一般
投稿論文

センサ特性を考慮したスマートフォンアプリケーションに関する一考察

井上 晴可[†1]　窪田 諭[†1]　今井 龍一[†2]　田中 成典[†1]

[†1] 関西大学　[†2] 東京都市大学

スマートフォンには多数のセンサが搭載され，それらのセンサを利用するサービスが注目されている．センサは機種ごとに異なり，しかもセンサ精度が一定でないため，開発者がその特性を理解せずにアプリケーションを開発すると，機種によってはアプリケーションが正常に動作しないことがある．したがって，開発者は，センサの特性であるセンサ値の精度や計測間隔などを把握する必要がある．本研究では，アプリケーション開発者にスマートフォンのセンサに関する有益な情報を提供することを目的として，機種別，利用条件別，アプリケーション別の計測データを用いて特性を明らかにした．4 機種を用いて加速度，磁気，ジャイロ，重力と GPS の 5 種類のセンサのデータを取得する実験を行った結果，機種差分があること，センサ値の計測間隔は歩行時に高い精度で取得されることが分かった．そして，センサ特性がアプリケーションに及ぼす影響を定量的に把握するために，5 種類のセンサを用いて，3 つのアプリケーションを開発した．そこでは，センサ値の平滑化処理を施してノイズを除去し，各センサの取得結果に合った閾値を設定した結果，機種に依存することなくデータを取得することができた．

1. はじめに

スマートフォンに搭載されているセンサを利用したサービスには，ライフログサービスやITS（高度道路交通システム）などがある．ライフログサービスは，リアルタイムに取得可能なユーザの位置，行動や健康などの人に関する情報を自動で記録するもので，加速度センサ値を基に歩行時の歩数を算出する研究[1]などがある．ITSとスマートフォンを融合したサービスでは，自動車に固定したスマートフォンから収集した情報を基にヒヤリハットマップを生成し，Web サイトで公開[2]している．

スマートフォンに搭載されているセンサは機種ごとに異なるため，開発者がその特性を理解せずにアプリケーション（以下，アプリ）を開発すると，機種によってはアプリが正常に動作しないことがある．それは，機種ごとに取得するセンサ値の精度や計測間隔などの仕様が異なるためである．この課題を解決するために，開発者はセンサの特性を把握し，機種ごとのセンサ精度の差を考慮したキャリブレーションや許容誤差を設定する必要がある．本研究では，アプリ開発者にセンサ特性に関する有益な情報を提供することを目的として，機種別，利用条件別とアプリ別の計測データを分析してセンサの特性を定量的に明らかにする．実験の対象 OS は，多くのスマートフォンに利用されている Android[3],[4],[5] とする．

2. Android SDK で定義されるセンサ

Android SDK の Sensor クラスで定義されるセンサ[6]を**表 1** に示す．本研究では，加速度，磁気，ジャイロ，重力の各センサと LocationManager クラスで定義される GPS センサ[7]を対象とする．

Android SDK では**表 2** に示す計測間隔[8]が用意され，開発者は利用用途に合わせて計測間隔を指定する．同一時間で短い計測間隔を指定するとセンサの取得回数が増え，センサ値の変化を高い頻度で確認できる．SENSOR_DELAY_NORMAL（以下，NORMAL）は，ユーザ

表 1　Sensor クラスの定数

定数	センサ	取得データ	単位
TYPE_LINEAR_ACCELERATION	加速度	3 軸から受ける単位時間あたりの速度の変化率	m/s^2
TYPE_MAGNETIC_FIELD	磁気	3 軸の磁気強度	μT
TYPE_GYROSCOPE	ジャイロ	3 軸を中心とした回転速度	rad/s
TYPE_GRAVITY	重力	3 軸方向の重力	m/s^2

表 2　計測間隔の定数

定数	およその間隔
SENSOR_DELAY_NORMAL	200ms
SENSOR_DELAY_UI	60ms
SENSOR_DELAY_GAME	20ms
SENSOR_DELAY_FASTEST	0ms

インタフェースに適切な速度でセンサ値を取得する．SENSOR_DELAY_UI（以下，UI）は，画面の向きの変更に適する速度でセンサ値を取得する．SENSOR_DELAY_GAME（以下，GAME）は，ゲームのための最適な速度でセンサ値を取得する．SENSOR_DELAY_FASTEST（以下，FASTEST）は，できるだけ速い速度でセンサ値を取得する．計測間隔はセンサに依存するため，指定した計測間隔と実際の計測間隔が機種ごとに異なる課題がある．GPSセンサから位置情報を取得する計測間隔としては，時間間隔と距離間隔がある．本研究では，機種ごとのセンサ値の差異を定量的に評価する．

加速度センサは，X，Y，Z軸の単位時間あたりの速度の変化率を検出する．加速度センサ値には，重力加速度（約9.8 m/s^2）が含まれる．既存研究には，移動距離の推定[1]，歩行者の推定[9]，人の歩幅による混雑度の推定[10]，歩行時の心拍数の推定[11]がある．その課題としては，取得したいデータに合わせて微弱な振動を測定から除外することが必要になる点が挙げられる．

磁気センサは，X，Y，Z軸の電磁波の強さを検出する．既存研究には，周囲の金属や電子機器の磁束密度の変化を利用した非接触の入力手法[12]や加速度センサなどと組み合わせた人物の行動認識[13]がある．

ジャイロセンサは，X，Y，Z軸を中心とした物体の回転速度である角速度を検出する．既存研究には，曲がり角の推定[1]がある．その課題としては，曲がり角判定では，ジャイロセンサのキャリブレーション方法に依存する点が挙げられる．

重力センサは，X，Y，Z軸方向の重力を検出する．机などに置いた場合，重力方向にある軸の重力センサ値は重力加速度（約9.8 m/s^2）を取得し，その他の2軸では約0 m/s^{2}を取得する．重力加速度を計測するとスマートフォンの向きが分かる．

GPSセンサでは，緯度と経度を取得し，単位は度である．既存研究には，車両の走行軌跡データを用いた道路交通分析[14]，災害体験ゲームシステムの開発[15]などがある．

3. センサ精度の実験

3.1 実験方法

センサ精度は，取得するセンサ値と計測間隔に依存する．センサ精度を定量的に評価するために，表3の機種別と図1の利用条件別の実験を行う．Android OSは，機種AとBが4.x，機種CとDが2.xである．

3.1.1 Sensorクラスで定義されるセンサ

Sensorクラスで定義される加速度，磁気，ジャイロ，重力の各センサを対象に，4機種を用いて表2の計測間隔別と利用条件別の実験を行う．実験では，android.hardware APIを利用し，表1の定数によってセンサ値を取得する．利用条件には，利用頻度の高い固定静止，手持ち静止と手持ち歩行の3パターン（図1）を設定した．(a) 固定静止と (b) 手持ち静止では，スマートフォンの動きを検知するセンサが一定値を取得することを確認する．(c) 手持ち歩行では，動作を検知するセンサ値が機種ごとで異なること，そしてその他のセンサ値が固定静止および手持ち静止と同様の結果を得ることを確認する．固定静止では，平らな地面に設置した段ボール箱の上にスマートフォンを20秒間固定した．手持ち静止では，スマートフォンを右手で持ち脇を締め胸の高さで20秒間静止した．手持ち歩行では，スマートフォンを右手で持ち，10 mの距離を南北方向に0.9m/sの速度で歩いた．

3.1.2 GPSセンサ

アプリ開発者は，android.location APIを利用し，LocationManagerクラスで定義されるGPSセンサから位置情報を取得する間隔として時間間隔と距離間隔を設定する．Android Referenceによると，位置情報は，時間間隔と距離間隔の両方が満たされた場合に通知される．しかし，各機種において位置情報が，時間間隔と距離間隔の両方が満たされた場合に通知されるか，どちらか一方が満たされた場合に通知されるか明確ではないため，本実験によって検証する．本実験では，5パターンの時間間隔と距離間隔を設定（表4）した．

表3　搭載されているセンサ

センサ ＼ 機種	機種A (SC-03E)	機種B (ISW12HT)	機種C (SO-03C)	機種D (P-07C)
(1) 加速度	○	○	○	○
(2) 磁気	○	○	○	○
(3) ジャイロ	○	○	–	–
(4) 重力	○	○	○	○
(5) GPS	○	○	○	○

(a) 固定静止　(b) 手持ち静止　(c) 手持ち歩行

図1　利用条件の設定

実験(i)～(iii)では，同一位置にスマートフォンを固定し，設定した時間間隔で位置情報を取得できるかどうかを確認する．時間の通知間隔は0秒，300秒，600秒とする．実験(iv)と(v)では，スマートフォンをケースに入れて手で持ち，関西大学高槻キャンパス内の直線30mと50 mを歩行する．時間間隔を50m歩行するために十分な60秒に設定した場合に，その距離間隔で位置情報を取得できるかどうかを確認する．実験(i)～(v)では，4台同時に検証する．GPSセンサの精度では，国土交通省の街区三角点で取得したセンサ値の精度を比較した．

3.2　計測間隔の実験結果

3.2.1　Sensorクラスで定義されるセンサ

加速度，磁気，ジャイロおよび重力の各センサについて，4機種，3パターンの利用条件において，NORMAL，UI，GAMEとFASTESTを指定したときの結果を図2に示す．図2の縦軸は，センサ値を計測した間隔の平均時間（ms）を示す．

• NORMALの結果（図2（a））

機種AとBのすべてのセンサは，NORMALの指定どおり計測できた．機種Cは，固定静止と手持ち静止における磁気センサの平均時間が指定より2.1～3.4倍長くなった．機種Dは，固定静止と手持ち静止における加速度と重力センサが指定どおり計測できなかった．NORMALに指定すると，歩行の場合は計測間隔が約200msであるが，静止の場合に計測間隔が長くなる機種があった．歩行の場合には，NORMALに指定して利用できることが分かった．

• UIの結果（図2（b））

機種AとBのすべてのセンサは，UIの指定どおり計測できた．機種Cは，磁気センサの平均時間が指定より1.6～2.5倍長くなった．機種Dの加速度と重力センサは，固定静止と手持ち静止において5.8倍，手持ち歩行で1.4倍長くなった．磁気センサは，1.7倍になった．UIでは，歩行の計測間隔が約60msであるが，静止の計測間隔が長くなる機種があった．歩行の場合には，UIに指定して利用できることが分かった．

• GAMEの結果（図2（c））

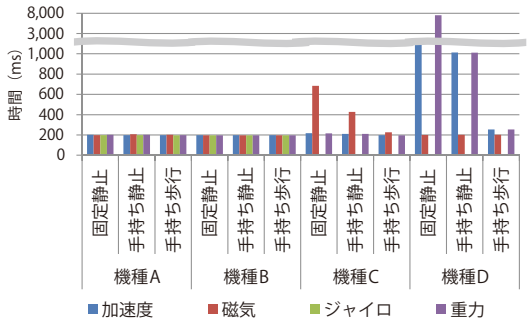
機種AとBのすべてのセンサは，GAMEの指定どおり計測できた．機種Cは，磁気センサが指定より5.1倍長くなった．機種Dのすべてのセンサは指定より1.6～5.4倍長くなった．GAMEでは，計測間隔が20msより長くなる機種があったが，NORMALやUIより短い間隔で計測できるため，高頻度でセンサ値を取得する場合に適することが分かった．

• FASTESTの結果（図2（d））

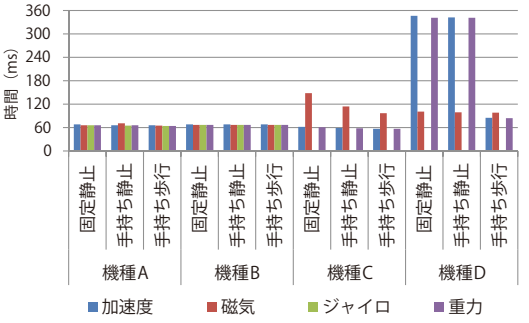
機種Aのジャイロセンサ，機種Bのすべてのセンサと機種Cの加速度と重力の各センサがGAMEより短くなる

表4　GPSセンサの設定値

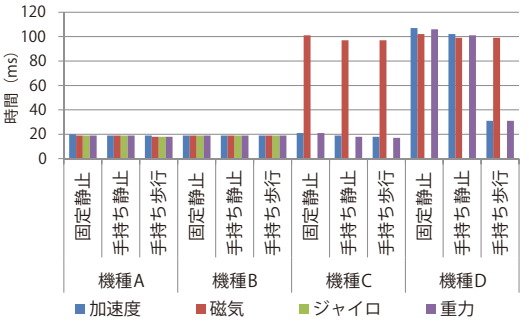
	i	ii	iii	iv	v
時間間隔(秒)	0	300	600	60	60
距離間隔(m)	0	0	0	30	50



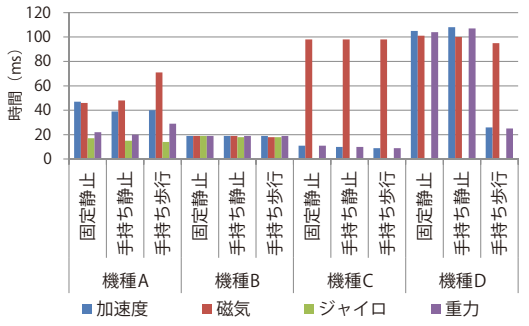
(a) NORMALの結果



(b) UIの結果



(c) GAMEの結果



(d) FASTESTの結果

図2　計測間隔の平均時間

が，機種Dはすべてのセンサの平均時間が20ms以上になった．FASTESTでは，GAMEよりも計測間隔が長くなる機種があり，機種ごとのばらつきが大きいため，計測間隔の指定として適さないことが分かった．

3.2.2 GPSセンサ

実験(i)～(v)の結果を表5に示す．表中では，GPSセンサから位置情報を取得したときの時間と距離の間隔を平均して示す．実験(i)では，機種A，BとDは時間間隔が0.82～0.83秒，距離間隔が0.00～0.19mとなった．実験(ii)では，時間間隔の平均が最小の機種Dでは8.11秒，最大の機種Bでは128.40秒となり，設定した300秒より短い時間間隔になった．実験(iii)では，時間間隔の平均が最小の機種Cでは4.26秒，最大の機種Aでは26.33秒となり，設定した600秒より短くなった．実験(ii)と(iii)では，設定した時間間隔どおりに位置情報を取得できなかった．

実験(iv)では，距離間隔の平均が最小の機種Dでは0.00m，最大の機種Bでは8.05mとなり，設定した30mより短くなった．実験(v)では，距離間隔の平均が最小の機種Aでは0.58m，最大の機種Bでは12.01mとなり，設定した50mより短くなった．実験(iv)と(v)では，設定した距離間隔どおりに位置情報を取得できなかった．

実験結果より，アプリ開発者が設定する時間間隔と距離間隔で必ずしも位置情報を取得できないことが分かった．各機種のGPSセンサから位置情報を通知する間隔は，Android Referenceの仕様どおりに動作するとはいえない．以上より，本研究では，時間間隔を0秒，距離間隔を0mに設定することとした．

3.2.3 バッテリの消費量

アプリの開発では，バッテリの消費量を考慮してセンサの計測間隔を指定する必要があるため，バッテリの消費量を計測間隔別に評価した．センサの計測間隔を短くすれば，バッテリの消費量が増えると考えられる．スマートフォンを満充電にし，バックグラウンドでほかのアプリを稼働せず，加速度，磁気，ジャイロと重力のセンサを同時に稼働し，1時間後のバッテリの消費量を計測間隔ごとに調査した．実験結果を図3に示す．機種Aでは，計測間隔をFASTESTに指定すると途中でアプリが停止した．すべての機種に共通して，センサの計測間隔を短くすると，バッテリの消費量が増えた．アプリ動作時のバッテリの消費量は，計測間隔だけでなく，利用時までのバッテリの充電頻度や使用年数，容量，ほかのアプリの動作によって異なる．そのため，複数のセンサを連続で利用するアプリでは，アプリ開発者は計測頻度の少ない計測間隔を指定することや利用者は充電しながら利用することが望ましい．

表5 時間間隔と距離間隔の平均

機種＼実験		i	ii	iii	iv	v
A	時間（秒）	0.83	21.69	26.33	7.40	15.80
	距離（m）	0.05	7.92	0.13	1.21	0.58
B	時間（秒）	0.83	128.40	6.26	7.50	16.20
	距離（m）	0.00	5.52	0.10	8.05	12.01
C	時間（秒）	7.33	119.73	4.26	4.00	9.00
	距離（m）	1.79	9.14	0.00	4.20	7.88
D	時間（秒）	0.82	8.11	9.52	1.00	18.50
	距離（m）	0.19	3.84	3.24	0.00	6.84

3.3 センサ値の実験結果

表3のセンサ（1）～（4）について，4機種，3パターンの利用条件と4つの計測間隔を指定し，機種ごとのセンサ値を比較した．固定静止と手持ち静止の利用条件では，安定して一定値を取得できるかどうかを確認するため，機種ごとのセンサ値の標準偏差を算出した．GPSセンサにおいて，設定どおりに位置情報を取得できるかどうかを確認する．

3.3.1 Sensorクラスで定義されるセンサ

• 加速度センサ

固定静止の標準偏差を図4に示す．固定静止では全機種が安定した値を取得した．固定静止でNORMALのZ軸の値を図5に示す．機種B～Dは約9.80m/s^{2}を取得するが，機種Aは約10.40m/s^{2}であり，ほかの計測間隔でも同様の結果になった．X軸とY軸の加速度センサ値では，機種Aが機種B～Dより大きかった．手持ち静止と手持ち歩行では，機種ごとにセンサ値に差があるが，計測結果（波形）は類似した．

加速度センサの精度は，計測間隔に依存しないが，機種ごとにセンサ値が異なる．計測間隔は，NORMAL，UIとGAMEからアプリの処理に合わせて指定する必要がある．

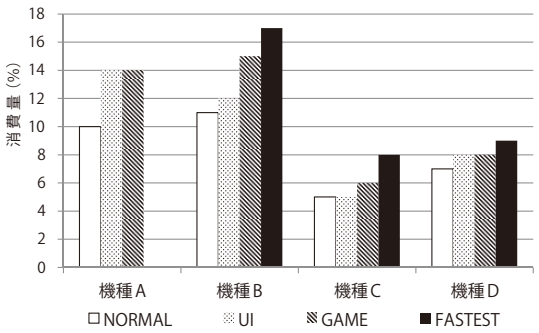


図3 バッテリの消費量

• 磁気センサ

固定静止の標準偏差を図6に示す．すべての計測間隔において各機種の標準偏差が1.6 μT以下になった．磁気センサは，すべての機種で指定する計測間隔に依存しなかった．固定静止でNORMALのZ軸の値を図7（a）に，手持ち歩行でGAMEのZ軸の値を図7（b）に示す．図7（a）より，機種ごとで約50 μTの差があった．これは，固定静止のX軸とY軸の磁気センサ値と手持ち静止の3軸の磁気センサ値においても同様の結果であった．図7（b）より，手持ち歩行では，その他の利用条件時と同様に機種ごとの差はあるが，方向転換を伴う10秒付近で約30 μT増加して機種ごとで類似した計測結果（波形）になった．

磁気センサの精度は計測間隔に依存しないが，機種ごとでセンサ値の差が大きい．そのため，磁気センサを利用する場合は，閾値の設定などの処理は行わない方がよい．計測間隔は，NORMALとUIを指定することが好ましい．

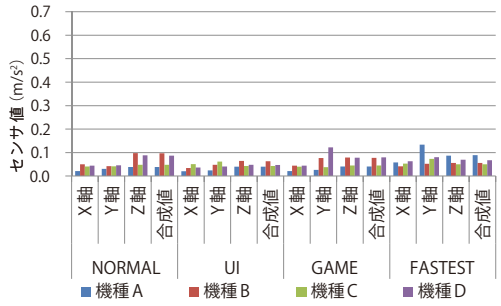


図4　加速度センサの標準偏差（固定静止）

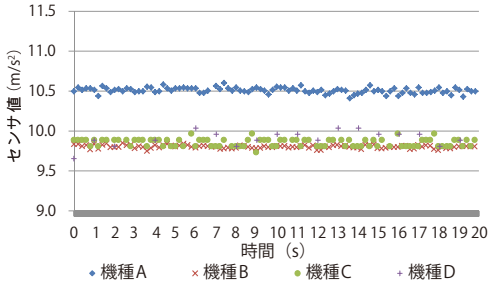


図5　固定静止 NORMAL の加速度センサ（Z 軸）

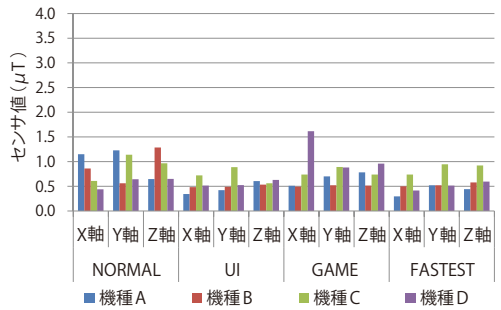


図6　磁気センサの標準偏差（固定静止）

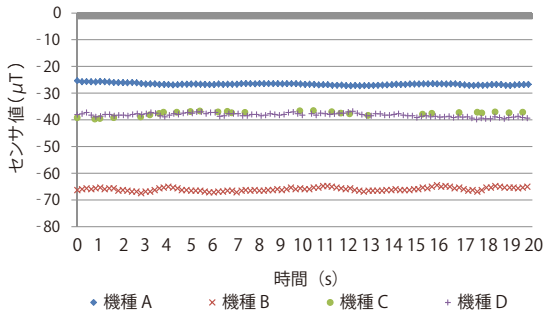
• ジャイロセンサ

固定静止の標準偏差を図8に示す．機種AはNORMALで標準偏差が0.01rad/s以下，その他の計測間隔で0.22rad/s以下になった．機種Bは，すべての計測間隔に共通して0.01rad/s以下になった．ジャイロセンサは，すべての機種で指定する計測間隔に依存しなかった．固定静止でNORMALのZ軸の値を図9（a）に，手持ち歩行でNORMALのZ軸の値を図9（b）に示す．図9（a）より，固定静止において機種ごとには差がなかった．X軸とY軸も同様の結果になり，すべての計測間隔に共通した．手持ち静止は，すべての計測間隔の各機種の特徴が固定静止と類似した．図9（b）より，各機種の差は小さく，計測結果（波形）が類似した．手持ち歩行では，方向転換を伴う10秒付近でセンサ値が約3.0rad/sになった．その他の計測間隔においても同様の結果になった．

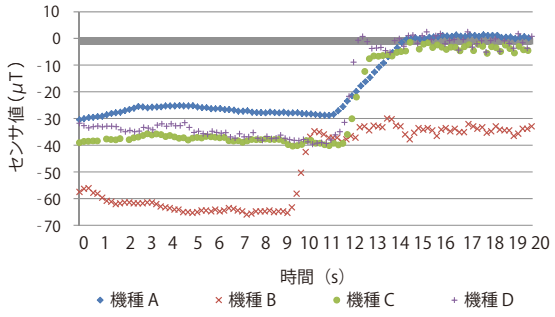
ジャイロセンサの精度は，計測間隔に依存せず，機種ごとで差が小さかった．計測間隔は，NORMAL，UIとGAMEからアプリの処理に合わせて指定するとよい．特に，NORMALの標準偏差が小さい．

• 重力センサ

固定静止の標準偏差を図10に示す．機種A～Cの標準偏差は，0.22 m/s²以下になり安定した値を取得した．機種Dは，NORMALとUIにおいて，標準偏差が331.6 m/s²以上になり



（a）固定静止 NORMAL のZ 軸



（b）手持ち歩行 GAME のZ 軸

図7　磁気センサ

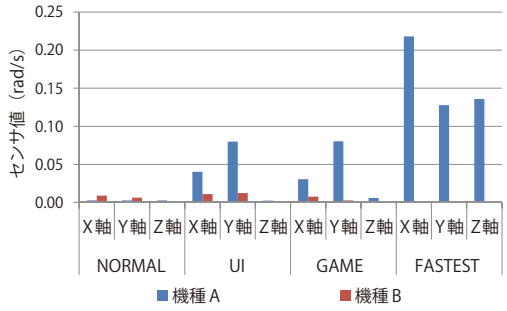
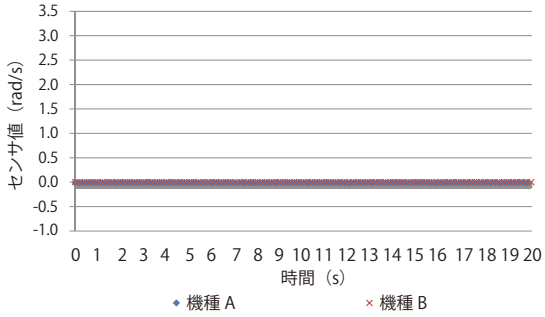
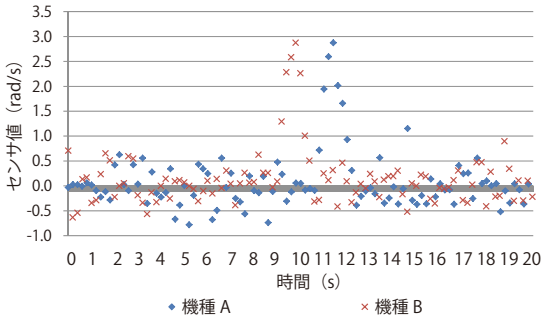


図 8　ジャイロセンサの標準偏差（固定静止）



（a）固定静止 NORMALのZ軸



（b）手持ち歩行 NORMALのZ軸

図 9　ジャイロセンサ

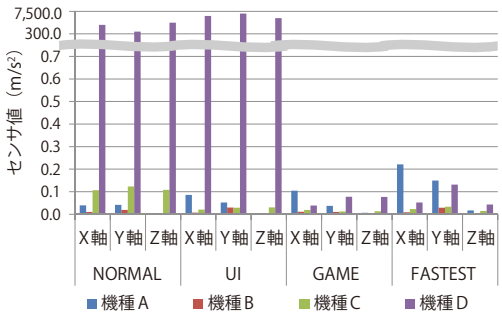
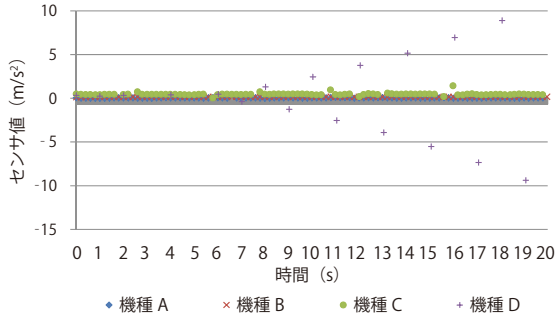
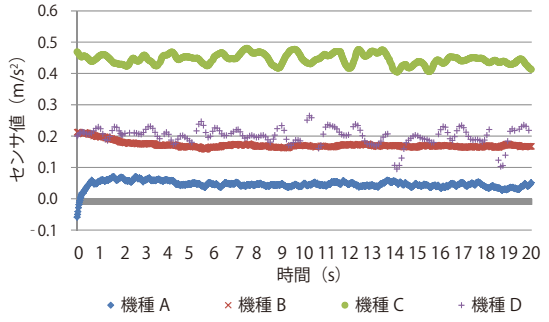


図 10　重力センサの標準偏差（固定静止）



（a）固定静止 NORMALの X 軸



（b）固定静止GAME の X 軸

図 11　重力センサ

ばらつきが大きくなった．機種DのGAMEとFASTESTは，その他の機種と類似した．固定静止でNORMALのX軸の値を図11（a）に，固定静止でGAMEのX軸の値を図11（b）に示す．図11（a）より，機種A～Cは約0 m/s² を取得したが，機種Dはセンサ値が0 m/s² を境に発散した．固定静止ではUIも同様の結果になった．図11（b）より，機種ごとのセンサ値の差は0.48 m/s² 以下になった．機種Dのセンサ値は，NORMALとUIを指定した場合は0 m/s² を境に発散し，GAMEとFASTESTを指定すると約0 m/s² を取得した．重力センサの精度は，指定する計測間隔に依存する機種があることが分かった．機種A～Cのセンサ値は，すべての利用条件で指定する計測間隔に依存しなかった．

重力センサの精度は，固定静止と手持ち静止で計測間隔に依存した．計測間隔は，NORMAL，UIとGAMEからアプリの処理に合わせて指定するとよい．ただし，7秒以上静止するアプリを開発する場合には，NORMALとUIを指定しないことが望ましい．

3.3.2　GPSセンサ

GPSセンサは，高架鉄道橋と高層ビルに囲まれた道路上の街区三角点を用いて実験した．正解データと取得値には，10～20mの誤差があった．GPSセンサの精度は，取得できる衛星の数に依存した．同時間帯では，機種ごとで取得する衛星番号は類似した．取得間隔は，設定どおりの間隔で位置情報を取得する可能性が高いと考える時間間隔を0秒，距離間隔を0mに設定することが好ましい．

4. アプリにおけるセンサ値取得精度の実験

4.1 実験方法

機種ごとのセンサ値の精度と計測間隔がアプリに及ぼす影響を確かめるために, 第3章で検証した加速度, 磁気, ジャイロ，重力とGPSのセンサを用いて表6に示すアプリを開発した．実験では, それぞれの利用場面を想定し, 歩数計アプリは手持ち歩行時，腹筋アプリは腹筋時，方向支援アプリは手持ち歩行時で検証した．また，バックグラウンドで動くアプリが，センサ値の計測間隔と実装したアプリに与える影響を確認するために，ほかの5つのアプリの動作有無によって実験した．

4.1.1 歩数計アプリ

歩数計アプリは，加速度センサと重力センサから取得したセンサ値の計測結果（波形）を利用して歩数を算出する．計測間隔は，NORMAL，UI とGAMEの3パターンを指定した．

まず，3軸の加速度センサ値からそれらの合成値Aを式（1）によって算出する．

$$A=\sqrt{Ax^2+Ay^2+Az^2} \tag{1}$$

AxはX軸, AyはY軸, AzはZ軸の加速度センサ値である．

次に，加速度センサ値には，センサの精度やセンサ感度，手振れによるノイズを含むため，ローパスフィルタを用いてノイズとなる高周波成分を取り除き，低周波成分のみ取り出して平滑化する．ローパスフィルタを用いて補正した加速度センサの合成値の算出方法を式（2）に示す．

$$S_t=0.9^{*}S_{t-1}+0.1^{*}A_t \tag{2}$$

Sは補正後の加速度センサの合成値，Aは補正前の加速度センサの合成値, tはセンサ値を取得した回数である．

加速度センサ値には，X，Y，Z軸の単位時間あたりの速度の変化率と重力加速度（約9.8 m/s²）が含まれる．歩数計アプリでは，歩行時の加速度の変化率を用いて歩数を算出するため，S_tから地球の重力加速度値を減算する．地球の重力加速度の減算には，3軸の重力センサで計測した10個の重力センサの合成値の平均値を用いた．計測間隔がNORMALのときでも10個取得するのに十分な時間として，アプリ開始時に3秒間静止することを条件とした．図11（a）より，計測間隔がNORMALとUIでは，7秒以上静止すると約0 m/s²を境に発散する機種があるため，静止時間を3秒間とした．重力センサの合成値の算出方法を式（3）に示す．

$$G=\frac{\sum_{t=1}^{10}\sqrt{Gxt^2+Gyt^2+Gzt^2}}{10} \tag{3}$$

tはセンサ値を取得した回数，GxはX軸，GyはY軸，GzはZ軸の重力センサ値である．補正後の加速度センサの合成値S_tから重力センサの合成値Gを減算して，歩行時の加速度センサの合成値を求める．除去する算出方法を式（4）に示す．

$$S_t'=S_t-\mathrm{G} \tag{4}$$

最後に，歩行時の加速度センサの合成値が波形になる特性を利用して歩数を算出する．歩行時の加速度センサ値を図12に示す．これは，歩行時の約3秒間を抜粋したものである．手持ち歩行において，波形の山と谷の値は機種ごとに異なるが，波形の山と谷の差は類似する．本研究では，波形の山と谷の差が4.0m/s²以上になる場合に1歩とする．実験では, 機種A～Dの4台を対象とし，同一人物がスマートフォンを片手で持ち，平坦な直線をできるだけ同じ速さで50歩歩いた．

4.1.2 腹筋アプリ

腹筋アプリは，ジャイロセンサから取得したセンサ値の計測結果（波形）を利用して腹筋回数を算出する．計測間隔は, NORMAL, UIとGAMEを指定した．実験では，ジャイロセンサが搭載されている機種AとBを対象とし，同一人物が腹上でスマートフォンを横向きに持ち，できるだけ同じペースで10回腹筋した．腹筋時は，スマートフォンを横にするため，変動が大きいY軸のジャイロセンサ値に注目する．ジャイロセンサ値は，センサの精度や感度，手振れによるノイズを含むため，式（2）の

表6　開発したアプリ

No.	アプリ	センサ
1	歩数計アプリ	加速度，重力
2	腹筋アプリ	ジャイロ
3	方向支援アプリ	加速度，磁気，GPS

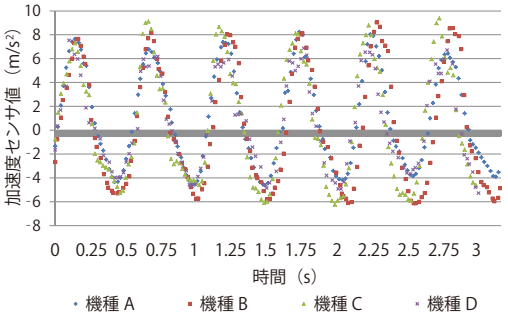


図12　歩行時の加速度センサ値

ローパスフィルタを用いて低周波成分のみを取り出す．次に，ノイズ除去後のY軸の値は滑らかな波形になる特性を利用して腹筋回数を算出する．腹筋時のジャイロセンサ値を図13に示す．これより，機種ごとでジャイロセンサ値の差が小さかったため，波形の山が1.5rad/s以上，谷が-1.5rad/s以下になる場合に腹筋1回と見なした．

4.1.3　方向支援アプリ

方向支援アプリは，加速度センサと磁気センサから取得した方位角およびGPSセンサから取得した位置情報から設定したスタート地点とゴール地点の距離，現在地とゴール地点の距離，スマートフォンの方位を表示する．方位は高頻度に取得する必要がないため，加速度センサと磁気センサの計測間隔はNORMALに指定した．実験では，加速度，磁気とGPSの各センサが搭載されている機種A～Dを対象とし，位置情報と方位が正しく取得できることを確認した．スタート地点をJR新大阪駅周辺のコンビニエンスストア，ゴール地点をJR新大阪駅に設定し，同一人物が4台同時に持って歩いた．

4.2　実験結果と考察

4.2.1　歩数計アプリ

歩数計アプリでは，平坦な直線を歩き，正しい歩数（50歩）を計測できるかを検証した．表7に示すように，機種ごとの歩数，計測間隔の平均時間と標準偏差，バックグラウンドで動くアプリの有無別に行った．機種A～Cは，指定した計測間隔から大きくばらつくことがなかった．機種Dは，すべての計測間隔においてその他の機種に比べて計測間隔の平均時間と標準偏差が長くなり，ばらつきが大きかった．UIとGAMEでは，すべての機種で正しい歩数を計測できたが，NORMALを指定すると計測頻度が少なく歩行時の波形特性を利用できないため，歩数を正確に計測できなかった．その要因は，計測間隔が約0.2秒になると，1歩（0.5秒）でセンサ値を2回しか取得できないため，波形を構成できないことによるものであった．

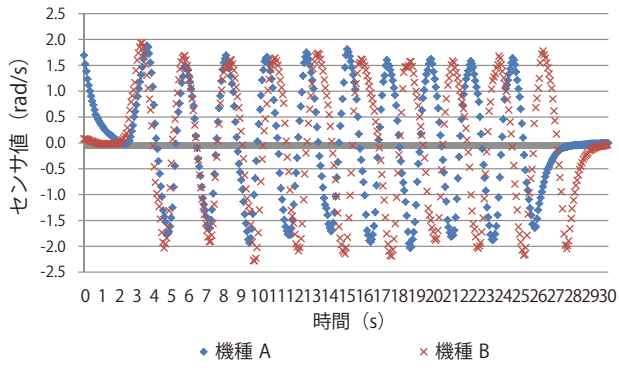


図13　腹筋時のジャイロセンサ値

バックグラウンドのアプリの動作有無における各機種の計測間隔について述べる．NORMALの平均時間の差は2.60ms以下，UIの平均時間の差は0.91ms以下，GAMEの平均時間の差は0.81ms以下であった．UIとGAMEでは，歩数が一致したため，バックグラウンドのアプリの影響を受けないと分かった．加速度センサを用いて歩数を算出する場合は，UIまたはGAMEに指定することが好ましい．

4.2.2　腹筋アプリ

腹筋アプリでは，正しい腹筋回数（10回）を計測できるかを検証した．表8に示すように，機種ごとの腹筋回数，計測間隔の平均時間と標準偏差，バックグラウンドで動くアプリの有無別に行った．腹筋1回につき約3秒の速さで検証すると，すべての計測間隔において，機種AとBで正しい腹筋回数を算出できた．機種AとBの計測間隔の平均時間と標準偏差について，差が小さかった．バックグラウンドのアプリの有無で比較すると，すべての計

表7　歩数の算出結果

計測間隔	バックグラウンドアプリの有無		機種A	機種B	機種C	機種D
NORMAL	無	歩数	8	12	47	37
		計測間隔の平均時間（ms）	199.70	199.42	199.46	233.08
		計測間隔の標準偏差	3.17	9.34	2.21	114.91
	有	歩数	8	13	45	38
		計測間隔の平均時間（ms）	202.30	198.80	199.23	232.52
		計測間隔の標準偏差	22.67	6.97	4.52	115.36
UI	無	歩数	50	50	50	50
		計測間隔の平均時間（ms）	66.80	68.97	59.52	68.25
		計測間隔の標準偏差	10.64	12.61	3.87	30.46
	有	歩数	50	50	50	50
		計測間隔の平均時間（ms）	67.18	69.31	59.74	67.34
		計測間隔の標準偏差	10.93	12.08	5.89	32.96
GAME	無	歩数	50	50	50	50
		計測間隔の平均時間（ms）	20.28	20.23	19.64	26.24
		計測間隔の標準偏差	9.53	8.22	2.16	15.80
	有	歩数	50	50	50	50
		計測間隔の平均時間（ms）	19.86	21.04	19.48	25.50
		計測間隔の標準偏差	4.47	9.01	1.76	13.87

表 8　腹筋回数の算出結果

計測間隔	バックグラウンドアプリの有無		機種 A	機種 B
NORMAL	無	腹筋回数	10	10
		計測間隔の平均時間 (ms)	199.53	199.25
		計測間隔の標準偏差	2.42	6.69
	有	腹筋回数	10	10
		計測間隔の平均時間 (ms)	199.41	199.19
		計測間隔の標準偏差	2.12	3.76
UI	無	腹筋回数	10	10
		計測間隔の平均時間 (ms)	66.16	69.38
		計測間隔の標準偏差	6.37	11.29
	有	腹筋回数	10	10
		計測間隔の平均時間 (ms)	66.14	69.44
		計測間隔の標準偏差	8.41	11.03
GAME	無	腹筋回数	10	10
		計測間隔の平均時間 (ms)	19.50	19.48
		計測間隔の標準偏差	6.37	11.29
	有	腹筋回数	10	10
		計測間隔の平均時間 (ms)	19.49	19.48
		計測間隔の標準偏差	4.82	3.34

測間隔で平均時間の差は0.12ms以下であり，すべての計測間隔で正しい腹筋回数を計測したため，その影響を受けないといえる．機種Aと機種Bは，すべての計測間隔において，計測間隔を指定どおり正確に得られなくても，正しい腹筋回数を計測できることが分かった．ジャイロセンサを用いて腹筋回数を算出する場合は，腹筋速度が速い人を考慮するとUIまたはGAMEに指定することが好ましい．

4.2.3　方向支援アプリ

方向支援アプリでは，位置と方位が正確であるかを検証した．位置情報の精度は，時間帯によって取得できるGPS衛星の数や利用する環境によって機種ごとの位置情報の精度が異なった．方位は，検証場所ではすべての機種が正しく取得できたが，建物などに接近した場合では正しく取得できなかった．また，計測間隔の指定は，NORMALの通知頻度で十分であった．

4.2.4　考察

実験では，加速度センサと重力センサを用いた歩数計アプリ，ジャイロセンサを用いた腹筋アプリ，加速度センサ，磁気センサおよびGPSセンサを用いた方向支援アプリを利用し，機種ごとに実装されているセンサが異なっても，アプリが開発者の意図したとおりに動作するかどうかを検証した．Androidスマートフォンの場合，実験結果から，計測間隔を厳密に指定することが難しいことが分かった．計測間隔の指定が必要な歩数計アプリと腹筋アプリにおいて，バックグラウンドで動くアプリの有無と計測間隔を変えて実験した．その結果，開発したアプリは，バックグラウンドで動くアプリの有無に関係なく稼働し，指定した計測間隔どおりにセンサ値を取得できなくても，正しい結果を出力した．

歩数計アプリでUIとGAMEを指定した場合は，センサ値に平滑化処理を施してノイズを除去し，閾値を設定したことにより，機種ごとのセンサ値に依存せずに歩数を計測できることを示した．腹筋アプリでは，センサが取得するノイズを除去することにより，正確な腹筋回数を計測できた．特に，スマートフォンを腹上で横向きに持ち，3秒に1回のペースで腹筋して計測したが，計測間隔をNORMAL，UIとGAMEに指定して正しく計測できたため，腹筋速度を変えても正確に計測できるといえる．方向支援アプリでは，市街地で実験して正しい結果が得られたため，GPSセンサの位置通知間隔として時間間隔を0秒，距離間隔を0mに設定することが有効であり，屋外であれば利用できると考える．

スマートフォンのアプリでは，バックグラウンドのアプリの状況や利用環境によって同一機種，同一端末でも常に同じ計測間隔を指定どおりに得ることは保証されない．本研究では，計測間隔の指定どおりにセンサ値を取得できなくても，歩数計アプリでは加速度と重力の各センサをUIまたはGAME，腹筋アプリではジャイロセンサをUIまたはGAME，方向支援アプリでは加速度，磁気とGPSの各センサの取得間隔を設定し，平滑化処理を施すことによりアプリが正確に計測結果を出力することを示した．これは，センサ値が大量に取得された場合に，指定した計測間隔に近い値であれば，アプリに提案手法を適用できることを意味する．ただし，歩数計アプリは，スマートフォンを手に持って平坦な直線を歩行した場合のみを対象としたため，曲がり角や坂道，スマートフォンを身につけた場合など条件を変えて正しい歩数を計測できるかどうかを確認する必要がある．

5.　おわりに

本研究では，アプリ開発者にスマートフォンのセンサ精度に関する有益な情報を提供することを目的として，機種別，利用条件別，アプリ別の計測データを用いて機種ごとのセンサ値と計測間隔を考察した．まず，固定静止，手持ち静止，手持ち歩行の利用条件を設定し，加速度，磁気，ジャイロと重力の各センサの計測間隔とセンサ値の精度を検証した．その結果，計測間隔の精度は，利用条件によって異なった．

次に，歩数計アプリ，腹筋アプリと方向支援アプリを

開発した．センサ値の平滑化処理を施してノイズを除去し，各センサの取得結果に適する閾値を設定した結果，機種に依存することなく，センサ値を正しく処理することができた．

今後の課題は，異なるOSに適用することと，同一機種のOSをバージョンアップした場合の検証を行うことである．そして，機種ごとのセンサ特性を把握した上で，膨大なスマートフォンのセンサ値の解析を目指す．

参考文献

1) 鈴木惇也，秋山征己，田中　博，五百蔵重典：スマートフォン内蔵センサを用いた歩行位置推定に関する基本実験と評価，高度交通システム（ITS）研究報告，情報処理学会，Vol.51, No.10, pp.1-7 (2012).
2) DENSO：スマホを活用したヒヤリハットマップ生成，http://www.globaldenso.com/en/newsreleases/events/tokyomotorshow/2013/booth/pdf/safety/22_Generating%20Near-miss%20Maps%20Using%20Smartphones.pdf（2014 年 11 月 11 日現在）
3) インプレス R&D インターネットメディア総合研究所編：スマホ白書 2012，インプレスジャパン (2012).
4) インプレス R&D：スマートフォン／ケータイ利用動向調査 2013，http://www.impressrd.jp/news/121120/kwp2013（2014 年 11 月 11 日現在）
5) 総務省：情報通信白書平成 24 年版，ぎょうせい (2012).
6) Google : Sensor, http://developer.android.com/reference/android/hardware/Sensor.html（2014 年 11 月 11 日現在）
7) Google : LocationManager, http://developer.android.com/reference/android/location/LocationManager.html（2014 年 11 月 11 日現在）
8) Google : SensorManager, http://developer.android.com/reference/android/hardware/SensorManager.html（2014 年 11 月 11 日現在）
9) 岩本健嗣，杉森大輔，松本三千人：3 軸加速度センサを用いた歩行者推定手法, 情報処理学会論文誌, Vol.55, No.2, pp.739-749 (2014).
10) 米村　淳，荻市智彦，井戸上彰，小花貞夫：スマートフォンを用いた人の混雑度推定手法の提案と評価，モバイルコンピューティングとユビキタス通信研究会研究報告，情報処理学会，Vol.67, No.5, pp.1-8 (2013).
11) 隅田麻由，水元旭洋，安本慶一：スマートフォンを用いた歩行時心拍数推定法, 情報処理学会論文誌, Vol.55, No.1, pp.399-412 (2014).
12) 山本涼太，宮下芳明：イヤホンを用いたスマートフォンの操作と個人認証，インタラクション 2013，情報処理学会，Vol.17, pp.626-631 (2013).
13) 加藤大智, 山岸弘幸, 鈴木秀和, 小中英嗣, 渡邊　晃:スマートフォンとセンサを活用したリモート見守りシステムの提案，マルチメディア，分散協調とモバイルシンポジウム 2011 論文集，情報処理学会，Vol.2011, pp.691-696 (2011).
14) 太田恒平，大重俊輔，矢部　努，今井龍一，井星雄貴：携帯カーナビのプローブ交通情報を活用した道路交通分析，土木計画学研究・講演集，土木学会，Vol.47, pp.1-12 (2013).
15) 浦野　幸，于　沛超，遠藤靖典，星野准一：実環境における災害体験ゲームシステムの開発，情報処理学会論文誌，Vol.54, No.1, pp.357-366 (2013).

井上 晴可（学生会員）k795202@kansai-u.ac.jp
1989 年生．2012 年関西大学総合情報学部総合情報学科卒業．2014 年同大学院総合情報学研究科知識情報学専攻博士課程前期課程修了．現在，同大学大学院総合情報学研究科総合情報学専攻博士課程後期課程在学中．

窪田 諭（正会員）skubota@kansai-u.ac.jp
1975 年生．1998 年関西大学工学部土木工学科卒業．2000 年同大学院工学研究科土木工学専攻博士課程前期課程修了．同年，（株）オージス総研入社．2008 年岩手県立大学ソフトウェア情報学部講師．2013 年関西大学環境都市工学部都市システム工学科准教授，現在に至る．地理情報システム，社会基盤情報の研究に従事．博士（工学）．

今井 龍一（正会員）imair@tcu.ac.jp
1975 年生．1998 年関西大学工学部土木工学科卒業．2000 年同大学院工学研究科土木工学専攻博士課程前期課程修了．同年，日本工営（株）入社．2009 年，博士（工学）東京大学．2010 年から 2015 年まで，国土交通省国土技術政策総合研究所防災・メンテナンス基盤研究センターメンテナンス情報基盤研究室研究官．2013 年から 2015 年まで，関西大学大学院総合情報学研究科連携大学院客員教授．2015 年から東京都市大学工学部都市工学科准教授，現在に至る．

田中 成典（正会員）tanaka@res.kutc.kansai-u.ac.jp
1963 年生．1986 年関西大学工学部土木工学科卒業．1988 年同大学院工学研究科土木工学専攻博士課程前期課程修了．同年，（株）東洋情報システム（現在，TIS（株））に入社．1994 年関西大学総合情報学部専任講師．1997 年助教授，2004 年教授，2006 年から学生センター副所長，現在に至る．2002 年から 1 年間, カナダの UBC にて客員助教授．博士（工学）．2000 年（株）関西総合情報研究所を起業, 設立当初から現在まで取締役会長．

投稿受付：2014 年 12 月 26 日
採録決定：2015 年 6 月 29 日
編集担当：高橋　修（佐賀大学）