

平成 23 年度 修 士 論 文

修士論文題目

エンタテインメントシステムにおける
加速度センサを用いたユーザ解析と
非言語評価手法の提案

指導教員 白井 暁彦

神奈川工科大学 情報工学専攻

学 籍 番 号 1085020

学 生 氏 名 加藤 匠

提 出 日 平成 24 年 1 月 27 日 指 導 教 員 印

受 理 日 平成 24 年 月 日 情報工学専攻主任 印

論文要旨

本論文は「エンタテインメントシステムにおける加速度センサを用いたユーザ解析と非言語評価手法の提案」に関する研究について、まとめたものである。

序論である第1章では、本研究が扱う課題、Sustanime プロジェクトの概要や目的、本研究の目的について述べる。

第2章では、加速度センサを扱った研究や、ユーザ解析を行っている関連研究について、それぞれの研究や提案システムが持つ利点や問題点についても述べる。

第3章では、序論で示した課題について、スマートフォンアプリと加速度センサを用いた姿勢判別実験を行った。「立った状態」、「座った状態」、「歩いている状態」の3つの動作中にアプリの画面を視聴させ、その加速度を取得し、解析した。結果、重力加速度を解析することで座り込む動作や立ち上がる動作、歩行状態から静止状態になる瞬間が読み取りやすくなり、姿勢判別できる可能性があることがわかった。

加えて映像視聴実験を報告する。2つの抽象的なアニメーション作品「雲散霧消」と「阿吽二字」を16人の被験者に立姿勢で視聴させ、その加速度を取得し、解析した。取得した加速度から確率的な雑音を消すためにローパスフィルタ、ハイパスフィルタによるフィルタリングも実装している。結果、「雲散霧消」視聴中、全被験者を通して3箇所ですばやんな挙動が多く発生する傾向が見られた。作者である早川貴泰氏のヒヤリングによって、3箇所とも設計上において狙ったハイライトシーンであり、映像視聴において本手法を用いることで作家の意図する部分で非言語フィードバックを得られる可能性があることがわかった。

第4章では、ユーザから取得できる「デバイス情報」、「バッテリー情報」について述べる。エンタテインメントシステムへの応用についても述べ、今後の課題についてもまとめる。

結論として、今後も継続して研究を続け、動的にユーザの状態を解析することで必ずユーザが満足するようなコンテンツの開発の可能性について述べている。

目 次

1. まえがき	1
2. 関連研究	3
3. 加速度センサを用いたユーザ解析と非言語評価手法	8
3.1 スマートフォンの加速度センサ	8
3.2 非言語評価手法	9
3.2.1 現在時間の取得	10
3.2.2 デバイスの向きを取得	11
3.3 姿勢判別実験	12
3.3.1 姿勢判別実験：プロトコル	12
3.3.2 姿勢判別実験：言語的アンケートによる結果	16
3.3.3 姿勢判別実験：加速度による評価	16
3.4 抽象的な動画を用いた実験	21
3.4.1 抽象的な動画を用いた実験：プロトコル	22
3.4.2 加速度計測実験：設問結果	24
3.4.3 加速度計測実験：加速度の解析結果	25
4. エンタテインメントシステムへの応用	28
4.1 ユーザから新たに取得する情報	28
4.1.1 デバイス情報を取得	29
4.1.2 バッテリー情報を取得	30
4.2 まとめと今後の課題	31
5. むすび	32

参考文献

付録1. 第15回日本バーチャルリアリティ学会大会：

「摂動応答と重心動揺計を用いた嗜好画像のリアルタイム推定手法の提案」

2. エンターテインメントコンピューティング2011：

「加速度センサを用いたエンタテインメントシステムの非言語評価手法の提案」

1. まえがき

近年、ゲームやアプリ等のデジタルコンテンツやサービスが数多く創出されている。ポートフォリオ(作品集)に関する書籍も増えており、メディア芸術を目指す若手作家や学生が急増している^{1) 2) 3)}。しかし作家にとって面白い作品、感動する作品を開発できたとしても、必ず商業として成功するとは限らないという難しい課題がある。また、作品を見る機会が限られていることや、YouTube、ニコニコ動画、vimeoといったネットワーク動画サイトで作品が無料で消費され、金銭的な問題も大きく、継続性の確保は難しい^{4) 5) 6) 7) 8) 9)}。そこで顧客が必ず満足する作品を生み出すことと、作家自身が持続的・継続的に活動できる地盤を作ることができれば、創作活動維持において価値が高い。

そこで持続的に活動できる環境の構築として、白井研究室では文化庁平成22年度メディア芸術人材育成事業「Sustanime プロジェクト」を受託、実施した¹⁰⁾。本プロジェクトでは作家が持続的に活動し、セルフプロデュースできるよう iPhone/iPad を用いた動画再生ができる。費用や制作時間のかかるポートフォリオ(作品集)に対し、「デジタルポートフォリオ」としてのアプリを作成でき、自らの映像作品を全世界に公開できる。「Sustanime プロジェクト」についてまとめたコンセプト図を図1.1に示す。

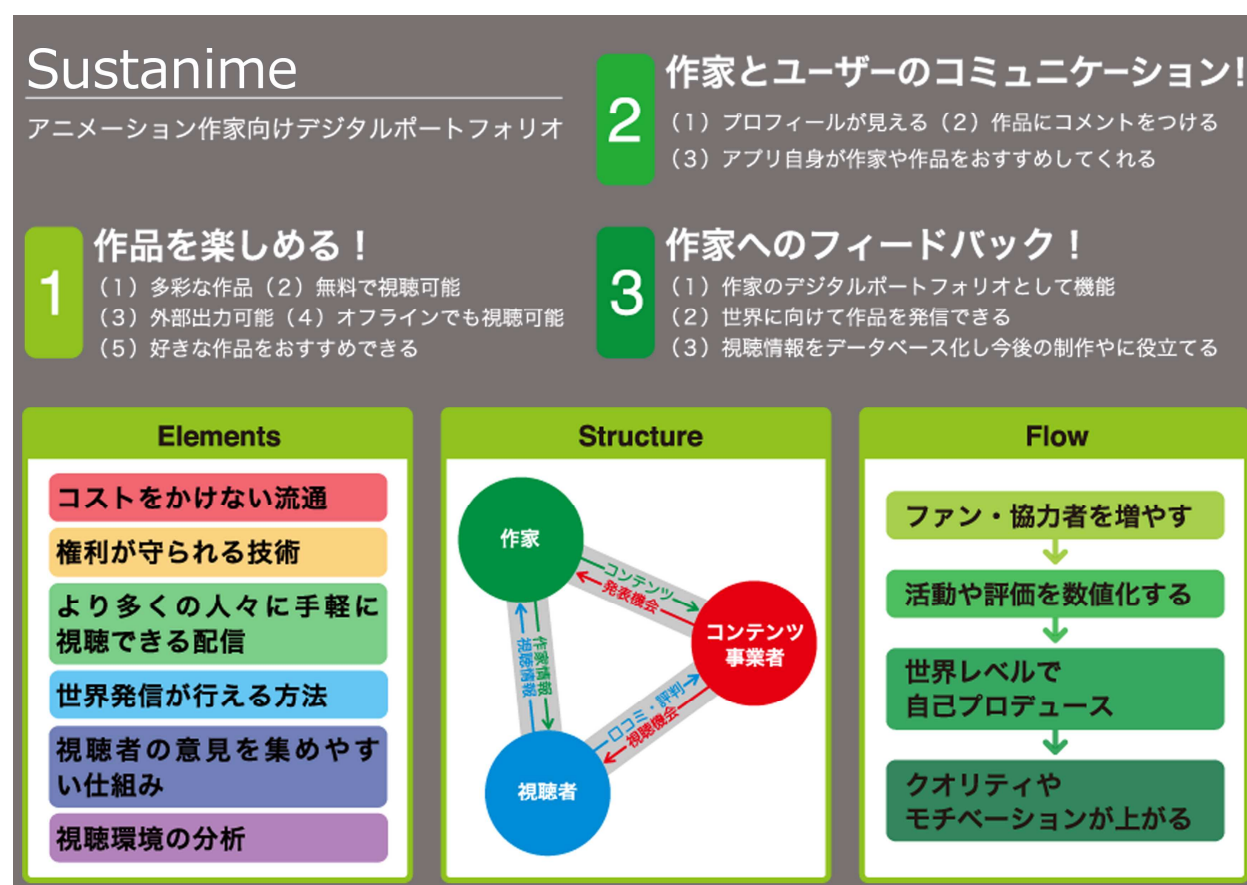


図1.1 「Sustanime プロジェクト」コンセプト図

「Sustanime プロジェクト」では、メディア芸術において受け手であるユーザは閲覧もしくは消費する方法しか与えられておらず、作家への還元方法を持っていないことが多いことを問題点として挙げ、ユーザの視聴情報を作家に「還元」することが作家にとってモチベーションと作品のクオリティ向上にもつながるというコンセプトで研究を進めている。この還元には作品から受けた感動、面白かった、つまらなかったといった心情的・フィードバックを伝えることだけでなく、視聴者がどのような姿勢で、どのような時間帯に視聴しているかといった、作品に対する物理的姿勢そのものも含まれている。

本論文では **Sustanime** プロジェクトでの開発事例を基に、エンタテインメントシステムのユーザ解析と非言語評価手法の提案を行う。具体的には動画視聴中止のイベント、加速度センサから得られる加速度を回収、解析することで、ユーザの挙動を通して意識的な入力を強いることなく非言語で自然に測定することができる。ユーザから得られた視聴情報を意味のある情報として作家に還元する手法を提案する。

2. 関連研究

消費者から作家に視聴情報を「還元」という課題において、言語的なレビューやアンケートではユーザから具体的で有益な情報を得ることは難しく、作家がフィードバックを得られない可能性がある。一例として、YouTube でのコメント欄と、AppStore で配信されている iPhone アプリ「ニコニコ動画」のレビュー欄を図2.1に示す。



図2.1 左：YouTube，右：AppStore「ニコニコ動画」レビュースクリーンショット

YouTube の例を見ると、まずコメントを投稿するためにはアカウントを作成しなければならない手間がかかる。7638回再生(2012年1月26日現在)されていても、1人のユーザからもコメントが得られておらず、レビューに参加しないユーザからフィードバックを得ることはできない。ユーザがどのような環境で映像を視聴しているかも不明確であり、そもそも映像を見ていない可能性もある。作品の先進性からコメントし難いものもあるだろう。AppStore の例では、意見を得たとしても、それが作家側から考えると全ての情報が必ずしも有益とはいえない。このような言語的なレビューではユーザから具体的で有益な

情報を得ることは難しく，作品の向上につながるフィードバックは得られない。

そこで，本研究ではスマートフォンに搭載された加速度センサを用いてユーザから加速度を取得し，解析する．アンケートを用いて主観的にコンテンツを評価するのではなく，物理的に評価ができればコンテンツ開発においてモチベーションの維持やクオリティの向上につながるだろう．本研究において(1)ユーザがどのようにして作品を消費しているのかわからない，(2)言語的なアンケートではフィードバックを得るのは難しい，(3)取得したデータを作家に還元，といった課題が考えられる．図2.2に3つの課題と手段を示す．

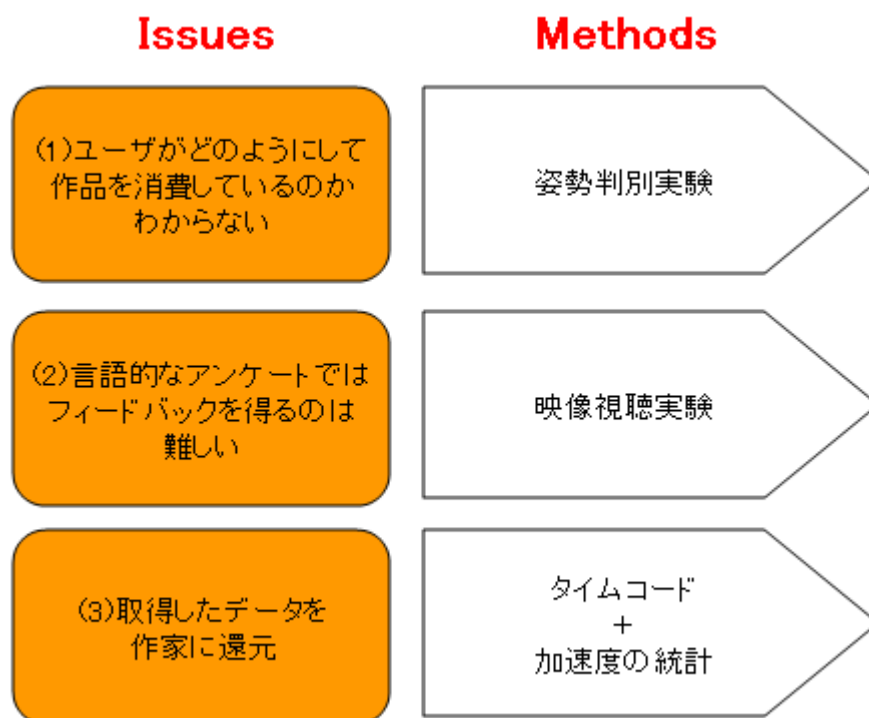


図2.2 本研究の課題

ユーザから取得した情報を解析することで新たな知見を得られる可能性は高い¹¹⁾．服部らはユーザの行動解析に加速度を用い，サーバ側にデータを蓄積し，意味を抽出している¹²⁾．

田淵らの研究「無線加速度センサを用いた人の日常行動識別におけるデータ収集条件の影響評価」では左手首，右手首，左腿，右腿，左足首，右足首の最大6箇所 に加速度センサを装着し，立つ，座る，歩く，階段：上りなどの10種類の行動を識別し，「両手首・両足首」の4箇所の装着位置で高い識別結果(89.8%)を得ている¹³⁾．しかしユーザに装置を取り付ける必要があり，省電力化のための最適なセンサ設計も行う必要がある．

NHK 放送技術研究所は「テレビ視聴インターフェース UTAN の提案」を行っている¹⁴⁾．コンセプト図を図2.3に示す．この提案は誰がどのように，どのテレビ番組を見ているかの視聴状況を推定し，状況に合った情報を提示することで，テレビ視聴をきっかけとして多様なコンテンツを提供しているものである．インターフェースとしてタブレット端末も使い，テレビと連携もできる．ユーザの視聴状況の推定には顔画像認識，表情認識，動作認識技術を用いている(図2.4)．しかしながら各種認識を行うためにはカメラやコン

ピュータ等の機器が必要になる。また、認識精度についても考えるべき点は多く、コストもかかる。認識を妨げないように顔を見せる必要もあり、常にユーザの顔にカメラを向けることになるため、ユーザが自然な状態であるとはいえない。

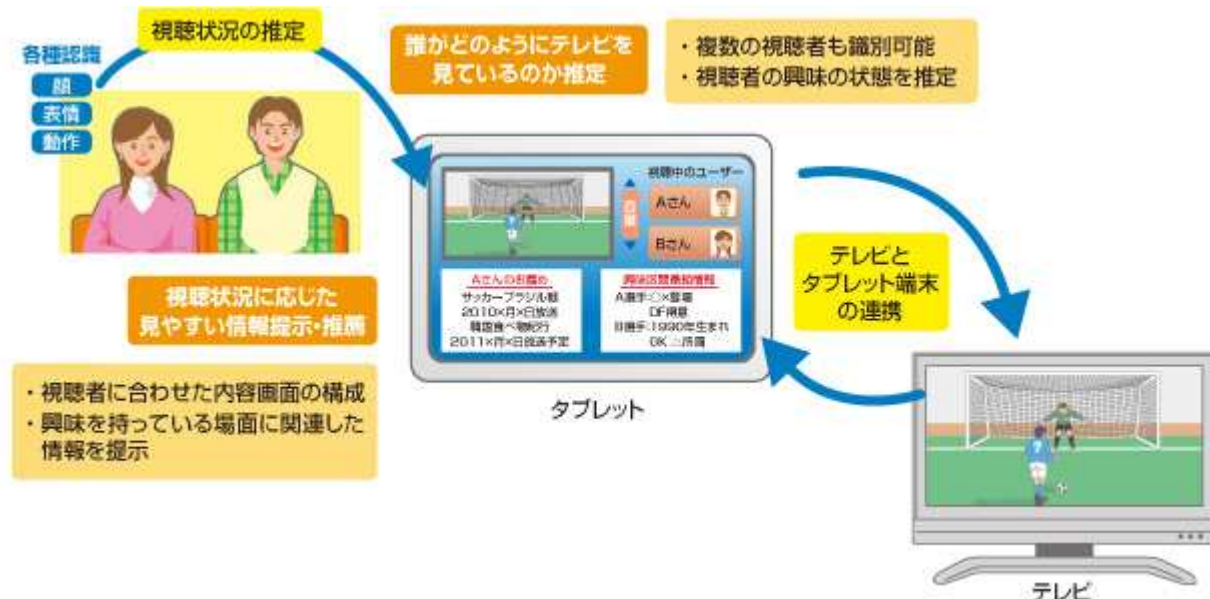


図2.3 NHK 放送技術研究所による提案(コンセプト図)¹⁴⁾



図2.4 NHK 放送技術研究所による提案(各種認識技術)¹⁴⁾

IVRC2010作品「ひとめぼれ実験装置」では、コンセプト「ユーザの意識／無意識下の嗜好を予測し、それを適切な形で提案すること」を目的として、摂動を加えた被験者の重心動揺を計測する事で2つの画像対に対する嗜好を意識的な入力を強いることなく、リアルタイムに検出するシステムについて提案した¹⁵⁾。重心を利用することで手掌や言語を使

用せず，自然な状態での測定が可能になるが，加えて被験者の上体に摂動（perturbation）を加えた状態での重心に注目することで恣意的な回答を無効にし，より明確な定位反応を取得することを特徴としている．しかしながら装置全体が大規模すぎるという問題があった．「ひとめぼれ実験装置」の全体図(装飾無し)と構成図を図2.5と図2.6に示す．



図2.5 「ひとめぼれ実験装置」全体図(装飾無し)

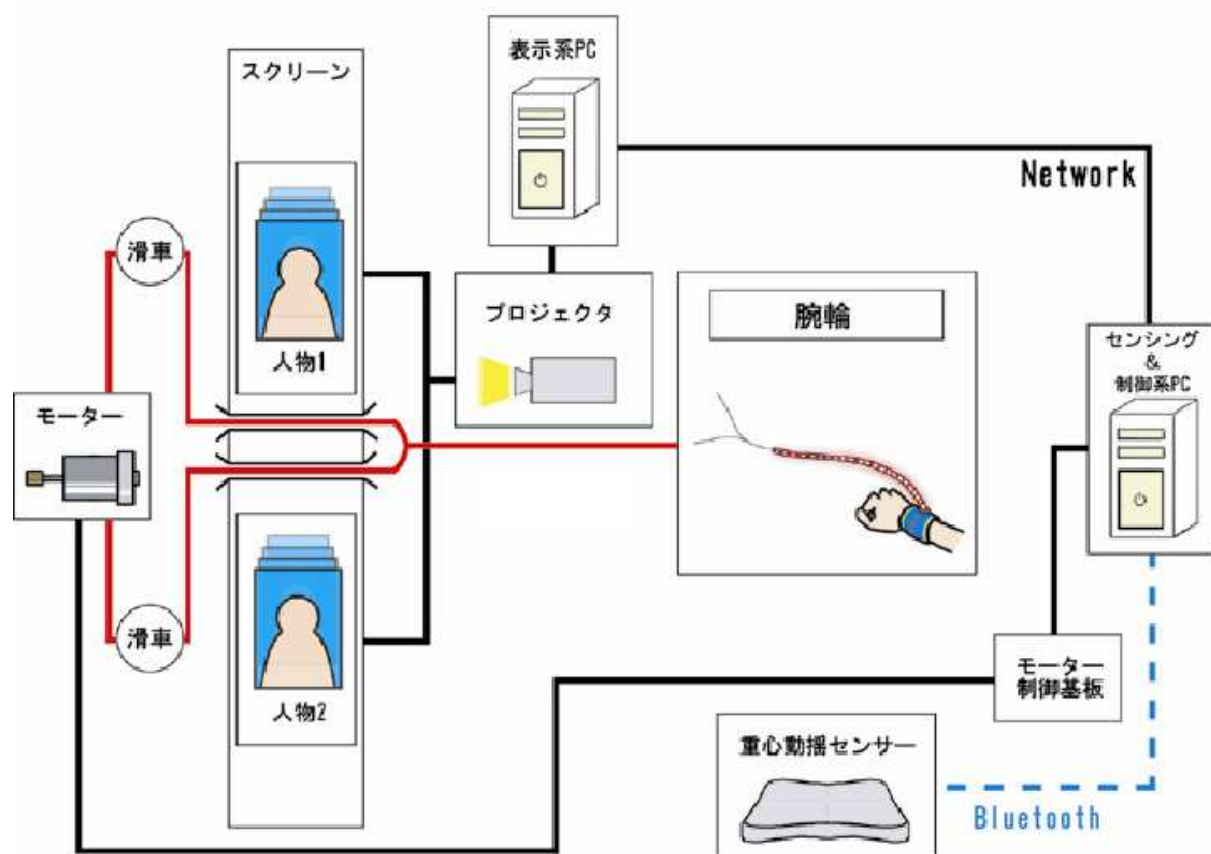


図2.6 「ひとめばれ実験装置」システム構成図

3. 加速度センサを用いたユーザ解析と非言語評価手法

本章では、代表的なスマートフォンである iPhone3GS に標準機能として搭載される高精度加速度センサから取得したデータに注目し、映像視聴時におけるユーザの振る舞いや挙動を意識的な入力を強いることなく測定できる非言語評価手法について報告する。スマートフォンに搭載された加速度センサを利用することで、言語を使用せず、被験者に特別な装置や器具を装着することなく、自然な状態での計測が可能になる。

3.1 スマートフォンの加速度センサ

本論文の実験に用いる iPhone3GS では ST Microelectronics 社製の加速度センサ「LIS331DLx」が搭載されており、加速度を最小で 16.2mg/digit まで取得することができる高精細センサである。このデータを人間の意志で制御することは難しく、恣意的でないといえる¹⁶⁾。加速度センサの各軸の方向については図 3.1 に示す¹⁷⁾。

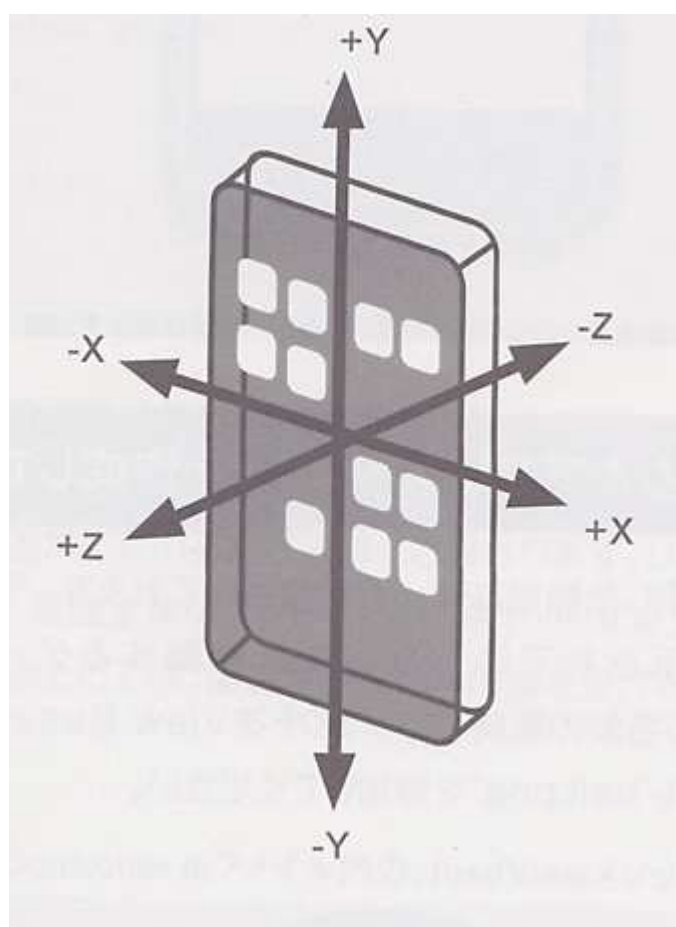


図3.1 加速度センサの各軸の方向¹⁷⁾

3.2 非言語評価手法

前提として、完全に静止した姿勢を取る事は人間には不可能である．映像の視聴に関わらず静止状態のユーザはわずかに動いており、確率的な雑音を含むデータであるとみなすことができる．本論文では確率的な雑音を消すためにローパスフィルタによるフィルタリングを実装している．ローパスフィルタについて図 3.2 に示す．

```
gx = acceleration.x * 0.1 + gx * (1.0 - 0.1);
gy = acceleration.y * 0.1 + gy * (1.0 - 0.1);
gz = acceleration.z * 0.1 + gz * (1.0 - 0.1);
```

図3.2 ローパスフィルタ(重力加速度)

本論文では 0.1 秒に 1 回加速度を取得しており、iPhone プラットフォーム(以下、iOS)では `acceleration` がそれぞれ iPhone が取得した加速度(x, y, z)である．ローパスフィルタでは取得してきた加速度に 0.1 を掛けることで、急に取得されるような大きな加速度による急激な値の変化を抑えている．そして 0.1 秒前に取得されていた重力加速度(gx, gy, gz)に(1.0 - 0.1)を掛けたものを足すことで、一つ前の重力加速度からは大きく値が変化しないようにしている．これによって確率的な雑音を切り捨てることができる．

次にハイパスフィルタについて図 3.3 に示す．

```
ax = acceleration.x - gx;
ay = acceleration.y - gy;
az = acceleration.z - gz;
```

図3.3 ハイパスフィルタ(重力を差し引いた加速度)

ハイパスフィルタでもローパスフィルタと同様で、`acceleration` がそれぞれ iPhone が取得した加速度(x, y, z)である．取得した加速度から重力加速度(gx, gy, gz)を引くことで、瞬間的な加速度(ax, ay, az)が取得できる．これによって瞬間的な身体反応が取得できる．

本論文では重力加速度(gx, gy, gz)を式(1)を用いて $G_{\text{Magnitude}}$ (以下、 G_{Mag})として利用し、瞬間的な加速度(ax, ay, az)には式(2)を用いて、 $A_{\text{Magnitude}}$ (以下、 A_{Mag})として利用した．

$$G_{\text{Mag}} = \sqrt{g_x^2 + g_y^2 + g_z^2} \dots\dots (1)$$

$$A_{\text{Mag}} = \sqrt{a_x^2 + a_y^2 + a_z^2} \dots\dots (2)$$

3.2.1 現在時間の取得

iOS では NSDate クラスを利用することで日時の情報を扱うことができ、現在時間の取得も可能になる¹⁸⁾ ¹⁹⁾。本研究では0.1秒毎に時間取得を行うようにタイマーを設定している。設定部分のソースコードを図3.4に示す。現在時間は年、月、日、時、分、秒、ミリ秒まで取得を行い、それらを文字列としてテキストに書き込んでいる。今回実装したプログラム「driveClock」を図3.5に示す。実際に実行し、作成されたテキストを図3.6に示す。

```
//タイマーのセット (0.1秒)
[NSTimer scheduledTimerWithTimeInterval:0.1 //タイマーを発生させる間隔 (0.1秒毎)
                                target:self //メソッドがあるオブジェクト
                                selector:@selector(driveClock:) //呼び出すメソッド
                                userInfo:nil //メソッドに渡すパラメータ
                                repeats:YES]; //繰り返し
```

図3.4 タイマーの設定を行うソースコード

```
-(void)driveClock:(NSTimer *)timer
{
    // 現在の日時を生成
    NSDate *date = [NSDate date];
    // NSDateFormatterを生成
    NSDateFormatter *form = [[NSDateFormatter alloc] init];
    // 日時の形式を細かく指定 年年年年-月月-日日 曜曜 時時:分分:秒秒.ミリ秒ミリ秒ミリ秒
    [form setDateFormat:@"yyyy-MM-dd HH:mm:ss.SSS"];
    // 書式付き文字列を生成
    NSString *str = [form stringFromDate: date];
    NSString *str2 = [[NSString alloc] initWithFormat:@"%s\n", str];
    NSLog(@"%@", str2);
    [self writeLog: str2 fileName:@"timeline.txt"]; // timeline.txtに書き込み、なければ生成も
    // formインスタンスを破棄
    [form release];
}
```

図3.5 「driveClock」ソースコード

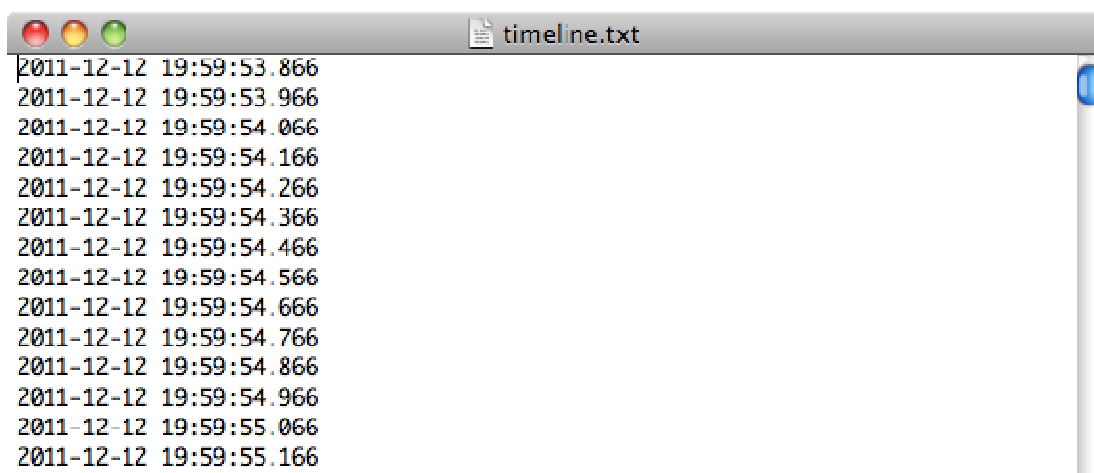


図3.6 現在時間を書き込んだテキストのスクリーンショット

3.2.2 デバイスの向きを取得

iOS では UIDevice クラスを利用することでデバイス(iPhone)のさまざまな情報を取得・設定することができる^{20) 21)}。UIDevice クラスのプロパティである `orientation` を参照することでデバイスの向きを取得でき、今回実装したプログラム「orientationGet」を図3.7に示す。デバイスの向きに応じて `orientation` には0から6の整数が代入されるようになっており、その対応表を表3.1に示す。

```

-(void)orientationGet{
    //NSLog(@"orientationGet");
    // 現在の向き取得
    UIInterfaceOrientation orientation = [[UIDevice currentDevice] orientation];
    NSLog(@"%d", orientation);
    if(orientation == 0){
        NSLog(@"向き不明");
    }else if(orientation == 1){
        NSLog(@"縦HOMEボタン下");
    }else if(orientation == 2){
        NSLog(@"縦HOMEボタン上");
    }else if(orientation == 3){
        NSLog(@"横HOMEボタン右");
    }else if(orientation == 4){
        NSLog(@"横HOMEボタン左");
    }else if(orientation == 5){
        NSLog(@"画面上置き");
    }else if(orientation == 6){
        NSLog(@"画面下置き");
    }
}

```

図3.7 デバイスの向きを検出するプログラム「orientationGet」

表3.1 デバイスの向き対応表

Orientation	デバイスの向き
0	向き不明
1	縦, HOME ボタン下
2	縦, HOME ボタン上
3	横, HOME ボタン右
4	横, HOME ボタン左
5	画面上置き
6	画面下置き

3.3 姿勢判別実験

3.3節では重力加速度で姿勢を判別する実験を行う．具体的には「立った状態」,「座った状態」,「歩いている状態」の3つの動作中に画面を見続けてもらい,各動作における被験者の重力加速度に注目する．そこで各動作における統計的な傾向を観察する．3.2節のローパスフィルタ処理を行い,重力加速度に式(2)を用いて, G_{Mag} として評価した．

3.3.1 姿勢判別実験：プロトコル

今回,男性の被験者を9名用意し,その内3名には同じ実験を続けて3回実施した．被験者が座るために椅子を用意し,歩行用に直線約30メートルの廊下を用意した．

事前に iPhone3GS 本体に実験用に開発したアプリをインストールしておく．今回のアプリでは映像は再生されず,次に被験者が行う実験手順が被験者に指示される(図3.8～図3.12)．フェーズ1を立位,フェーズ2を座位,フェーズ3を歩行中,フェーズ4をフリー歩行としている．「次へ」ボタンを押して実験を開始すると,表示される数字が0から1ずつ,1秒毎に増加していく．さらにボタンを押すことで次の画面に遷移し,再び数字が0から1ずつ増加する．ボタンを押すタイミングは強制ではないが,各動作で15秒は画面を見てもらうように指示される．画面を集中して見るように説明し,これからどんな実験を行うかを先に説明する．歩く時は廊下をまっすぐ歩くように指示し,曲がったり引き返したりしないように指示した．

また,歩行が終了し,実験開始位置まで戻った後に設問に口頭で答えてもらった．設問は「(1)iPhone を持っている時と持っていない時で違いはあるか, (2)集中して見れたか, (3)スマートフォンは普段使うか, (4)普段歩きながら携帯電話を使用するか」の4問である．2回実験を実施した被験者には「(5)最初の実験と次の実験で,自分の中で違ったことはあるか」と設問し,3回実験を実施した被験者には「(6)3回実施して,自分の中で違ったことはあるか」と設問した．ビデオによる記録を並行して行った．

3.2.1節と3.2.2節の内容を踏まえ,今回データとして取得した．テキストとして保存する情報は「現在時間」,「デバイスの向き」,「重力加速度 g_x, g_y, g_z 」,『「次へ」ボタンを押した瞬間』の4つである．実際に取得したデータを保存したテキストを図3.13に示す．



図3.8 実験用アプリ：実験開始時とその画面



図3.9 実験用アプリ：フェーズ1とその画面

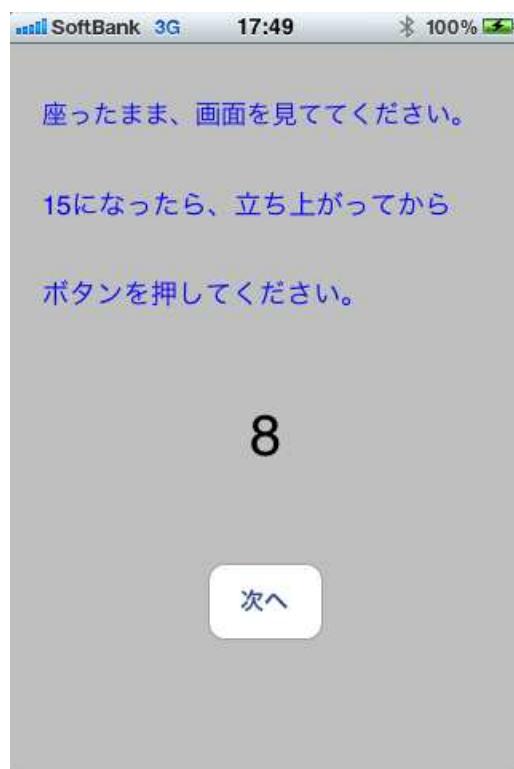


図3.10 実験用アプリ：フェーズ2とその画面

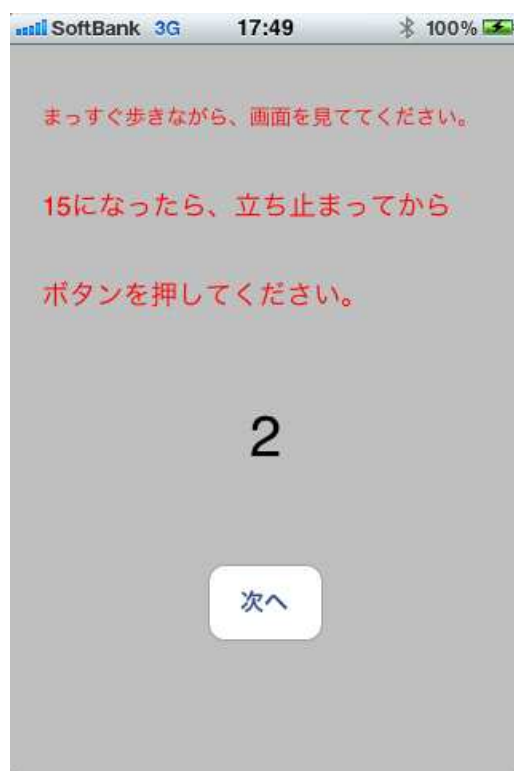


図3.11 実験用アプリ：フェーズ3とその画面



図3.12 実験用アプリ：フェーズ4とその画面

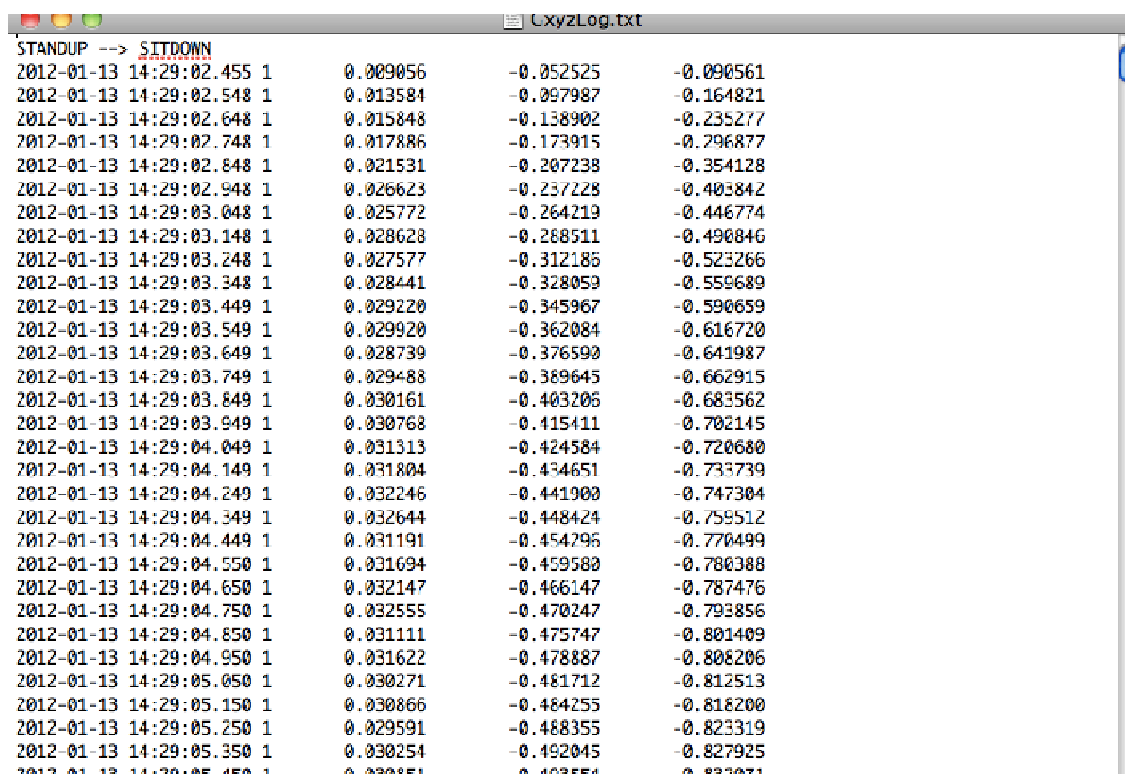


図3.13 実験用アプリで保存されたテキスト

3.3.2 姿勢判別実験：言語的アンケートによる結果

3.3.1節の実験後のアンケート結果を表3.2に示す．設問1において，違和感を感じたと回答した5名について，ヒアリングでは「iPhoneに集中しているから，少し周りに鈍感になった気がする」「画面を見ていると前が見えないので，歩きがゆっくりになる」「歩いている時に周りを見てしまう」「指を動かさないのでこわばる感じ」「二の腕あたりが疲れた」と述べている．設問2において，集中できないと回答した4名は「歩いている時は周りを確認するから集中できない」と述べている．設問4において「時々使う」と回答した3名については「バスや電車でジッとしている時ぐらいしか使わない」「何か物事が終わった後に使う」と述べている．「使わない」と回答した3名からは「スマートフォンの片手操作はできない」という意見が得られた．

アンケートの結果，歩行時は周りを見ながら歩くため，画面に集中しにくいことがわかった．

表3.2 アンケート結果(全9名)

設問 1	違和感は何にもない	4 名
	違和感を感じた	5 名
設問 2	集中できた	5 名
	集中できない	4 名
設問 3	使う	5 名
	使わない	4 名
設問 4	よく使う	3 名
	時々使う	3 名
	使わない	3 名
設問 5	スムーズに次の動作ができた	1 名
	すでにやることを知っているから，画面には集中できなかった	1 名
	立っているときに，左手をポケットに添えていた	1 名
設問 6	座る時の iPhone の向きが違ったかも	1 名
	iPhone を左右どちらの手で持つか，迷った	1 名
	ボタンを押す秒数が近付くと，気持ちがソワソワする	1 名

3.3.3 姿勢判別実験：加速度による評価

3.3.1節の実験の結果から，動作中の各被験者の重力加速度を解析した．実験中の被験者の重力加速度 g_x, g_y, g_z の動きをそのまま表示した結果を図3.14に示す．右方向に時間 milli-second(1000 分の1 秒)，上下方向は重力加速度の値を示す．

立っている状態と座っている状態の差異はほとんど見られないが，椅子に座り込む動きや椅子から立ち上がる動きはグラフから読み取ることができる．また，歩いている状態の時には重力加速度の振幅が小刻みに増減する傾向が読み取れ，静止状態とは違うことがわかる．60000ms 付近から波形が大きく反応しているのは，実験が終了し，戻ってくる被験者が iPhone から目を離し，手を振って歩行していることがビデオ観察から確認できた．

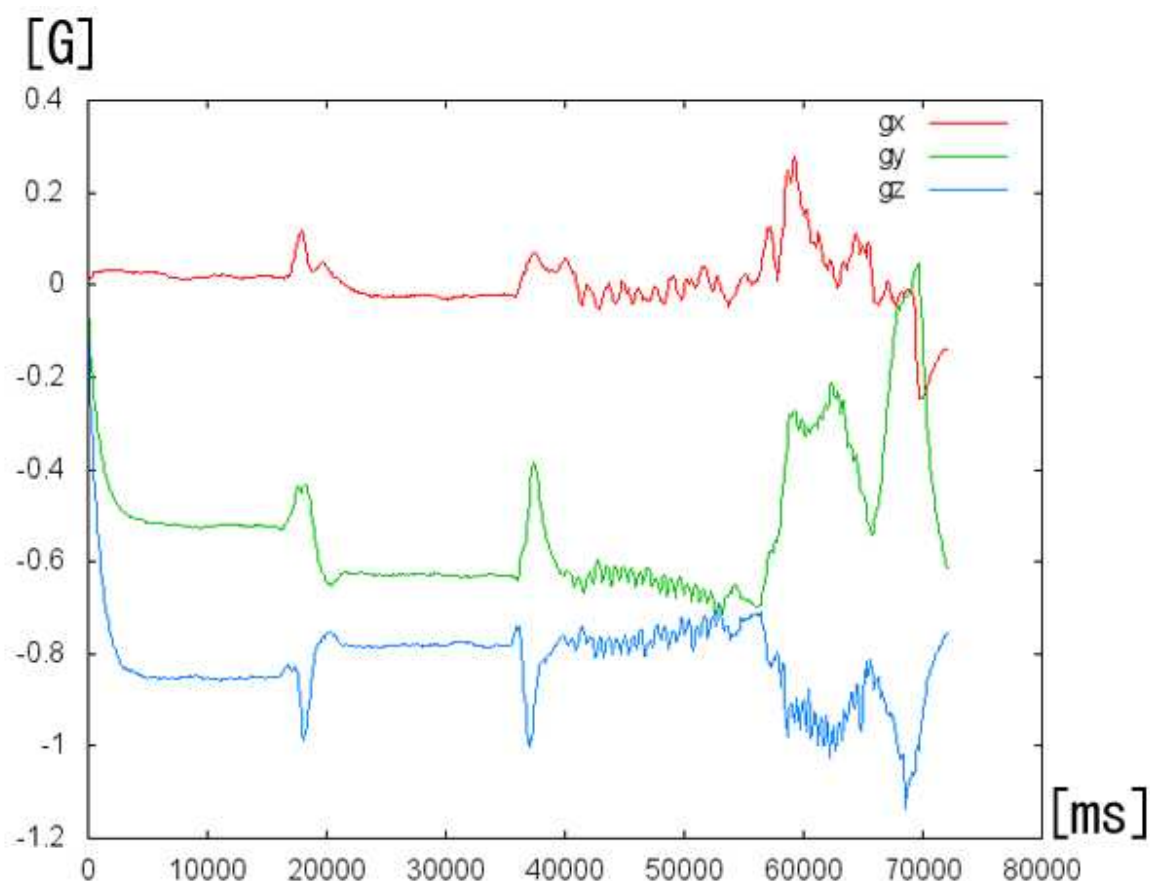


図3.14 重力加速度 g_x, g_y, g_z

次に式(1)を用い，被験者の重力加速度を G_{Mag} として加速度表現した結果を図3.15に示す．右方向に時間 milli-second(1000 分の1 秒)，上下方向は G_{Mag} の値を示す．

実験開始直後の A では重力加速度の値が入力され始め，B は立位での視聴，C で座り込み，D では座位での視聴，E では椅子から立ち上がり，F で歩行しながら視聴，G で立ち止まり，H では画面を見ずに自由に歩いていることが読み取れる． g_x, g_y, g_z として見るよりも，座り込む動作や立ち上がる動作，歩行状態から静止状態になる瞬間のイベントがマグニチュード表現により読み取りやすくなった．

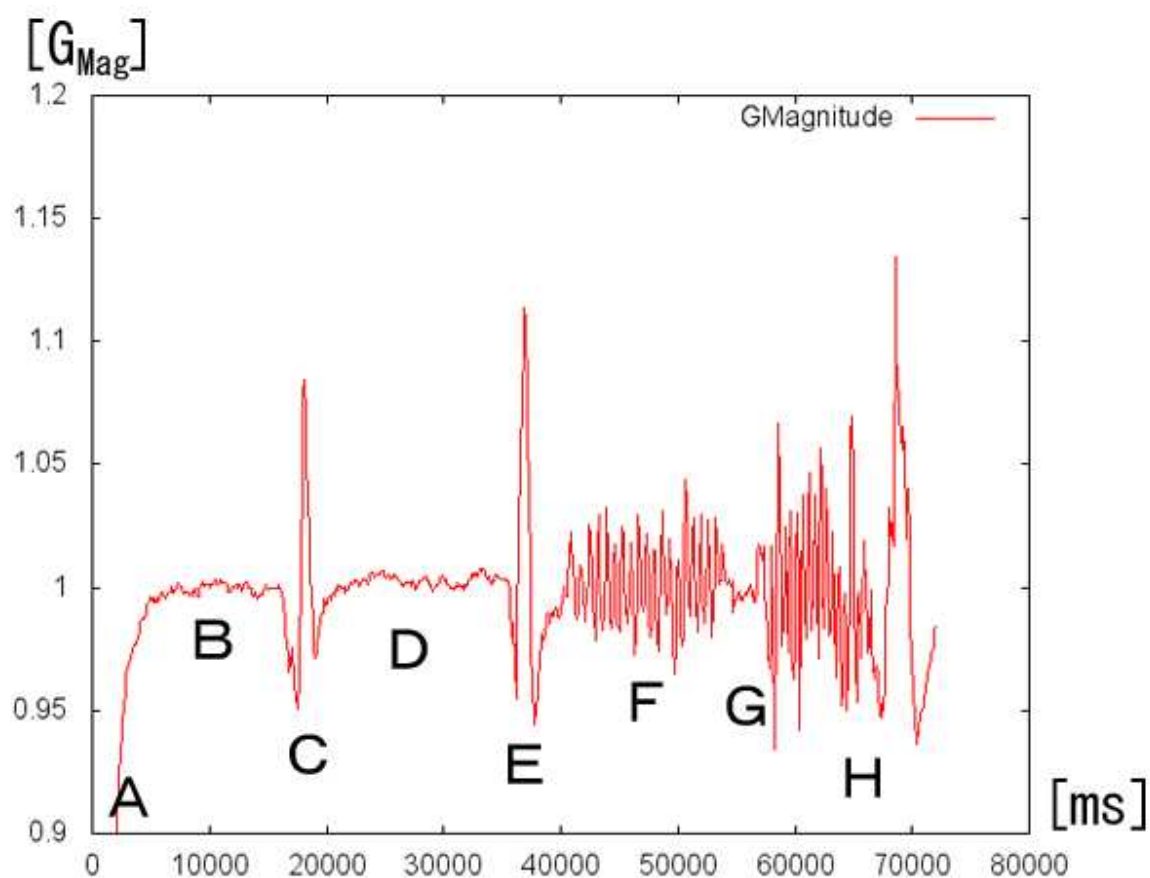


図3.15 重力加速度($G_{Magnitude}$ 表現)

次にデバイスの向きに応じて0から6の整数が格納される iOSAPI の `orientation` を，グラフで表現した結果を図3.16に示す．グラフからは被験者が iPhone を縦に持っていること，立ち上がる時や座る時には iPhone が画面上向きになることが読み取れる．

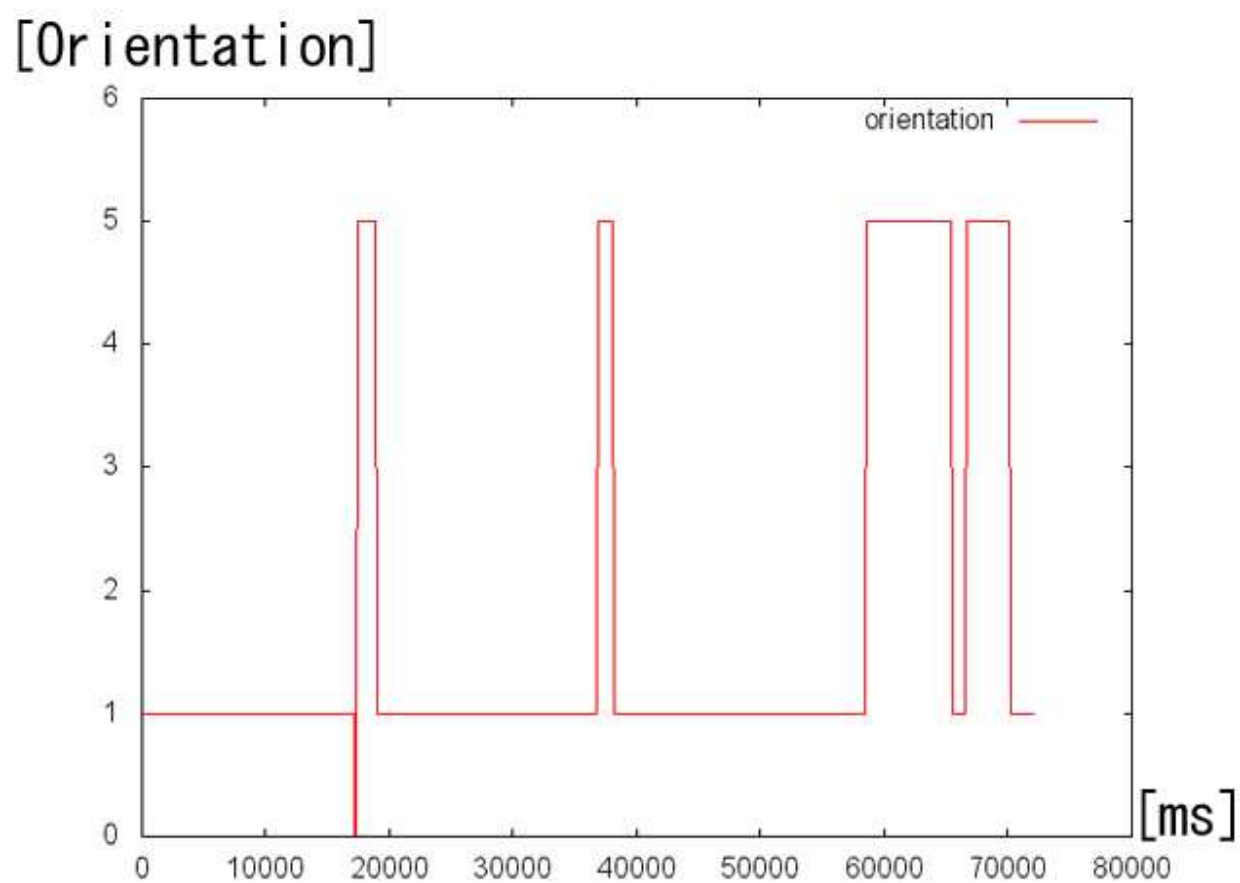


図3.16 Orientation の推移

ここで重力加速度の G_{Mag} と, Orientation の二つを合わせた図を図3.17に示す.

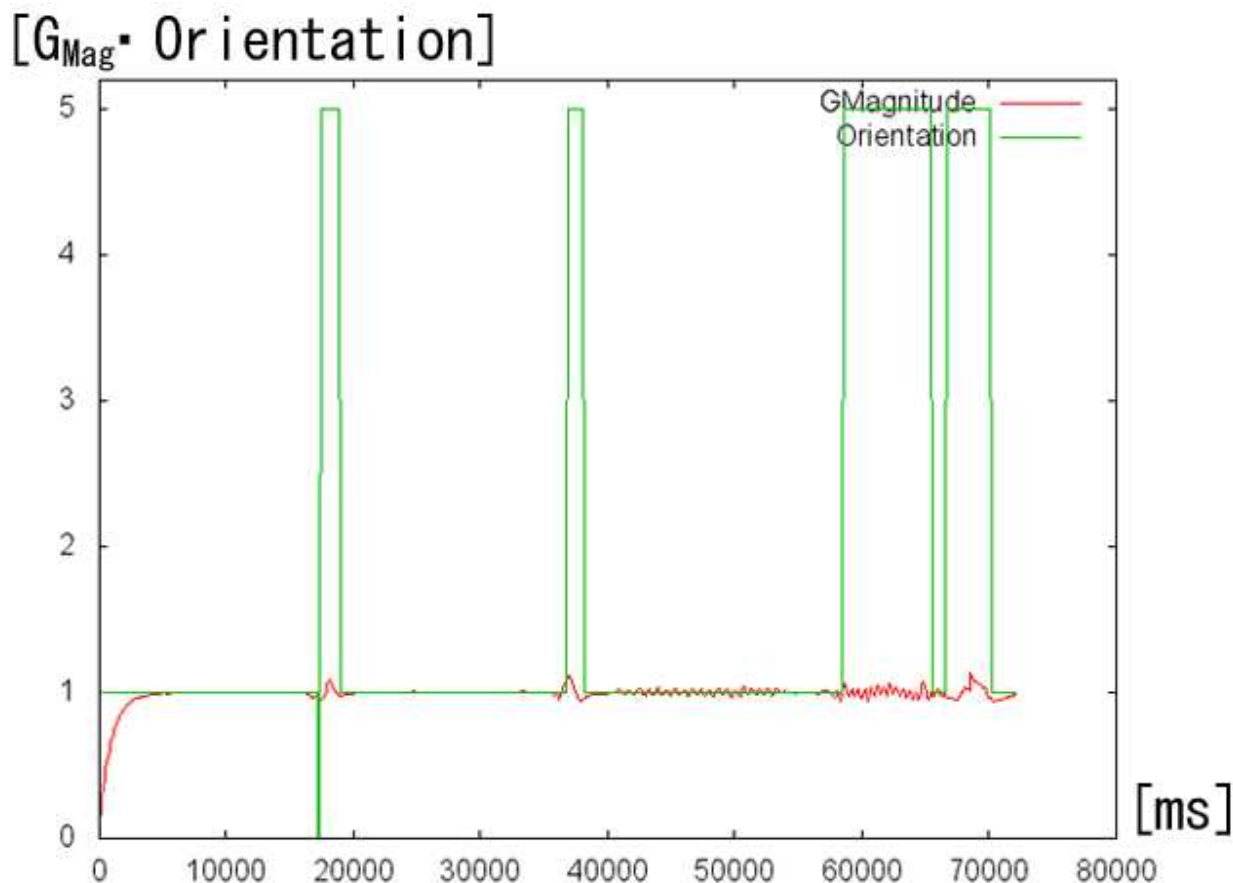


図3.17 G_{Mag} と Orientation の推移

グラフからは被験者が iPhone を縦に持っていることがわかる (Orientation=1). 座り込む時や立ち上がった時, フリーで歩き始めた時には Orientation が5を示しており, 立ち上がる時や座る時には iPhone が画面上向きになることが読み取れる. 座り込んだ時に一時, デバイスの向きが不明になっている (Orientation=0).

以上のことから, iPhone を持った状態において重力加速度を用いることで, ユーザの姿勢が判別できる可能性があることがわかった. デバイスの向きを得るだけであれば Orientation を活用して簡潔に判別できるが, ユーザの姿勢まで判別するのであれば重力加速度を解析したい.

3.4 抽象的な動画を用いた実験

抽象的な動画を用いた実験には「Sustanime 早川貴泰」iPhone アプリを用いた。アプリの画面を図3.18に示す。このアプリは「Sustanime プロジェクト」によって開発されたプロトタイプで、抽象的なアニメーション作家早川貴泰氏の映像作品が読み込み時間等を気にせず、快適に高精細な映像作品を視聴できるアプリである。しかしながら、作品の先進性から、レビューやアンケートでは作品の向上につながるフィードバックは得られない。そこで任意の被験者における統計的な傾向を観察し、その後、提案システムにおける評価実験を行う。

被験者の視聴する映像作品には「Sustanime 早川貴泰」 iPhone アプリ実験用バージョンから選択できる「雲散霧消」と「阿吽二字」の2作品を用いた。この2作品は図3.19、図3.20のような抽象的なアニメーション作品であり、かつ、再生時間は前者が3分11秒、後者が3分10秒と近似しているため、本実験における時間的な差異は見られないとし、採用した。

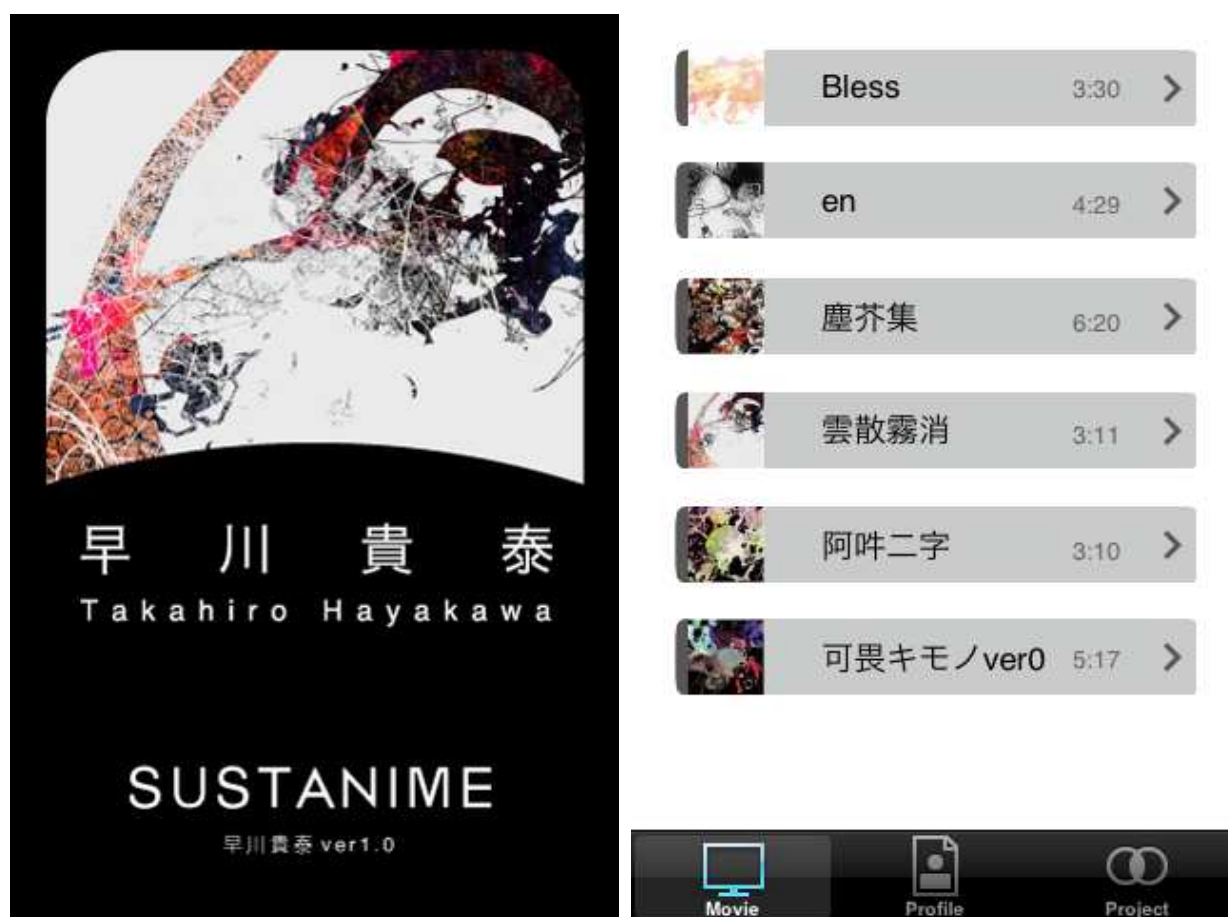


図3.18 iPhone「Sustanime 早川貴泰アプリ」での画面

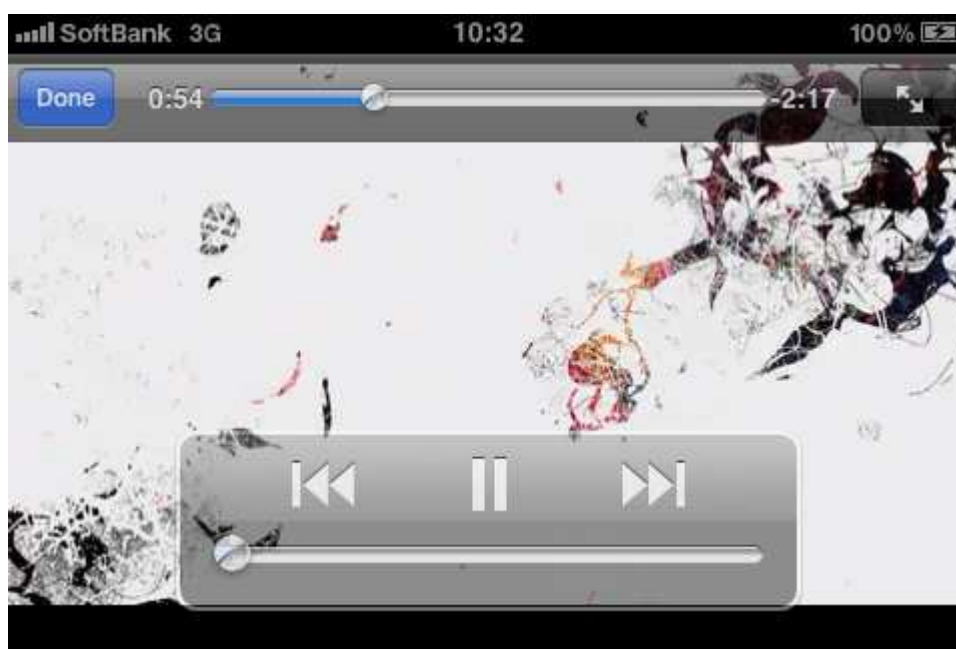


図3.19 Sustanime 早川貴泰アプリ 雲散霧消

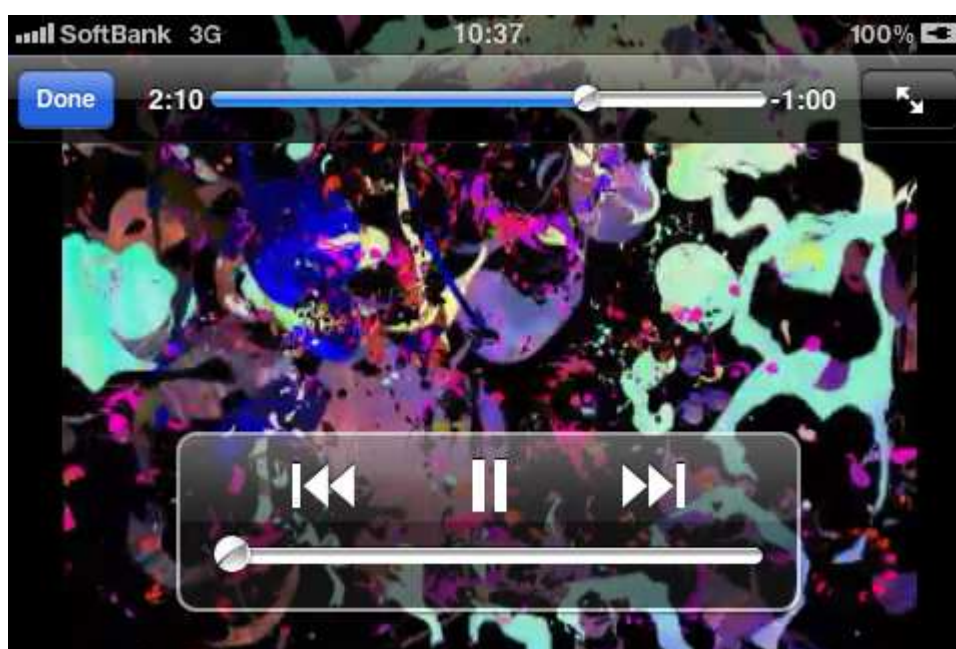


図3.20 Sustanime 早川貴泰アプリ 阿吽二字

3.4.1 抽象的な動画を用いた実験：実験プロトコル

事前に iPhone3GS 本体には「Sustanime 早川貴泰」アプリをインストールしておく。メニュー画面では被験者には、次に視聴してもらう映像作品を選択するように説明している。

今回、男性の被験者を13名、女性3名を用意した。それぞれのグループ構成については表3.3に示す。グループ A と C には「雲散霧消」の視聴後に「阿吽二字」を、グループ B

と D には「阿吽二字」の視聴後に「雲散霧消」の視聴を実施した(図3.21, 図3.22). 今回の実験では無意識下での反応を計測するため, 映像に集中してもらうよう説明した. 1本目の映像が終了した際には, 一度終了したと報告してもらった後に, 次に視聴する映像を指定した.

また, 映像の視聴が終わった後に, 4つの設問に口頭で答えてもらった. 設問は「(1)どちらの映像が好きか, (2)どちらの映像が面白いか, (3)途中で飽きたか, (4)集中して視聴できたか」, の4問である. ビデオによる記録を並行して行った.

表3.3 グループ構成

グループ	男性	女性
A	3 名	1 名
B	2 名	2 名
C	4 名	0 名
D	4 名	0 名

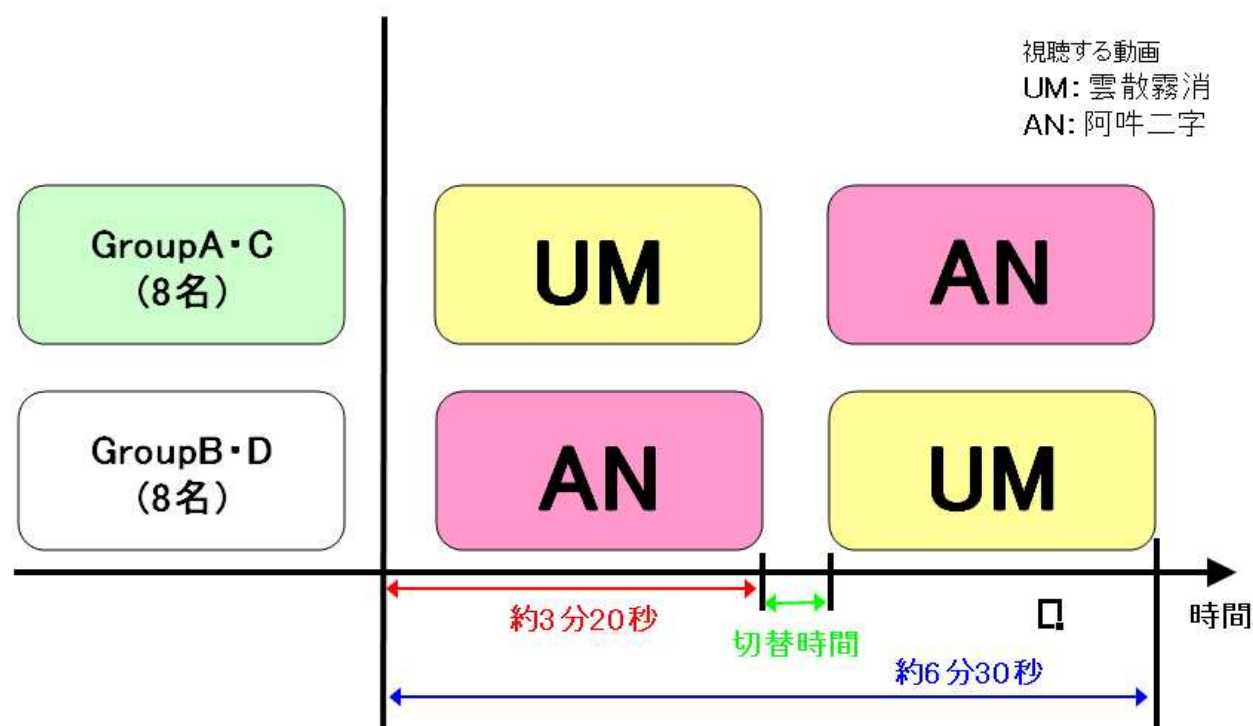


図3.21 実験プロトコル：動画視聴順



図3.22 実験：立姿勢における視聴状態

3.4.2 加速度計測実験：言語的アンケートによる結果

前節の実験結果について述べる．設問の回答や，映像視聴時における加速度についても解析した．実験後の言語によるアンケート結果を表3.4に示す．

被験者16名中7名が好きな映像と面白いと思う映像に差異が見られ，ヒアリングでは『阿吽二字』に使われている色数が少なく，目にやさしくて好きだが，面白いと思うのは『雲散霧消』と述べられている．

設問3において，「飽きた」と回答した4名は『雲散霧消』を視聴中に『飽きた』，「映像のどこに注目して見れば良いのかわからない」と述べている．

設問4において，「集中できなかった」と回答した4名についてヒアリングでは「立っているので，動いてはいけない気がした」，「映像冒頭が集中しにくかった」「映像のどの部分に集中すれば良いのかわからなかった」「カメラがあると意識してしまう」と述べている．アンケートの結果から，カメラが被験者を意識させてしまうことがわかった．

表3.4 言語によるアンケート結果(全16名)

設問 1	雲散霧消が好き	5 名
	阿吽二字が好き	11 名
設問 2	雲散霧消が面白い	8 名
	阿吽二字が面白い	8 名
設問 3	飽きた	4 名
	飽きない	12 名
設問 4	集中できた	12 名
	集中できなかった	4 名

3.4.3 加速度計測実験：加速度による評価

3.4節の実験の結果から映像視聴中の各被験者の瞬間的な加速度(a_x, a_y, a_z)を解析し、被験者の動きを A_{Mag} として表現した結果を図3.23、図3.24に示す。右方向に時間[s]、上下方向は式(2)によって得られる A_{Mag} の値を示す。グラフ上限を0.2G としており、高精度加速度センサならではの分解能によるデータであるといえる。ビデオでは確認できないほどの微小な動きが記録されていることがわかる。

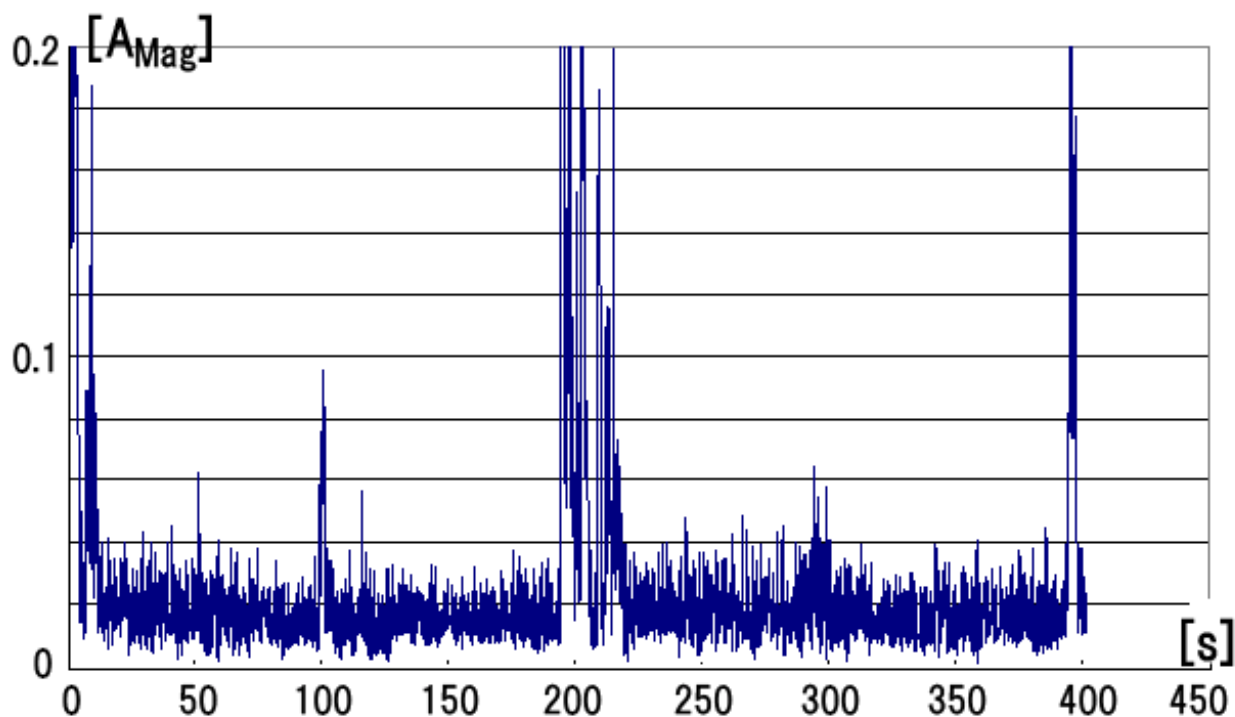


図3.23 「雲散霧消」→「阿吽二字」を視聴した被験者 A

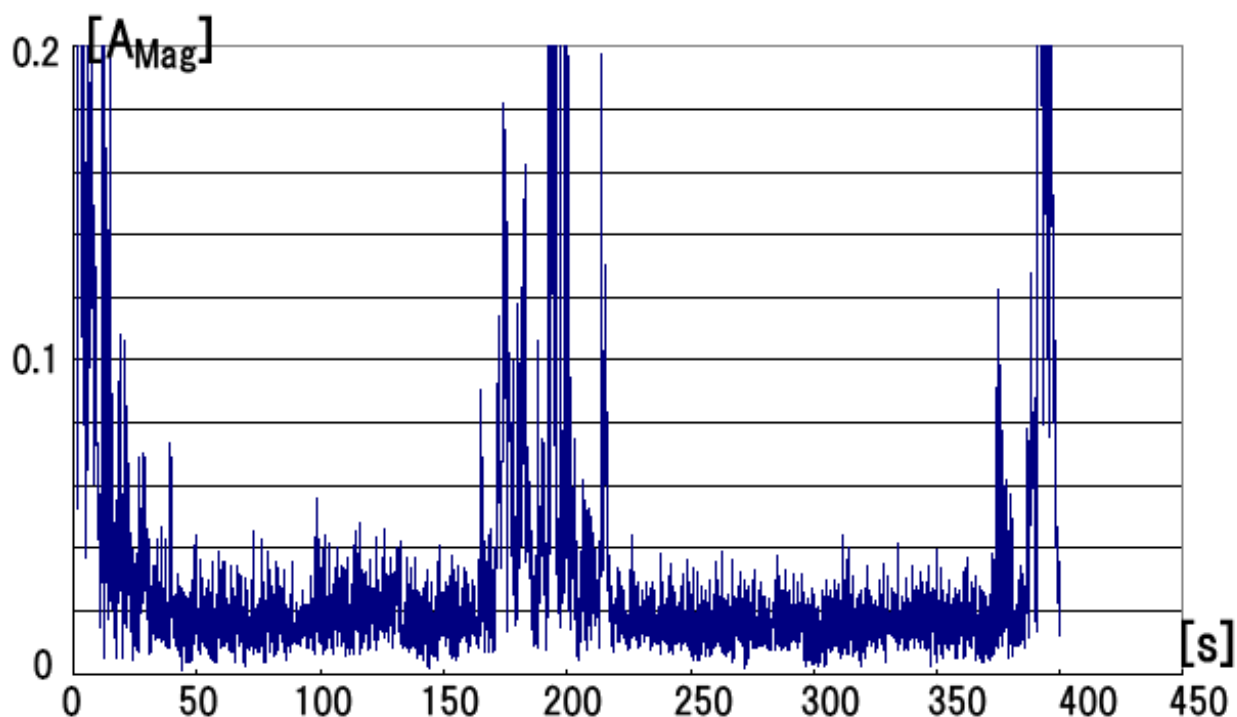


図3.24 「阿吽二字」→「雲散霧消」を視聴した被験者 B

図3.23, 図3.24は共に200秒付近の A_{Mag} の振幅が高まっていることが読み取れる．これは1本目の映像が終了し，2本目の映像視聴に以降する際の A_{Mag} である．映像が終了し，iPhone から目を離し，動いていることがビデオ観察から確認できた．

ここで3.4.1節におけるグループ A の各被験者の平均としたものを図3.25に示す． A_{Mag} の振幅が高い部分ほど，多くの被験者に反応が見られた箇所といえる．

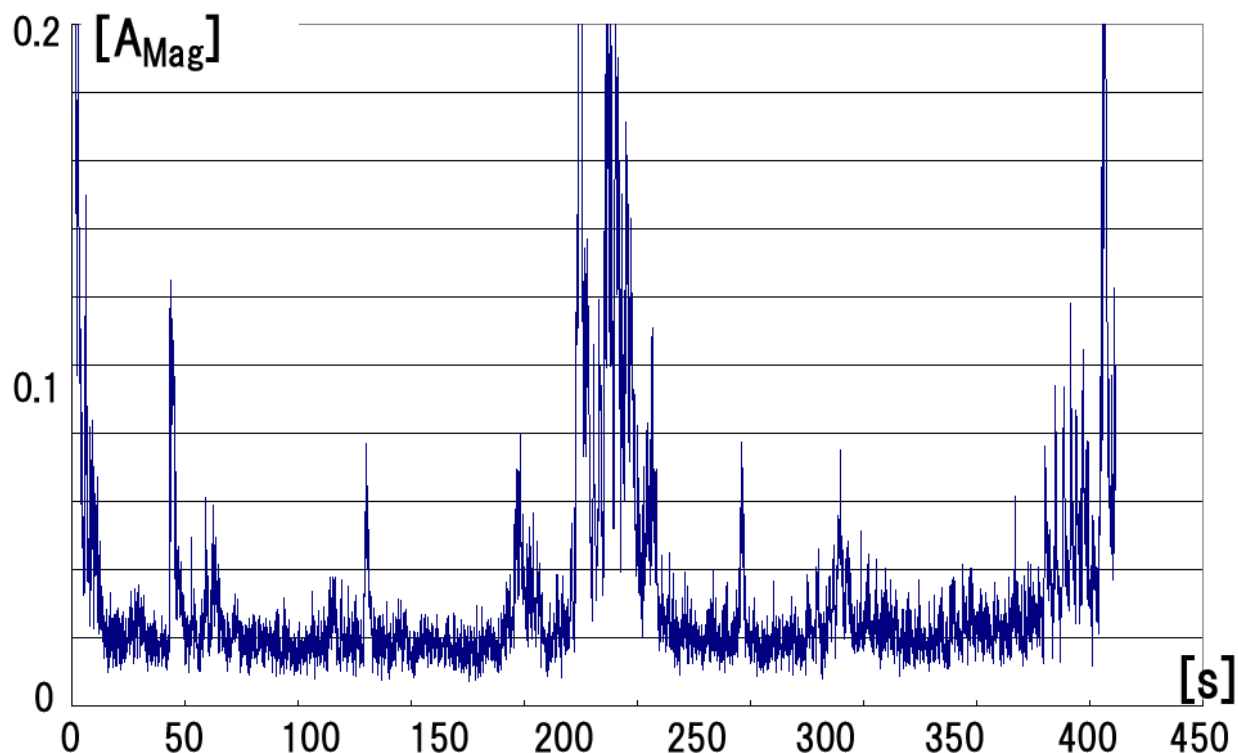


図3.25 グループ A 全被験者の加速度の経過(平均)

すると「雲散霧消」視聴中，映像再生時間約57秒，1分52秒，2分 51秒の3箇所で被験者の瞬間的な加速度が高まっている傾向が見られた．

「雲散霧消」の作者である早川貴泰氏にこのタイムコードが制作上どのような意味を持つかヒヤリングしたところ，「3箇所とも設計上において狙ったハイライトシーンであり，映像作品として盛り上がるように，より印象的なグラフィックを心がけて制作している．音楽も同様である．私の作品は抽象的な作品ではあるが，印象的なグラフィックとするために，具象をほんの少し見せるパートでもある」との反応を得た．

このことから，映像視聴において加速度を用いることで，作家の意図する部分で何らかの反応が見られる可能性があることがわかった．被験者が細かく恣意的に動くことができないことは図3.14で明確であり，重力加速度と瞬間的な加速度を組み合わせることでより精度の高い解析結果が得られるだろう．

4. エンタテインメントシステムへの応用

加速度センサを用いたユーザ解析と非言語評価手法についてまとめたものを図4.1に示す。3章の実験結果から、スマートフォンを用いることでユーザの意識的な反応から無意識的な反応まで取得することができる。重力加速度や瞬間的な加速度を解析することで映像への反応や姿勢の変化が得られ、ボタンイベントや各種センサを解析することで意識的な反応が取得できる。

今後は現在時間など、どんなアプリでも必要になる情報を共通フォーマットとして用意し、各アプリケーションに応じて情報を取得していくことで、より高い質の情報を作家へ還元できるだろう。

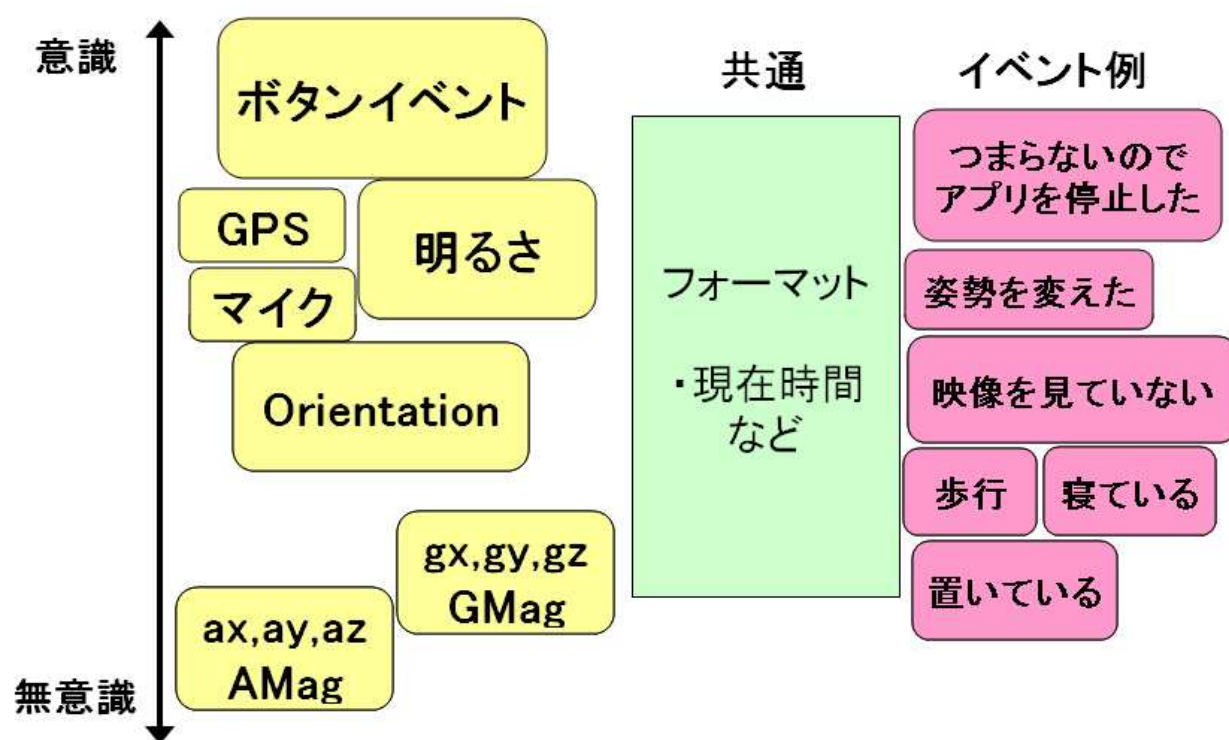


図4.1 ユーザ解析と非言語評価手法のまとめ

4.1 ユーザから新たに取得する情報

スマートフォンアプリを用いている利点を活かし、随時アプリに機能追加することでユーザ解析の精度を高めていくことができる。3章の実験後、取得できる視聴情報を追加した。具体的には「デバイス情報」、「バッテリー情報」、「HOME ボタン検出」、「起動回数」、「映像再生をやめた瞬間」の取得が可能になった。

4.1.1 デバイス情報を取得

3.2.1項と同様，UIDevice クラスを利用してデバイスの ID，名前，モデル，OS のバージョン等を取得することができる．今回実装したプログラム「deviceGet」を図4.2に示す．デバイス情報を取得するためにそれぞれ参照している UIDevice クラスのプロパティと，取得できる情報の対応表を表4.1に示す．実際にプログラムを実行した時のコンソール画面を図4.3に示す．

```
// デバイス情報の取得
-(void)deviceGet{
    NSLog(@"deviceGet");
    // UIDeviceクラスのインスタンスを取得する
    dev = [UIDevice currentDevice];
    dev_name = dev.name;
    NSLog(@"デバイス名: %@", dev_name);
    dev_model = dev.model;
    NSLog(@"デバイスモデル: %@", dev_model);
    dev_localizationModel = dev.localizedModel;
    NSLog(@"ローカライズバージョン: %@", dev_localizationModel);
    dev_systemName = dev.systemName;
    NSLog(@"OS名称: %@", dev_systemName);
    dev_systemVersion = dev.systemVersion;
    NSLog(@"OSバージョン: %@", dev_systemVersion);
    dev_uniqueIdentifier = dev.uniqueIdentifier;
    NSLog(@"デバイスID: %@", dev_uniqueIdentifier);
}
```

図4.2 デバイスの情報を取得するプログラム「deviceGet」

表4.1 UIDevice のプロパティと取得できる情報

プロパティ名	取得できる情報
name	デバイスの名前
model	デバイスのモデル
localizedModel	ローカライズのバージョン
systemName	OS の名前
systemVersion	OS のバージョン
uniqueIdentifier	デバイスの ID



図4.3 「deviceGet」を実行したコンソール画面のスクリーンショット

4.1.2 バッテリー情報を取得

3.2.1項と同様，UIDevice クラスを利用してデバイスのバッテリー状態を取得することができる．バッテリー状態を取得するために実装したプログラム「batteryGet」を図4.4に示す．バッテリー状態を取得するためにそれぞれ参照している UIDevice クラスのプロパティと，取得できる情報の対応表を表4.2に示す．実際にプログラムを実行した時のコンソール画面を図4.5に示す．

```

// バッテリー状態の取得
-(void)batteryGet{
    NSLog(@"batteryGet");
    // UIDeviceクラスのインスタンスを取得する
    dev = [UIDevice currentDevice];
    dev.batteryMonitoringEnabled = YES; // YES:バッテリーの残量を監視する(デフォルトはNO)
    NSLog(@"バッテリー残量: %f", dev.batteryLevel);
    if(dev.batteryState == UIDeviceBatteryStateUnknown){
        // バッテリー状態取得不能
        NSLog(@"バッテリー状態取得不能");
    }
    if(dev.batteryState == UIDeviceBatteryStateUnplugged){
        // バッテリー使用中
        NSLog(@"バッテリー使用中");
    }
    if(dev.batteryState == UIDeviceBatteryStateCharging){
        // バッテリー充電中
        NSLog(@"バッテリー充電中");
    }
    if(dev.batteryState == UIDeviceBatteryStateFull){
        // バッテリーフル充電状態
        NSLog(@"バッテリーフル充電中");
    }
}
}

```

図4.4 バッテリーの状態を取得するプログラム「batteryGet」

表4.2 UIDeviceのプロパティと取得できる情報

プロパティ名	取得できる情報
batteryLevel	バッテリー残量. 0.0(0%)から 1.0(100%)の間で取得可
UIDeviceBatteryStateUnkown	バッテリー状態取得不能
UIDeviceBatteryStateUnplugged	バッテリー使用中
UIDeviceBatteryStateCharging	バッテリー充電中
UIDeviceBatteryStateFull	バッテリー100%状態

```
| batteryGet
| バッテリー残量: 0.750000
| バッテリー充電中
```



図4.5 「batteryGet」を実行したコンソール画面のスクリーンショット

4.2 まとめと今後の課題

加速度センサを用いたユーザ解析手法はスマートフォン等を用いるコンテンツであれば産業での実用化・応用可能性はあるだろう。例えば、動画視聴サービスに組み込むことで多くのユーザからフィードバックを得られるだけでなく、コメントやレビューをしない視聴者からの反応も取得でき、映像評価として応用できる。動画視聴時の加速度やデバイスの向きなどに注目することで、映像視聴環境の快適性の評価もできるだろう。映像作家が提案システムの組み込まれた動画視聴サービスに作品を投稿することで、作品クオリティ向上システムとしても活用できる。動画視聴サービスではコマーシャルや宣伝広告が入る場合があるが、宣伝効果の測定も可能だろう。

映像視聴と同様に「見る」ことがユーザの主な動作になる電子書籍をスマートフォンで読む際には、どのページで集中し、飽きているかが測定しやすいため、電子書籍の品質評価もできるだろう。現在、図書館で電子書籍を借りられる例もある²²⁾。大学附属図書館では図書ID、分類記号番号、請求記号、貸出者(仮)ID、所属、身分、貸出年月日時刻、返却年月日時刻など膨大なデータが存在し、それらから統計的解析による全体の傾向に関する知見が得られる。さらに学生の性格や専攻、利用目的等も分析できる²³⁾。学校などの教育環境で電子書籍や映像教材を活用した場合、その教育環境の評価ができるだけでなく、学生がどれだけ授業に集中しているかも評価できる可能性がある。

映像視聴以外にも、新たに開発するデジタルコンテンツのプロトタイプにも応用可能性がある。例えば、加速度センサは任天堂社製Wiiリモコン、SONY社製PlayStation Move モーションコントローラやワイヤレスコントローラ(SIXAXIS)等にも搭載されている²⁴⁾ ²⁵⁾ ²⁶⁾。テストプレイで多くのユーザがプレイする際に加速度を取得することで、ゲームシステム等における体験評価にも応用できるだろう。各ユーザが好むゲームデザインの動的生成にも応用可能性がある。デジタルコンテンツそのものの品質評価だけでなく、将来的には品質評価の困難な次世代エンタテインメントシステムを開発する場合においても役立つだろう。

5. むすび

視聴者から作家に視聴情報を「還元」するという課題において、加速度とデバイスから得られる視聴情報の還元を実現した。提案するシステムでは、iPhone3GS に搭載された加速度センサによって取得した加速度をフィルタリングし、 G_{Mag} 、 A_{Mag} による値に注目することで、意識的な入力を強いることなくユーザの反応を測定できる。自然な状態での測定も可能となった。

立姿勢における2つの映像作品を視聴した際の加速度に加え、実験後の口頭設問による被験者の回答や、作品を制作した作家に直接ヒヤリングすることで、提案手法の尤もらしさを確認した。作家が意図して作りこんだハイライトシーンに、より顕著な反応を示すことがわかった。また、デバイスから現在時間、デバイスの向きを取得することでユーザの視聴状態を推定することも可能になった。結果についてまとめたものを図5.1に示す。

今後は映像視聴だけでなく、スマートフォンのゲームアプリを遊んでいる時の加速度にも注目したい。測定に扱うデジタルコンテンツの種類を増やしていくことで知見を深め、より具体的な情報を作家へ還元したい。コンテンツに関わらず、より自然に、よりカジュアルにデータ取得をすることで被験者の負担を減らし、大量のデータを得られることが期待できるだろう。カメラもアプリに組み込むことで、より自然に、倫理的に配慮した実験も可能になるだろう。また、アプリを用いている利点を活かし、コンテンツに対してユーザの興味が沸いた瞬間に、ユーザ自身から入力を取得することで、波形の解析精度をより高めることができると期待される。今後も継続して研究を続け、動的にユーザの状態を解析することで必ずユーザが満足するようなコンテンツの開発を可能にしたい。

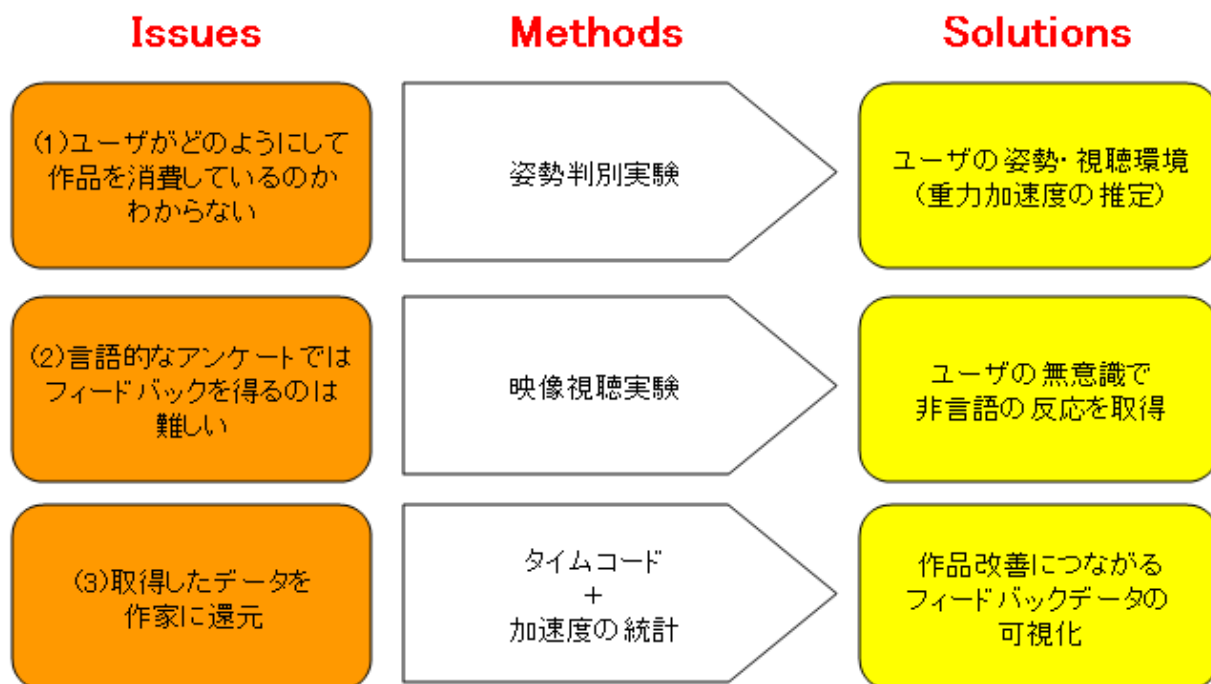


図5.1 本研究の課題と方法，解決方法

謝辞

この研究は平成22年度文化庁メディア芸術人材育成支援事業によって支援された。また神奈川工科大学情報学部情報メディア学科各位には実験にご協力いただいた。ここに謝意を表したい。

日々、夜遅くまで多大なる御指導と激励を頂戴した白井暁彦准教授に心より感謝いたします。6年間、親身にご指導いただいた服部元史教授に深く感謝いたします。制作の基礎となる技術を教えてくださいました深野暁雄先生に深く感謝いたします。貴重な体験の場を提供してくださり、何度も映像提供していただいた早川貴泰氏に深く御礼申し上げます。Sustanime プロジェクトのメンバーとして声をかけていただき、アプリのプログラミングだけでなく、多大なご指導いただいた田中健司氏にこの場を借りて深く感謝いたします。プログラミングにおいてご指導くださいました尾形薫先生に深く感謝いたします。美術・デザイン向上のために貴重な御指導を下されました中村泰清先生に深く御礼申し上げます。本研究に関して貴重な御意見をくださいました服部哲准教授に深謝いたします。日頃から切磋琢磨し、叱咤激励し合った山下泰介君、山本倫之君に心より感謝いたします。そして白井研究室一同にこの場を借りて御礼申し上げます。

参考文献

- 1) 4D2A・ワークスコーポレーション書籍編集部, 岡本淳: クリエイティブ業界に就職するためのポートフォリオの実例集, pp.34-109, 株式会社ワークスコーポレーション (2011).
- 2) 尾形美幸: クリエイティブ業界を目指す人のためのポートフォリオ見本帳, pp.18-89, 株式会社エムディエヌコーポレーション (2011).
- 3) アニメーターweb 学校法人格アニメ学校一覧2008,
<http://www7a.biglobe.ne.jp/animation/pdf/schools/gakkouhoujinnkaku2008.pdf>
- 4) CALF, <http://calf.jp/>
- 5) アニメーターweb 新人アニメーターの月収を正確に知っていますか?,
<http://www7a.biglobe.ne.jp/animation/pdf/sinjin-animator.pdf>
- 6) アニメーターweb 日本のアニメーション制作現場の実情と課題,
<http://www7a.biglobe.ne.jp/animation/pdf/sitauke.pdf>
- 7) YouTube, <http://www.youtube.com/>
- 8) ニコニコ動画, <http://www.nicovideo.jp/>
- 9) vimeo, <http://vimeo.com/>
- 10) SUSTANIME ホームページ, <http://www.sustani.me/>
- 11) 加藤匠, 白井暁彦, 田中健司, 早川貴泰, 服部元史: 抽象的なアニメーション作品視聴に対する加速度センサを用いた自然なユーザ解析手法の提案, IT を活用した教育シンポジウム講演論文集, Vol.5, pp.53-56 (2011.03).
- 12) 服部祐一, 中村優斗, 平川剛, 井上創造: 動画像と加速度データを用いた行動情報共有システム, インタラクション2011論文集, pp.475-478 (2011.03).
- 13) 田淵勝宏, 納谷太, 大村廉, 野間春生, 小暮潔, 岸野文郎: 無線加速度センサを用いた人の日常行動識別におけるデータ収集条件の影響評価, TVRSJ, Vol.12, No.1, pp.25-36 (2007).
- 14) NHK 放送技術研究所 テレビ視聴インターフェース UTAN の提案,
<http://www.nhk.or.jp/strl/open2011/tenji/10.html>
- 15) 加藤匠, 横田真明, 山下泰介, 服部元史, 白井暁彦: 摂動応答と重心動揺計を用いた嗜好画像のリアルタイム推定手法の提案, 第15回日本バーチャルリアリティ学会大会論文集, pp.234-237 (2010.010).
- 16) ST Microelectronics 社 LIS331DL_DataSheet,
http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/DATASHEET/CD00172345.pdf
- 17) 藤枝崇史: そのまま使える iPhone ゲームプログラム, pp.215-226, 株式会社 毎日コミュニケーションズ (2010).
- 18) 鶴薗賢吾: 基礎からの iPhone SDK 改訂版, pp.140-157, ソフトバンククリエイティブ株式会社 (2009).

- 19) 林晃：Objective-C 逆引きハンドブック，pp.436-468，シーアンドアール研究所 (2010).
- 20) iPhone アプリ開発の虎の巻 UIDevice,
<http://iphone-tora.sakura.ne.jp/uidevice.html>
- 21) 福井高専 IT 研究会 OfficialWiki UIDevice,
<http://profo.jp/wiki/index.php?UIDevice>
- 22) 千代田 Web 図書館,
<http://weblibrary-chiyoda.com/>
- 23) 南俊朗：利用者満足度アップを目指す図書館マーケティング - データ解析による図書館サービス進化への期待 -, 情報の科学と技術, 60巻, 6号, pp.242-248 (2010.04).
- 24) 白井暁彦・小坂崇之・くるくる研究室・木村秀敬：WiiRemote プログラミング, pp.1-7, オーム社(2009).
- 25) 任天堂ホームページ,
<http://www.nintendo.co.jp/index.html>
- 26) プレイステーションオフィシャルサイト PlayStation®Move モーションコントローラ, <http://www.jp.playstation.com/ps3/peripheral/cechzcm1j.html>