A，f_2^A），Receiver の Accprint を（f_1^B，f_2^B）としたとき, $|f_1^A - f_1^B|$，$|f_2^A - f_2^B|$が同じジェスチャによって得られた Accprint であると判断する誤差 Δf に対して,

$$|f_1^A - f_1^B| < \Delta f，\ |f_2^A - f_2^B| < \Delta f$$

を満たしたとき，Accprint の照合に成功したと判断する.

Accprint を照合した結果，自分の Accprint と一致しないと判断したとき, Provider は認証失敗とし通信を終了する．認証に成功した場合は，ファイルを送信し通信を終了する．その後，その他の目的の端末への接続を試みる.

Receiver は Accprint を取得した後, Provider に Accprint を送信する．Provider が照合し，認証に成功した場合は Provider からファイルが送られてきて通信が終了する.

4. EriCC-M の実装

EriCC-M での Bluetooth 通信の実装には SPP(Serial Port Profile)を用いる．Android に搭載されている Bluetooth にはピコネット通信を行うためのプロファイルがないため，ソケット通信である SPP を用いる.

Provider は，EriCC-M の Bluetooth 通信において Slave の役割を果たす．まず BluetoothDevice で UUID を呼び出し，BluetoothSocket を取得，初期化する．この時，呼び出した UUID は Server と Client が接続する上でのキーとして用いられる．Master である Receiver の UUID と一致していなければならない.

初期化が終了すると，BluetoothSocket.connect()メソッドを呼び出すことで Receiver との接続を試みる.

Receiver は，EriCC-M の Bluetooth 通信において Master の役割を果たす．まず BluetoothServerSocket をオープンする必要がある．オープンし続けることにより，クライアントの接続要求を受けることと，接続済みの BluetoothSocket を提供することができる.

BluetoothServerSocket のオープンは，listenUsingRfcomm-WithServiceRecord(String, UUID)を呼び出すことで実装できる．String はサービスの識別に用いられるもので，String と UUID はともに SDP(Service Discovery Protocol)に組み込まれる.

BluetoothServerSocker のオープンが終了すると，accept()を呼び出して Slave である Provider からの接続要求の受付を開始する．接続は，要求してきた Provider の UUID が Receiver の UUID と一致した場合のみ受け付けられ，その場合のみ accept()が接続済みの BluetoothSocket を返す.

接続が完了したら，他の接続要求を受け付けないように BluetoothServerSocket.close()を呼び出し，BluetoothServer-Socket を閉じる.

Bluetooth 通信の実装により，Provider と Receiver の接続が完了する．それによって Provider，Receiver ともに BluetoothSocket を持ち，これらを用いてデータの送受信を行うことが可能となる．まず，socket.getInputStream()，socket.getOutputStream()にて, ソケットを用いて転送をハンドリングする InputStream と OutputStream を取得する.

データの読み書きは read()と write()メソッドで行う．EriCC-M では，Receiver が送る Accprint と Provider が送るテキストデータはともに文字列として送信する.

Provider は，Receiver から送られてきた Accprint が自分の Accprint と⊿f の誤差以内に収まっているかの照合を行う.

5. 評価

EriCC-M システムの評価を行った結果を述べる.

まず，定量的評価として，Accprint の照合の際，(f1， f2)のずれを⊿f によって照合判断を行うため，⊿f の閾値を計測し，EriCC-M システムの性能を示す．また，Accprint 取得の際，使用する端末によって差が現れてしまう可能性も考慮し，複数の端末を用いて Accprint を取得してその端末差を求め，EriCC-M システムの性能を示す.

EriCC-M での通信相手を特定するために，Accprint の照合を行う. その際, (f1， f2)のずれを⊿f の閾値で照合成功，照合失敗の判断をするため, ⊿f を最適化する必要がある.

実験では, 平均 11.73Hz で 1 つの⊿f に対して 100 回の試行を行い, 正当な Receiver を Receiver とみなす確率を α (⊿f)，不当な Receiver を Receiver とみなす確率を β (⊿f)として，α と β の確率を算出した．実験環境を表 5.1 に示す.

表 5.1 実験環境

使用デバイス	Galaxy S
OS	Android 2.1
開発環境	JDK 1.7.0

この実験では，2 台のスマートフォンを必要とするため，1 台を Galaxy S(Provider)とし，Galaxy S の Accprint を(f_1^A, f_2^A)とする．もう一台を Galaxy SII (Receiver) とし，Galaxy SII の Accprint を(f_1^B, f_2^B)とする．プラットフォームとして Android を使用した．

Provider に対して正当な Receiver と不当な Receiver の加速度に，$\sqrt{a_x^2 + a_y^2 + a_z^2}$の計算を施し，それぞれ 100 個のデータを取得した．そのうちの 1 つずつを図 5.1， 図 5.2 に示す． 図 5.1，図 5.2 には重力加速度が含まれている．本来であれば取り除くべきであるが，今回は x，y，z のいずれに現れるか判断できないことと周波数軸上の値に変換するときに取り除くことができるので重力加速度を含めたままとした．

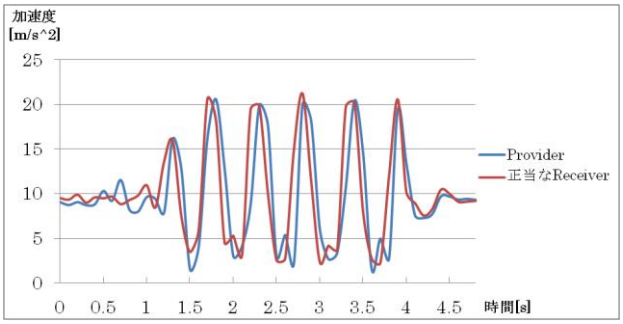

図 5.1 取得した加速度(正当な Receiver)

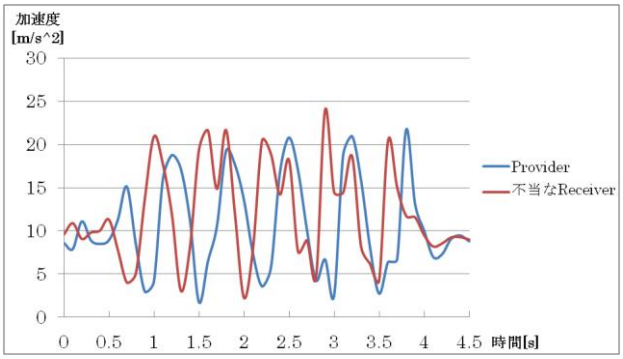

図 5.2 取得した加速度(不当な Receiver)

取得した加速度データに対して離散フーリエ変換を行った．図 5.1 の加速度信号のデータを離散フーリエ変換したグラフを図 5.2， 図 5.3 に示す．周波数が 0 の時は重力加速度により非常に大きな値となるので除いている．

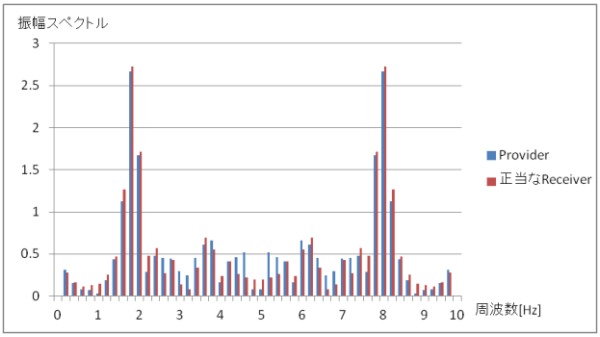

図 5.3 正当な Receiver

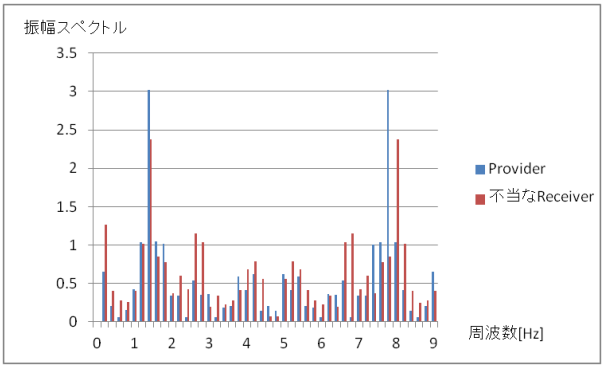

図 5.4 不当な Receiver

⊿f の計測では， それぞれの Accprint のずれの差を算出し， その差が⊿f の閾値以内に収まっていれば， 照合成功とし，反対に⊿f の閾値以内に収まっていなければ照合失敗とする．この時， 正当な Receiver を Receiver とみなすジェスチャ動作と不当な Receiver を Receiver とみなすジェスチャ動作の 2 つに分けて計測を行った．

正当な Receiver を Receiver とみなすジェスチャ動作とは，連絡先を送信する人と受信する人がお互いにスマートフォンを手に持ち，スマートフォンを握った手のまま握手をする．その時の手の振る動作のことを言う．また， 不当な Receiver を Receiver とみなすジェスチャ動作とは，正当な Receiver を Receiver とみなすジェスチャ動作をしている人のジェスチャ動作を真似した，手の振る動作のことを指す．

まず， 平均 11.73Hz で正当な Receiver を Receiver とみなすジェスチャ動作を 100 回行い， その動作で得られた Accprint の平均値が， Galaxy S は(2.3, 1.8)で，Galaxy SII も同様に(2.3, 1.8)となった．Galaxy S と Galaxy SII の Accprint の平均値を 表 5.2 に示す．

表 5.2 正当な Receiver を Receiver とみなすジェスチャ動作の Accprint の平均値

端末	GalaxyS	Galaxy SII
(f_1,f_2)の平均値	(2.3, 1.8)	(2.3, 1.8)

同じように，平均 11.73Hz で不当な Receiver を Receiver とみなすジェスチャ動作も 100 回行い，その動作で得られた Accprint の平均値が，Galaxy S は(2.2, 1.7)で，Galaxy SII は(3.0, 2.0)となった．Galaxy S と Galaxy SII の Accprint の平均値を 表 5.3 に示す．

表 5.3 不当な Receiver を Receiver とみなすジェスチャ動作の Accprint の平均値

端末	Galaxy S	Galaxy SII
(f_1,f_2)の平均値	(2.2, 1.7)	(3.0, 2.0)

⊿f の最適化をする必要があるため，⊿f の差が| 0.2|ごとに増加していく［−0.1 ≤ 0.1］～［−1.0 ≤ 1.0］までを範囲とした．

このとき，⊿f の差が| 0.2| ごとに増加していく［−0.1 ≤ 0.1］～［−1.0 ≤ 1.0］までの範囲内でどの範囲が照合判断に適しているかを評価するために，平均 11.73H z で 1 つの⊿f に対して 100 回の試行を行った．

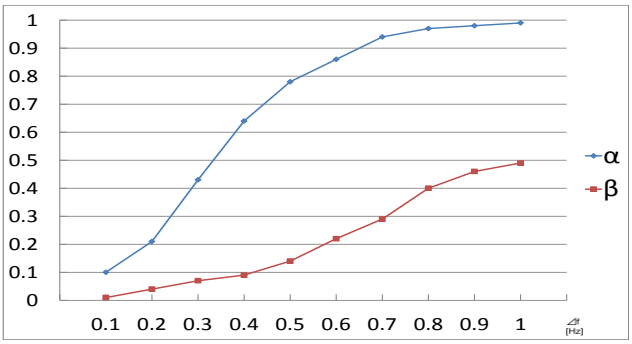

図 5.5 α と β の照合成功確率のグラフ

図 5.5 では，平均 11.73Hz で 1 つの⊿f に対して 100 回の試行を行い，α と β の照合確率を $|f_1^A - f_1^B| < \Delta f$，$|f_2^A - f_2^B| < \Delta f$ で照合した結果をグラフにした．X 軸の単位は ⊿f [Hz] であり，Y 軸の単位は照合確率である．

次に，EriCC-M に用いる端末に搭載されている加速度センサの性能について計測を行った．EriCC-M では，複数の端末を使用するので，様々な種類の端末が使用されることになる．それぞれの端末の加速度を測定し，認証の合否の決定をするので，端末ごとの加速度センサに誤差があると認証の合否に支障をきたすことがある．そこで，3 種類の異なる Android 端末を用いて，加速度センサ固有の誤差について調査した．

実験では，3 台の異なるスマートフォンを使用する．それぞれ，Galaxy S，Galaxy SII，Nexus One である．

3 台のスマートフォンを机の上に置いて静止させ，各端末に搭載されている加速度センサが感知する重力加速度を測定し，各端末の加速度センサの誤差について調査する．重力加速度は 20 回/sec の頻度で 100 回測定した．

各端末の重力加速度は以下の図 5.6 のようになった．

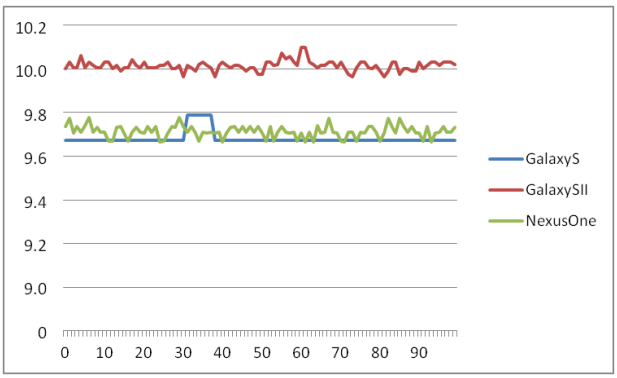

図 5.6 各端末の重力加速度の差

最後に，端末同士が Bluetooth で接続し，加速度を取得し，ファイルを送信し終えるまでの時間を計測した．EriCC-M では複数台の端末と通信を試みるので，1 台あたりにかかる時間の考慮は重要である．

この実験では，2 台の端末の接続開始時から接続完了まで，また，Accprint 取得終了からファイル送信完了までの 2 か所の時間を計測する．また，送信するファイルは同じもので，処理にかかる時間の誤差は考慮しない．ソケット通信を行う際も，以前接続した情報が残らないように端末の電源を毎回切って実験を行った．結果は，以下の図 5.7，図 5.8 のようになった．

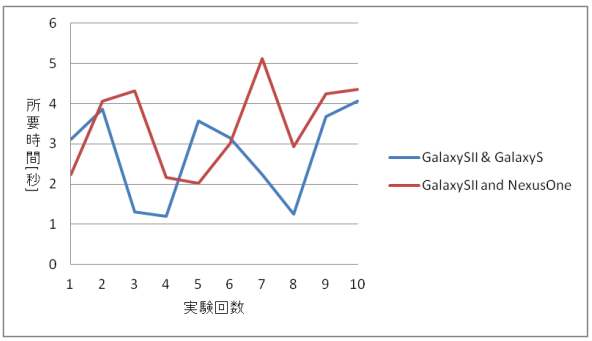

図 5.7 Bluetooth ソケット通信接続の所要時間

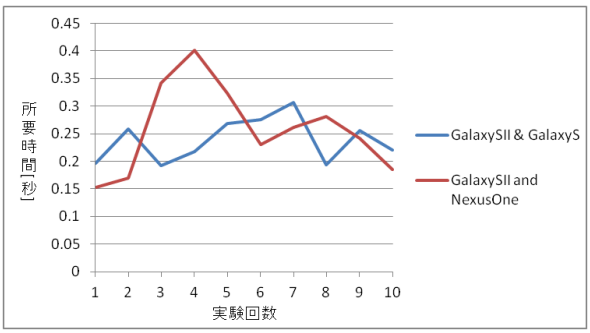

図 5.8 Accprint 取得終了からファイル送信までの所要時間

6. 考察

最初に，Accprint の計測と⊿f の最適化を行った．まず Accprint の計測では表 5.1.2 と表 5.1.3 を見ると， どちらの Accprint の平均値も（2.0～3.0，1.5～2.0）の間で，近似した値となっているため，正当な Receiver を Receiver とみなすジェスチャ動作と不当な Receiver を Receiver とみなすジェスチャ動作では，ジェスチャに特徴的な大きな振りをしない限り，動作に大きな違いがないことがわかる．また， Receiver と Provider の Accprint の値の差が小さいことから，一つの⊿f に対しての閾値の範囲を小さくすることで，より正確な照合判断が可能になると考えられる．したがって， 一つの⊿f に対しての閾値の範囲を|0.2|ごとに増加していく［−0.1 ≤ 0.1］～［−1.0 ≤ 1.0]とした．次に⊿f の最適化では， 図 5.1.5 の α と β を比較してみると，α は β よりも照合成功確率が高くなっていることがわかる．また， ⊿f の値を小さくすると， α と β の照合確率が小さくなり， 反対に⊿f の値を大きくすると， α と β の照合成功率が大きくなるが， 本来照合してはいけない β の照合成功確率も大きくなるため，⊿f の閾値の範囲は， α が 0.9 以上で， ベータが 0.3 以下である[-0.7 ≤ 0.7]の範囲にした．

次に，端末に搭載されている加速度センサ固有の誤差について調査した．計測地点での重力加速度はおよそ $9.80\mathrm{m/s^2}$だが，その値を正確かつ連続して取得することができた端末は 1 台も存在しなかった．また，端末が静止しているにも関わらず重力加速度のグラフは不安定な状態になっているものも見受けられた．実験を行った 3 台の端末を比較してみても，一致したものは 1 組も存在しなかった．この実験は，緯度，標高の同じ机の上で場所を変更せずに行ったので，重力加速度自体の誤差については考慮しない．このことより，加速度センサ固有の誤差によって，正当な Receiver を不当な Receiver に，また不当な Receiver を Receiver と誤認識してしまう可能性もあるということが考えられる．

最後に，端末同士の通信からファイル送信までにかかる時間について調査した．図 5.8 のグラフからわかるように，ソケット通信を確立するまでにかかった時間にはばらつきが出た．Galaxy SII と Galaxy S の組み合わせの場合，最大で約 2.98 秒の差が現れ，Galaxy SII と Nexus One の組み合わせでは最大約 3.19 秒の差が現れた．図 5.7 から，これは異なる端末によって生じる誤差ではないということが考えられる．このことから，ソケット通信が確立して Accprint を取得するまでに，多くの時間を費やしてしまう可能性があると考えられる．

図 5.8 は，Accprint を取得し終えて，Receiver が Provider に Accprint を送信し，Provider が照合して Receiver にファイルを送信する，という操作にかかった時間のグラフである．この図から見て分かるように，この操作はソケット通信の確立にかかった時間に比べて時間のばらつきが非常に少ない．端末による差も少なく，ファイルのサイズが同じであれば通信にかかる時間はあまり変わらないと言える．ただし，ファイルサイズ，文字数が多くなると処理に時間がかかってしまうと考えられる．

7. まとめ

本稿では，ジェスチャによって相手を特定し，マルチユーザ間でファイル送信を行う，EriCC-M について述べた．EriCC-M は，加速度センサと Bluetooth の組み合わせにより，2 台のスマートフォンを同時に同じ動作を施し，それを繰り返すことで，複数のユーザにファイルを送ることを可能にした．これは，今後さらに普及していくスマートフォンを利用する上で，見知らぬ人への個人情報流出をなくし，特定の相手とだけ通信することができる．

また，端末の加速度センサによって誤差が生じてしまう問題については，EriCC-M を使用するときに，静止している状態で重力加速度を取得し，その平均値を誤差としてあらかじめ考慮するなど，誤差を前もって知るような操作を施さなければならない．

参考文献

1. Lars Erik Holmquist， Friedemann Mattern， Bernt Schiele，Petteri Alahuhta，Michael Beigl and Hans-W. Gellersen． Smart Its Friends，Ubicomp，Vol. 2201，pp.116-122，(2001)
2. 岩崎陽平，河口信夫，稲垣康善.『Touch-and-Connect:ユビキタス環境における接続指示フレームワーク』,情報処理学会論文誌，Vol. 45，pp.2642-2645，(2004)
3. 鈴木源太，岩本健嗣， 高汐一紀， 徳田英幸.『u-Photo:ユビキタス情報を付加した画像を実現する環境情報スナップショットの開発』，情報処理学会研究報告，Vol. 63，pp.56-72，(2004)
4. ARToolkit, http://www.hitl.washington.edu/artoolkit/
5. 伊藤友隆， 河田恭兵， 中川直樹， 生天目直哉， 橋爪克弥，伊藤昌毅，中澤仁，高汐一紀，徳田英幸.『Snappy:振る動作による異種ネットワーク間での機器連携の実現』,情報処理学会研究報告.UBI，Vol. 17，pp.31-36，(2009)
6. Ballagas ， R. Rohs， M. Sheridan， J. Borchers，J: The Smart Phone: A Ubiquitous Input Device, IEEE Pervasive Computing，Vol. 5 (1) ，pp.70-77，(2006)
7. Bump - 電話間でまたはコンピュータに写真，ファイル，連絡先情報を簡単に転送，https://bu.mp/