

# Toss-It: モバイルデバイスにおける「トス」や「振り」の動作を用いた情報の移動を直感的に実現するインタフェース技術

矢谷 浩司<sup>†</sup> 田村 晃一<sup>†</sup> 廣木 桂一<sup>†</sup> 杉本 雅則<sup>†</sup> 橋爪 宏達<sup>‡</sup>

<sup>†</sup> 東京大学 <sup>‡</sup> 国立情報学研究所

携帯電話やPDAに代表されるような、モバイルデバイスが日常の生活に広く普及してきているが、そのインタフェースに関していくつかの問題を抱えている。特に、モバイルデバイスにおける他の機器との情報の移動に関しては、ユーザに煩雑な操作を強いる場合が多い。本研究では、モバイルデバイスにおける情報の移動を直感的に実現する、Toss-It というインタフェースを提案する。Toss-It によって、ユーザはボールをトスするように、あるいはカードを配るようにデジタルな情報の移動を実現できる。

## Toss-It: Intuitive Information Transfer Techniques for Mobile Devices with “Toss” and “Swing” Actions

Koji YATANI<sup>†</sup> Koiti TAMURA<sup>†</sup> Keiichi HIROKI<sup>†</sup>  
Masanori SUGIMOTO<sup>†</sup> and Hiromichi HASHIZUME<sup>‡</sup>

<sup>†</sup> University of Tokyo <sup>‡</sup> National Institute of Informatics

Recently, mobile devices have rapidly penetrated into our daily lives. However, several drawbacks of mobile devices have been mentioned so far. Toss-It provides intuitive information transfer techniques for mobile devices, by fully utilizing their mobility. By using the developed system called Toss-It, a user can send information from the user’s PDA to other electronic devices with a “toss” or “swing” action, as the user would toss a ball or deal cards to others.

### 1 背景

近年、携帯電話やPDA(Personal Digital Assistant)に代表されるような、モバイルデバイスが日常の生活に広く普及してきている。また、さらに今後もモバイルデバイスが、さまざまな場面において活用されていくことが大きく期待されている。モバイルデバイスは持ち運びが容易などいくつかの利点があるが、モバイル性を持つがゆえにそのインタフェースに関していくつかの問題を抱えている。例えば、情報を表示するディスプレイが限られていることや、文字などの入力手法が依然成熟していないことなどである。

モバイルデバイスにおいては、他のモバイルデバイスやコンピュータ機器(パーソナルコンピュータ

やプリンタ等)との間における情報の移動にも、解決すべき課題がある。例として、自分のモバイルデバイスから相手のモバイルデバイスへと、ファイルをコピーする場合を考える。これは赤外線通信やメモリカードなどを使って実現することができるが、これらの方法は人間にとってあまり直感的でない操作であったり、作業を完了するまでに多くの手順を必要としたりする場合がある。あるいは、複数のデバイスへとデータのコピーを行う際には、各々のデバイスに対して上述のような操作を何度も行わなければならない。

一方、現実世界において我々が近くの人にもものを渡す際には、相手に向かってものを投げ渡す(トスをする)という動作だけで済ませてしまうことがで

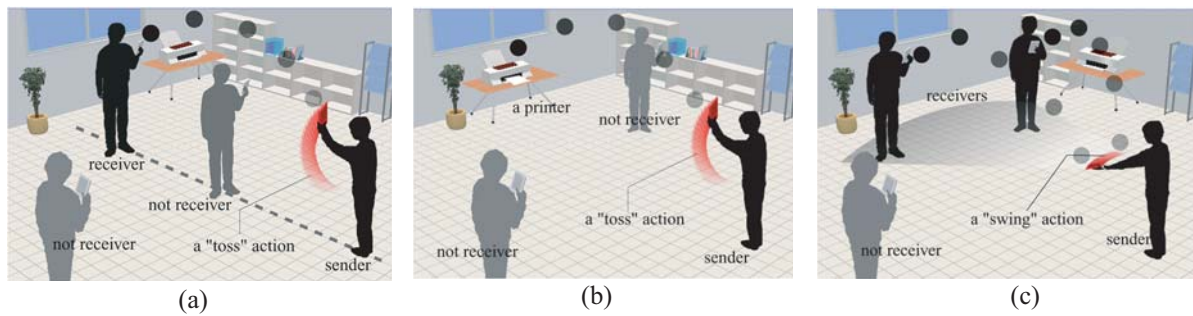


図 1. Toss-It を用いた直感的な情報の移動例: (a)PDA から PDA へ (b)PDA からプリンタへ (c)PDA から複数の PDA へ

きる。この現実世界での動作と同じように、デジタルな情報の移動を実現することができれば、前述のような煩わしい操作を行う必要性がなくなる。そこで本研究では、モバイルデバイス(本研究ではPDAを用いる)における情報の移動を直感的に実現する、Toss-It というインタフェースを提案する。Toss-It によって、図 1 に示すように、「トス」(縦に振る動作)で単数のユーザやデバイスに情報の移動が行える他、「振る」(横に振る動作)で複数のユーザやデバイスに対して一斉に情報の移動を行うことができる。

このように「トス」や「振る」という動作によって情報の移動を行うためには、Toss-It は大きく分けて以下の 3 つの要件を満たさなければならない。

- 「トス」もしくは「振る」動作を認識する
- 複数のユーザの位置と向きを認識する
- ファイル転送等のデータ通信ができる

Toss-It では、「トス」や「振る」動作を取得するために、PDA に加速度センサと角速度センサを取り付ける手法を採用した。動作の推定を行うその他の手法としては、カメラを用いた画像認識が挙げられる。しかし、カメラを用いる手法では、任意の場所、および向きで行われるユーザの動作を取得するには、多くのカメラを必要とする。一方、本手法はカメラを用いる手法と比較して、比較的低いコストで実現できる、外部の機器や複雑な設定を必要としない、といった利点がある。

ユーザの位置と向きとの認識に関しては、モバイルデバイスの持つ携帯性を損なうことのないものがよい。さらに、Toss-It においては各ユーザや各デバイスの相対的な距離と向きが推定できればよい。我々は現在、特殊な外部装置を必要としない位置認識技術の開発を進めているが、本稿においては位置認識技術としてステレオカメラを外部に設置し、赤外線

LED のマーカーを用いたもので代用し、Toss-It の実現可能性を示すことを優先した。

また、3 番目の要件に関しては現在一般的にモバイルデバイスでも利用されている無線 LAN と、ソケット通信を用いたプログラムを用いることとした。さらにサーバを準備し各々の通信を管理するようにした。

## 2 関連研究

### 2.1 モバイルデバイスに人間の動作を適用した研究例

Hinckley らの研究 [3] では、センサを用いてデバイスの傾きを感知できるようにすることで、例えば、画面のスクロールを PDA を傾げることで行えるなど、直感的な操作を実現している。この他にも、同様の手法によってモバイルデバイスのユーザインタフェースをより直感的なものにするを目指した研究がある [1, 5]。

Peephole Display[8] は、PDA のディスプレイの位置とバーチャルな空間の位置とを結びつけており、ちょうどのぞき窓から風景をのぞいているような感覚を与えている。これによって、PDA を持って上下・左右方向に動かすことで、ユーザが使用できる画面の大きさを PDA のディスプレイ以上に劇的に大きくすることができる。

また、[9] では、リビングルームにあるような電子機器を統一的に操作できる XWand というデバイスが提案されている。例えば、XWand を照明に向けて振ると照明の on/off ができたり、テレビに向けて XWand を時計方向/反時計方向にねじることでボリュームの大小を調節することができる。

以上のように人間の動作をモバイルデバイスに適用し、より直感的なインタフェースを提供するシステムは提案され続けている。しかしながら、情報の

移動のような複数のモバイルデバイス間でのインタラクションを支援するインタフェースに、ジェスチャを応用している研究は数少ない。

## 2.2 情報の移動を支援する研究例

Rekimoto は、電子的にエンハンスされたスタイラスを使用することにより、ハンドヘルドデバイス上のオブジェクトをそのスタイラスで持ち上げ (pick up)、他のハンドヘルドデバイスやスクリーンなどに移動させる (drop) ことができるシステムを提案している [4]。これによって、非常に直感的なデータのやりとりを行うことができる。同様に、赤外線タグを使用して情報の移動を行う端末を指して特定し、ファイルの移動等を直感的に行うことができるシステムがある [6]。

これらのシステムは、情報の移動を現実世界での行動に近い形で提供している。しかしながら、情報を移動する際には相手の端末まで十分近づいて使わなければならない点や、マルチキャスト機能を実現していない点で異なる。さらに、図 1(a) のように人を越えて情報の移動を実現するようなことは、[6] で用いられている、pointing という動作では十分に実現できない。

## 3 動作の認識

### 3.1 ハードウェア

「トス」や「振る」動作を認識するために開発した基板を、図 2 に示す。この基板では、検出範囲  $\pm 10$ [g]、感度  $4.0$ [%/g] の性能を持つ、2 軸の加速度センサである Analog Devices 社製 ADXL210 と、検出範囲  $300$ [deg/sec]、感度  $0.67$ [mV/(deg/sec)] の性能を持つ、1 軸の角速度センサである村田製作所製 ENC-03J の 2 種類のセンサを用いている。図 2 の基板には、4 つの加速度センサと 3 つの角速度センサが取り付けられており、これによって合計 11 の自由度を持っている。動作を認識する上では 6 自由度あれば十分であるが、より正確に動作を推定したり、複雑な動作を認識することを目指し、試行的に自由度の数を大きくしている。

なお、センサ群の出力はカットオフ周波数  $1$ [kHz] のローパスフィルタを通して、さらに信号を増幅して H8 マイクロプロセッサに送られる。H8 マイクロプロセッサではセンサ群の出力を AD 変換し、シリアル通信を介して PDA 本体へそのデータを送ることができる。

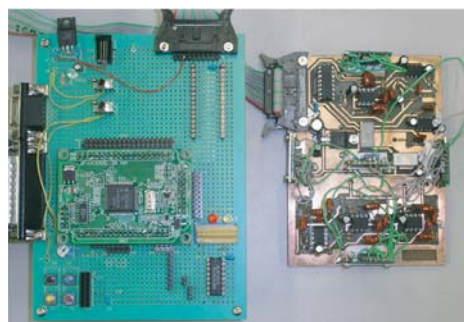


図 2. 慣性センサ群を搭載した基板

### 3.2 認識アルゴリズム

Toss-It においては、単に「トス」や「振る」動作を認識するだけではなく、「トス」する動作においてはその強さを、「振る」動作においてはその軌跡をも考慮する必要がある。これは、「トス」する動作では、図 1(a) のようにその強さによって情報の送信距離を、「振る」動作では、図 1(c) のようにその軌跡によって情報の送信範囲を決定するためである。本節では、「トス」や「振る」動作の抽出、「トス」の強さの推定、さらに「振り」の軌跡の推定の 3 つに関するアルゴリズムを順に説明する。なお、現在利用できる PDA では処理能力に限界があるため、現段階においてはサーバにてこれらの処理を行っている。

#### 3.2.1 「トス」や「振る」動作の抽出アルゴリズム

理想的には、図 2 の基板に取り付けられた慣性センサ、特に加速度センサの出力をそのまま用いることで、「トス」や「振る」動作を取得することが可能である。しかしながら、幾度かの試行実験の結果、「トス」や「振る」動作が終わった直後において、特に加速度センサにおいて無視できない揺らぎが計測されることが確認された。図 3 は「トス」する動作を行った際の加速度センサの典型的な出力をグラフにしたものである。この無視できない揺らぎによって、特に「振り」の軌跡の推定値が大きく違ってしまいうため、この揺らぎを除去する必要がある。

この揺らぎを除去するアルゴリズムを考える上で、アルゴリズムを単純化するための仮定をおいた。すなわち、「トス」や「振る」動作は静止状態 (速度、加速度共に 0) から始まって、静止状態で終了するという仮定である。また、スナップを利かせるような動作に関しては動作の開始から動作の終了までの時間が一般的に非常に短く、十分に検出することが難しいことが観察されたため、現段階では扱わないこととした。この仮定によって、ある一つの動作において速度の増加量と速度の減少量は等しくなると考

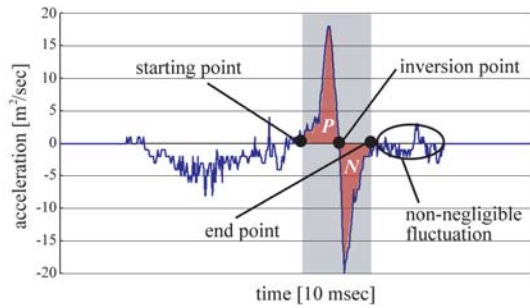


図 3. Toss-It で「トス」の動作を行った際の加速度センサの典型的な出力例

えることができる。すなわち、加速度センサの出力から積分を行った場合に、正の領域における積分量(図 3 の P 領域)と負の領域における積分量(図 3 の N 領域)は等しいことになる。

このことから、不要な揺らぎを除去し、「トス」や「振る」動作における慣性センサの一連の値を取得するアルゴリズムを実装した。このアルゴリズムの説明を図 3 を例に用いて説明する。まず加速度センサの出力値から加速度 0 の軸と交差する時点、すなわち加速度センサの出力値の符号が反転する時点を検出する。この時点と一つ前の符号が反転する時点(この時点を“starting point”と呼ぶ、図 3 参照)との間で、加速度センサの出力値を積分する。この積分値があらかじめ決められた閾値よりも大きく、かつ加速度の最大値が閾値より大きい場合、検出した符号が反転する時点を「トス」や「振る」動作における速度が減少し始める時点とみなす(この時点を“inversion point”と呼ぶ、図 3 参照)。続いて、inversion point から符号が反転した加速度センサの出力値の積分を始め、この負の領域における積分値が、starting point から inversion point までの積分値と、ある閾値以下の差で 0 に近づく時点を求める(この時点を“end point”と呼ぶ、図 3 参照)。end point が求めた時点で積分計算を終了し、starting point から end point までの時間で「トス」や「振る」動作が行われたと認識する。このアルゴリズムによって実際に認識されたものが図 3 の灰色の部分である。

さらに、PDA はユーザによって自由に回転することが可能であるから、「トス」の強さや「振り」の軌跡をより正確に計算するために、基準座標系と PDA 座標系における座標変換を考慮する必要がある。「トス」や「振る」動作は静止状態から開始するという仮定を用いて、starting point における座標変換はあらかじめ簡単な形で決定できるものとしている。ま

た、動作中は角速度センサの出力値から、基準座標系と PDA 座標系の変換行列を逐次計算することができる [10]。

### 3.2.2 「トス」の強さの推定

「トス」する動作において、トスするものが物理的な物体であれば、その飛距離は運動方程式を用いることで計算できる。この時、飛距離は初速度と投げ上げ角度によって決定される。Toss-It では、実際に物理的な物体を投げるわけではないので、「トス」する動作における初速度と投げ上げ角度を慣性センサの出力値から決定する必要がある。

我々は「トス」する動作に関する簡単な実験を何度か行った後、次のような仮定をおくこととした。すなわち、(1)「トス」する動作は PDA を床に垂直下向きにした状態から始まって、フォロースルーなしで終了する、(2)「トス」する際には物体を最高速度の時点で手から放す、である。この仮定によって、初速度を 4.1 節のアルゴリズムで抽出された時間における最高速度とした。一連の動作における最高速度は、加速度センサの値から積分計算を行うことによって得られる。具体的には、PDA 上で観測された 3 軸の加速度の値から、作用する重力加速度を除去する。その後、積分計算を行い、PDA 座標系から基準座標系へと座標変換を行うことで、基準座標系における速度を得ることができる。また、投げ上げ角度は 45[度] で一定とした。

### 3.2.3 「振り」の軌跡の推定

「振る」動作に関しては加速度成分を 2 回積分することによって動作の軌跡を推定している。推定をより簡単に行うために、「振る」動作によって描かれる軌跡は、体を中心とした円弧であると仮定した。これによって軌跡から円弧の中心角を求めることで、受け手の範囲をより容易に特定できることになる。また、「振る」動作においては鉛直方向の動作は大きくないとし、これを無視することとした。

PDA の軌跡ベクトルは加速度を 2 回積分することで求めることができる。この際、PDA 座標系から基準座標系への変換を同時に行うことで、基準座標系における変位ベクトルを求めることが可能となる。この求められた各点を結んでいくことによって、PDA の回転にとらわれない軌跡を得ることができる。上述の円弧の仮定により、動作の開始点と動作の終了点との直線を、円弧の弦と見立てることができる。この弦の長さや円弧の半径から、円弧の中心

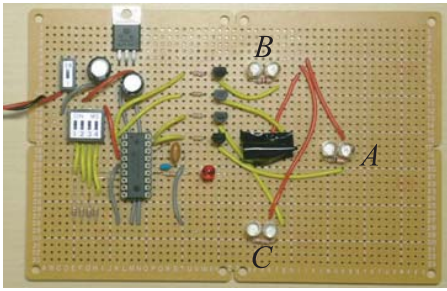


図 4. 赤外線 LED を用いたマーカー

角を計算することができる。なお、円弧の半径はあらかじめ実験をして決定した。

## 4 位置認識

屋内における位置認識技術に関しては、すでに様々なものが提案されている [2]。本研究では、それらの技術の中から、カメラを利用することとした。カメラを利用する位置認識システムは比較的構築が容易である点や、他技術と比べても高い精度の位置認識が行える点などといった利点がある。以下では、我々が開発した位置認識システムについて述べる。

### 4.1 デバイス

図 4 は、位置認識システムで使われるマーカーである。このマーカーでは赤外線 LED が二等辺三角形の形になるように配置されている。後述の認識処理においては、この二等辺三角形の形を認識することで、ユーザの位置と向きを求めることができる。また、頂点 A にあたる赤外線 LED は、デバイス毎に異なる点滅パターンで点滅し続けている。この点滅パターンを認識することで、ユーザの同定を行うことができる。

カメラに関しては、より正確な位置や向きをより容易に取得するため、複眼のカメラ (以下、ステレオカメラという。) を複数台用いている。さらに、ステレオカメラのレンズ部に 900[nm] 以下の光を遮断するフィルタを取り付け、赤外線 LED の光のみを取得するようにしている。

### 4.2 認識処理

マーカーの赤外線 LED をステレオカメラでとらえると、画像中で白色の円形として表示される。この画像から各赤外線 LED の 3 次元位置を決定するために、まず各赤外線 LED の画像中での重心位置を求める。具体的には、閾値を設けて白色の領域を抽出した後、領域毎で重心の座標を求め、これを各赤外線 LED の画像中での重心位置とする。同様にもう一方のカメラ画像からも各赤外線 LED の画像

中での重心位置を取得する。

続いて、得られた 2 つのカメラ画像における各赤外線 LED の重心位置の対応付けを行う。簡単に対応する点を見つけることが可能である。対応する重心位置が決定した後、左右画像での「ずれ」を計算する。この「ずれ」によって奥行き方向の距離を決定することができる。このようにして各赤外線 LED の 3 次元位置を求めることができる [7]。

さらに、求められた 3 つの赤外線 LED の 3 次元位置から、それらの重心を求めることで、ユーザの位置が推定できる。さらに、3 点の中から最も長い辺をなす 2 点を点滅していない点 (すなわち、頂点 B、及び頂点 C) と見なし、 $\vec{BA}$  と  $\vec{CA}$  の和をユーザの向きとして推定する。また、カメラからの画像を複数フレーム取得し続けて点滅パターンを取得し、ユーザの同定を行う。

## 5 評価実験

### 5.1 動作の認識に関する評価実験

動作を認識するために設計したハードウェアとアルゴリズムが、Toss-It におけるシステム要件をどの程度満たすものかを確認するため、評価実験を行った。実験においては、特に「トス」の強さや、「振り」の軌跡がどの程度正確に推定できているかを評価するものとした。具体的には、「トス」する動作に関しては、被験者から 1[m]、2[m]、3[m] 離れた場所に向かって情報を投げ渡す動作を、「振る」動作に関しては、45[度]、90[度]、135[度] の 3 つの角度を目標に動作を行ってもらったこととした。被験者は 6 名で、各々の目標値に対して 25 回動作を行ってもらい、その推定値を記録した。なお、実験前に動作に慣れてもらうために、簡単に Toss-It の説明をし、「トス」や「振る」動作の練習をしてもらった。

図 5 に、飛距離の推定値を目標の飛距離毎に分けて示す。また、図 6 に、角度の推定値を目標の角度毎に分けて示す。各推定値の分布のピークは各目標値付近に存在していることがわかる。各々の目標値に対する推定値の平均と分散を、表 1、表 2 に示す。これらの値から、目標飛距離の 1[m] と 2[m]、2[m] と 3[m] の間、及び目標角度の 45[度] と 90[度]、90[度] と 135[度] の間における平均値の差異に関して Welch の t-検定を行ったところ、その差異は有意であることが確認された (両側検定、 $p < .01$ )。

以上の実験から、上述のハードウェアとアルゴリズムによって「トス」や「振る」動作を取得するだ

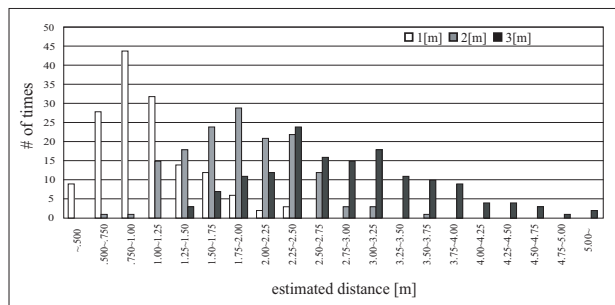


図 5. 推定飛距離の分布

目標飛距離 [m]	平均 [m]	標準偏差
1	1.06	0.418
2	1.90	0.513
3	2.88	0.806

表 1. 各目標飛距離に対する推定値の平均と分散

だけでなく、「トス」の強さや「振り」の軌跡がある程度正確に取得できていると考えられる。

## 5.2 位置認識に関する評価実験

4 節で述べた位置認識技術が、Toss-It において使用できるだけの十分な性能があるかを検証するために、位置認識技術に関しても実験を行った。実験では、カメラから 1.5[m] 離れた水平面上における位置と向きを測定し、その誤差を測定することとした。向きに関しては、床を垂直に見下ろすようにカメラを垂直に設置することを考えれば、床と平行な面上での向きが重要であると考えられる。そこで、カメラの奥行き方向に関しては、傾きを持たないようにして測定した。また、ユーザ同定に関しては 5 種類の点滅パターンを用意し、識別できるかどうかを実験した。

図 7 に、位置の測定における誤差分布を示す。図 7 に示されているとおり、誤差が最も大きくなると思われるカメラ画像の端の部分においても、6[cm] 程度の誤差で収まっており、十分な精度で位置情報を取得できると考えられる。

図 8 は誤差のばらつきが最も大きくなる、カメラ画像の端の点での向きの測定における誤差分布である。しかしながら、その誤差のばらつきは平均して約 10[度] になっている。

ユーザ同定に関しては、2~3[秒] 間点滅パターンを取得すれば正確な識別が可能であるとわかった。また、急激な動作をした場合などには点滅パターンを正確に取得することはできないが、動作後静止すれば点滅パターンを再取得することができ、ユーザ

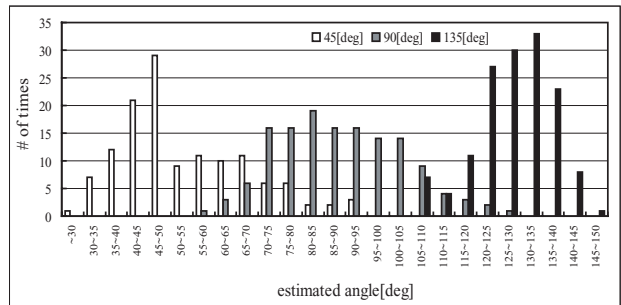


図 6. 推定角度の分布

目標角度 [度]	平均 [度]	標準偏差
45	54.2	14.8
90	89.2	14.3
135	129	8.80

表 2. 各目標角度に対する推定値の平均と分散

を再識別可能であることがわかった。

以上の実験から、本位置認識手法は Toss-It で使用する際には十分な精度を実現できていると考えられる。

## 5.3 ユーザシナリオによる評価実験

実際に Toss-It が使われる場面を想定した評価実験を行った。評価実験では、情報の送り手を 1 人、受け手の候補を 4 人とし、図 9 のように並んでもらうこととした。送り手はある程度自由に動くことができ、向きも自由に変えることができることとした。この状況で、受け手を誰にするかを変え、どの程度正確に情報の移動が実現できるかを検証した。具体的には「トス」による情報の移動に関する実験 (E1、E2、E3、E4)、「振り」による情報の移動に関する実験 (E5、E6、E7、E8) を図 9 のように決めた。

受け手の決定方法は、「トス」の場合は送り手の位置と向き、「トス」の強さから情報の着地点を計算し、その着地点から 2[m] 以内に存在し、かつ着地点から最も近いユーザを受け手とした。また、「振り」の場合は送り手の位置と向き、「振り」の大きさから送信範囲を決定し、その送信範囲内に存在するユーザ全員を受け手とした。さらに、送り手の位置と向きは送るべきファイルを選択した時点の情報を用い、ファイルを選択してから動作し終わるまで、位置と向きは変化しないものとした。

全ての実験を 5 人の被験者で行い、そのうち 4 人 (全員男性、右利き) に各実験において 10 回の試行を行ってもらった。なお、各実験前に Toss-It に慣れってもらうための練習を行ってもらった。

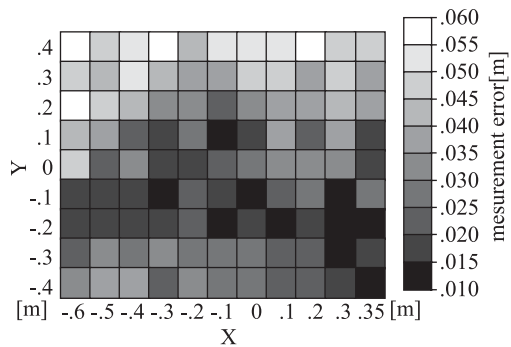


図 7. 位置の測定における誤差分布

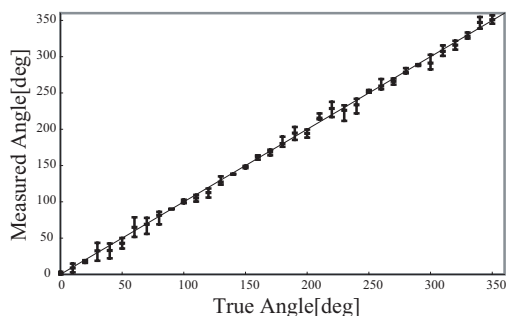


図 8. 向き of 測定における誤差分布 (カメラ画像の端の点)

表 3 に各実験の結果を示す。ここでの成功率とは、指定された相手に対してのみ情報を送ることができた回数の、全ての試行回数に対する割合である。この実験結果を分析したところ、以下のような知見を得た。

- 「トス」の動作の推定誤差に関する影響: E1 における全試行のうちの 10.0[%] は、user2 (user1 の奥に存在するユーザ) に渡っていた。この時、「トス」の推定飛距離が 2.25[m] ~ 5[m] になっており、このことが失敗の主要因であると考えられる。また、E2 では、「トス」の推定飛距離が 5[m] ~ 8[m] という大きすぎる値になって失敗するケースが 2.5[%]、推定飛距離が 1[m] 程度の小さすぎる値になって失敗するケースが 20.0[%] となっており、「トス」の推定に関する誤りが大きく起因していることが分かった。このような誤りは E3 や E4 においても見られた。
- ユーザの向きの認識誤差に関する影響: E1 のにおける全試行のうちの 10.0[%] は、user4 に渡っていた。このケースと成功したケースを比較すると、向きが大きく異なっていた。この中には実際に向くべき方向と、20[度] 以上違っているケースもあった。これは、5.2 節で示したカメラの認識誤差の範囲を大きく超えている。今回の実験で

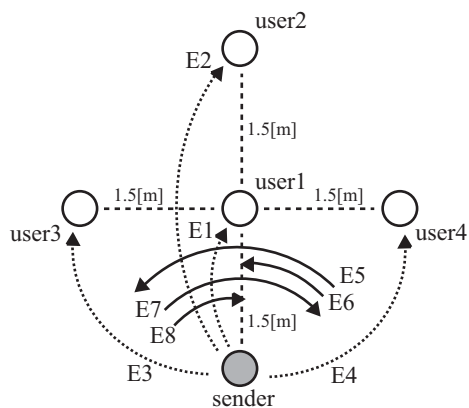


図 9. ユーザスタディにおけるユーザの配置と「トス」および「振り」に関する実験

実験	成功率 [%]	実験	成功率 [%]
E1	75.0	E5	85.0
E2	75.0	E6	60.0
E3	80.0	E7	90.0
E4	70.0	E8	55.0

表 3. 各実験における情報の移動成功率

は、マーカーを PDA に付けており、ユーザの向きとして認識されるのは PDA の向いている方向となり、実際のユーザの向きとは違ってしまう場合がある。このために、ユーザの意図とは違う方向に情報が「トス」されるケースが生じたと考えられる。また、この傾向は「振り」に関する実験においてもいくつか確認された。

- 全員に情報を移動する際の動作の特徴: E5 や E7 のように全員に対して PDA を「振る」場合、実際の角度よりも大きく振ることが多かった。その結果、成功率が他の実験よりも高くなったと考えられる。また、被験者からは、「全員に配るのは大きく振ればよいので、楽でとてもやりやすい。」という意見が得られた。
- 小さく「振る」動作の際の特徴: E6 や E8 のように一部のみにだけ配るような動作は、必要以上に大きく振りがちになり、失敗するケースがあった。被験者の中には振り始めの向きに余裕を持たせる (例えば、E6 であれば、user4 に直接向けてから振り始めるのではなく、ほぼ真横に向けておいてから振り始め、90 度に振って止める) などして、より正確に投げようとしていた。

表 3 で示した結果は、以上のような要因を 1 つ、あるいは複合的に含んでいることによって、失敗するケースが発生しているものと考えられる。

## 5.4 考察

本節では、主に 5.3 節で述べたユーザシナリオによる評価実験の結果をふまえて、より精度よく情報の移動を行うためにどのようなことが必要であるかについて議論を進める。

ユーザシナリオによる評価実験では、情報の移動に失敗したケースの多くは、間違っただけに情報が送ってしまうケースであった。これは、動作の推定アルゴリズムを改良することなどで、改善できる可能性がある。その一方で、送り手の動作だけではなく、受け手側の動作や状態を含めることが、1つの解決策であると考えている。例えば、PDA を斜めに傾けることで、情報を受け取る意図を表現することができれば、受け手をより明確に特定でき、失敗するケースを大きく減らせる可能性がある。

あるいは、送り手だけでなく受け手の向きも考慮することが考えられる。受け手の候補となるユーザが、送り手の方向に向いていない場合は、情報を受け取る意図がないか、その情報の移動において無関係である、と判断できれば、前述の例と同様、失敗するケースを減らすことができると考えられる。

また、今回の実験においては、「トス」で情報を移動させる際、受け手を着地点から 2[m] の範囲内のユーザと限定していた。しかし、例えば user4 のように送り手と受け手の直線上に受け手の候補となる他のユーザがいない場合は、送り手が 2[m] の範囲を超えて大きく投げた際でも、user4 を受け手とする方がより人間の感覚にあっていると考えることができる。このように受け手を決定するルールを変更すると、今回の実験においては E2 では 7.5[%] 程度、E4 では 5.0[%] 程度の改善が見込まれる。

上述の改良などにより、成功率の向上が望まれるため、「トス」や「振る」動作によって情報の移動を行うことは十分に可能であると考えられる。

## 6 結論と今後の課題

本稿では Toss-It という、モバイルデバイスにおける情報の移動を直感的に実現するインタフェースを提案した。ユーザの動作の認識に慣性センサを、ユーザの位置や向きの認識にカメラを用い、実際に「トス」や「振る」動作で情報の移動を実現できることを示した。

今後の課題としては、情報を誤って渡すことをどのように防ぐか、ということが挙げられる。現在の Toss-It では情報の受け手が誰になったかを確認する

手段が送り手に提供されていないため、「トス」や「振る」動作で送った情報が、送り手が意図としない受け手によって盗み見られてしまう、といったことが起こりうる。このような点に関してどのような解決策が有効か、現在検討を進めている。

また、5.4 節で述べた考察などを基に、さらなる改良を加えた段階で、本研究が提案するインタフェースの評価を行う予定である。

さらに、「トス」や「振る」動作だけでなく、より複雑な動作を取得できるように、Toss-It の改良を進めることも考えられる。複雑な動作を取得できれば、本稿で示した情報の移動だけでなく、多種の操作を実現したり、エンタテインメントゲームへの応用なども考えられる。

## 謝辞

本研究の一部は、(財) 柏森情報科学振興財団の助成を受けて行われた。

## 参考文献

- [1] Harrison, B., L., et al., “Squeeze Me, Hold Me, Tilt Me! An Exploration of Manipulative User Interfaces”, *In Proc. of CHI*, pp. 17–24, 1998.
- [2] Hightower, J., “Location Systems for Ubiquitous Computing”, *IEEE Computer*, Vol. 34, No. 8, pp. 57–66, 2001.
- [3] Hinckley, K., et al., “Sensing Techniques for Mobile Interaction”, *In Proc. of UIST*, pp. 91–100, 2000.
- [4] Rekimoto, J., “Pick-and-Drop: A Direct Manipulation Technique for Multiple Computer Environments”, *In Proc. of UIST*, pp. 31–39, 1997.
- [5] Rekimoto, J., “Tilting Operations for Small Screen Interfaces”, *In Proc. of UIST*, pp. 167–168, 1997.
- [6] Swindells, C., et al., “That One There! Pointing to Establish Device Identity”, *In Proc. of UIST*, pp. 151–160, 2002.
- [7] Trucco, E. and Verri, A., “Introductory Techniques for 3D Computer Vision”, Prentice Hall, 1998.
- [8] Yee, K., “Peephole Displays: Pen Interaction on Spatially Aware Handheld Computers”, *In Proc. of CHI*, pp. 1–8, 2003.
- [9] Wilson, A. et al., “XWand: UI for Intelligent Spaces”, *In Proc. of CHI*, pp. 545–552, 2003.
- [10] 宮川 他, 「筆記運動の 3 次元慣性計測による筆記パターンの再現」計測自動制御学会論文集, Vol. 38, No. 1, pp. 1–8, 2002.