

WebSocket を用いたスマートフォン上での エンタテインメントコンテンツ閲覧時のリアルタイム行動分析

北田 大樹^{1,a)} 白井 暁彦^{1,b)}

概要: スマートフォン上でのエンタテインメントコンテンツ閲覧時のリアルタイム行動分析手法について報告する。「楽しみ」は主観的な人間の感情であるが、「笑い」は視聴者の不随意運動として、スマートフォンの高分解能加速度センサにて得ることが可能である。スマートフォン上の映像視聴時に起きる自然な不随意運動を、スマートフォン上の高分解能加速度センサを用いて取得し、視聴者の主観的な情動を統計的かつ集合的に分類取得するアルゴリズムおよびアプリケーション開発を行っている。実験システムは、データ収集のために Ruby と Sinatra フレームワークによって開発されており、将来的な実装として、WebSocket のサーバーサイド JavaScript での実装である Node.js を用いて、Web ベースの分散非同期接続での実装可能性を検討し、LAN/WAN 環境においてベンチマークを行った。推定アルゴリズムは、ムービープレイヤーアプリケーション「L-PoD」に実装され、同一のエンタテインメント・コンテンツにおける被験者間の笑いの特性の違いを Property of Difference (PoD) として視覚化することができる。

Real-time behavior analysis method using WebSocket for entertainment content watching on smartphones

KITADA TAIKI^{1,a)} SHIRAI AKIHIKO^{1,b)}

Abstract: This article contributes to report a method to classify human laughing during video watching using Smartphone. Fun is a subjective human emotion but Laughing is possible to obtain as audience's involuntary movement. The algorithm have been implemented into a movie player application "L-PoD" and it can visualize subjects' Property of Difference (PoD) of laughing part during a same comedy content. For the data collection, the experimental system has been developed by Ruby and Sinatra framework and it has been compared with WebSocket implementation in server side JavaScript "Node.js", as a further implementation.

1. はじめに

Youtube やニコニコ動画といった動画共有サイトは、バーチャルに時間や空間といった制約を超えて、動画コンテンツを共有する感覚を味わえる現代的エンタテインメント・プラットフォームである [1]。また、初音ミクや MikuMikuDance といったユーザー生成コンテンツの一般化や、Oculus Rift などの家庭用 HMD の一般化により、今後より多様な感情や表現力を支えるコンテンツが、幅広いユーザによって創作されることが予測できる。

しかしながら、従来のニコニコ動画のような、コメント文による言語的な共有方法だけでは、コンテンツ提供者にとって有益なフィードバックであるクオリティの向上や、今後の創作活動のモチベーション維持に繋がる具体的なフィードバックを得られないことがある。加えて、エンタテインメントコンテンツにとって基本的な情動である「笑い」について、「どこで笑ったのか」、「どれぐらい笑ったのか?」といった単純な情報でさえ、エンドユーザにとって自然でストレスの無い方法で、リアルタイムで取得する方法は多くは存在しない。

このような背景から我々は、非言語による動画評価のフィードバックの実現として、スマートフォン上の映像視聴時に起きる自然な不随意運動を、スマートフォン上の

¹ 神奈川工科大学
Kanagawa Institute of Technology
^{a)} kitada@shirai.la
^{b)} shirai@mail.com

高分解能加速度センサを用いて取得し、視聴者の主観的な情動を統計的かつ集会的に分類取得するアルゴリズムおよびアプリケーション開発を行っている [2].

2. 関連研究

2.1 動画共有サイトにおける方法

動画共有サイトなどのエンタテインメントシステムにおいて、他のユーザがコンテンツをどのように理解したのか、という情報を共有する事は重要である [3]. YouTube においては視聴回数と「グッド！」といったリコメンドボタン、ニコニコ動画では再生回数に加えて、コメント数、またコメント内部に含まれる「w」（ネットスラングにおける笑いを意味する倍角小文字 w）の数を数える Web サービス「にこぐらふ」などがある [4].

2.2 笑い増幅器

笑い計測に注目した関連研究として、動画視聴者の笑い検出と笑いの増幅を行うことを目的とした福嶋らの「笑い増幅器」がある [5].

笑い増幅器は、映像視聴者の胸部に貼付した電極から得られる筋電位から笑いの検出とその程度を算出し、映像視聴者の笑いに同調した笑いを増幅させる効果があるとされているラフトラック（他者の笑い声）を周囲に配置した人形から再生することで、笑う群衆の中に身を置いているような没入感を与えることにより、動画視聴者に笑いを誘発させることで、笑いを増幅できるシステムである.

2.3 スマートフォンやタブレット端末を用いた動画

視聴時のユーザー分析

倉野らは、iPad に搭載されているマイクロフォン、前面カメラ、加速度センサの 3 つのセンサを複合して用いるマルチモーダルセンシングによって、記録された動画視聴者の発話やしぐさ、動画閲覧時の状況から閲覧している動画に対する視聴者の興味変化の推測をおこない、閲覧者の興味変化を推定できる簡易的なシーンの発見を報告している [6]. しかしながら、スマートフォンに搭載されたカメラやマイクロフォンによる記録は、プライバシーやリソース使用に対するエンドユーザの利益や実用性を考えると、最良の方法とは言えないだろう.

エンドユーザの動画視聴というエンタテインメントエクスペリエンスを阻害しない方法で、笑いのような情動を抽出する方法が研究されている.

加藤らは、抽象的なアニメーション作品視聴時における瞬間的な加速度を計測・分析することで、非言語による視聴情報を取得する手法を提案した [7], [8], [9], [10].

提案手法によって、視聴者から作家に視聴情報を還元するという課題に対して、iPhone3G に搭載された加速度センサによって取得した連続値に注目することで、意識的な

入力を強いることなく、視聴者の身体的反応を計測することができるようになった.

3. 笑い分類アルゴリズム L-PoD について

3.1 L-PoD アルゴリズム

加藤らの方法では、スマートフォンに実装されている 16bit 深度以上の分解能をもった加速度センサを用いて、人間の動作の中でも高周波成分にあたる動作に注目し、0.2[G] 近辺の運動に注目する事で、情動に関係のある不随意運動が検出可能であることが報告された.

そこで、加藤らのフィルタリング手法を利用し、エンドユーザの「笑い」に注目した、同一のスマートフォンデバイスでのコンテンツの再生時における、ユーザ間の高周波加速度センシングの「違いの特性」(PoD; Property of Difference) として取得できる簡易なアルゴリズム (以下、「L-PoD」) を提案する.

具体的には、人間の運動には随意・不随意があるが、笑いにおける不随意運動に注目する. 人間は完全に静止した状態で同じ姿勢を取り続ける事は不可能であるため、ローパスフィルタとハイパスフィルタによる加速度値のフィルタリングをおこなうことで、微小不随意運動を検出することができる (図 1).

(1) ローパスフィルタによるフィルタリング

$$\begin{cases} gx = \text{acceleration.x} * 0.1 + gx * (1.0 - 0.1) \\ gy = \text{acceleration.y} * 0.1 + gy * (1.0 - 0.1) \\ gz = \text{acceleration.z} * 0.1 + gz * (1.0 - 0.1) \end{cases}$$

(2) ハイパスフィルタによるフィルタリング

$$\begin{cases} ax = \text{acceleration.x} - gx \\ ay = \text{acceleration.y} - gy \\ az = \text{acceleration.z} - gz \end{cases}$$

図 1 ローパスフィルタとハイパスフィルタ

Fig. 1 Low-pass filter and High-pass filter

ローパスフィルタは、フィルタ係数 (0.1) を使用し、新しく得られた加速度 (フィルタリングされていない加速度データ) の 10% と前回取得した値 (直前にフィルタリングした値) の 90% を加算することで、ユーザーの 3 軸 (x, y, z) の重力加速度や緩やかな加速度運動を分離させる.

ハイパスフィルタは、取得した加速度から重力加速度 (g_x, g_y, g_z) を引くことで、瞬間的な身体の反応 (微小不随意運動) を取得することができる.

また、ローパスフィルタの値から、式 (1) を用いて、重力成分のマグニチュード (以降, G_{Mag}), ハイパスフィルタの値から式 (2) を用いて、微小不随意運動のマグニチュード (以降, A_{Mag}) として利用する. なお、笑いの検出と分類は、この A_{Mag} の値から判定する.

$$G_{mag} = \sqrt{g_x^2 + g_y^2 + g_z^2} \quad (1)$$

$$A_{mag} = \sqrt{a_x^2 + a_y^2 + a_z^2} \quad (2)$$

笑いの検出と分類をおこなうために集成的特徴 (= PoD) を得るため、各被験者の A_{Mag} から 1 秒ごとの $A_{Mag(i)}$ の平均値とユーザー全体の平均加速度 $\bar{\mu}$ を求め、分散および標準偏差を式 (3) から算出し、表 1 の分類表から視聴者全体の動画の時系列 (1 秒) ごとの笑いの度合いを 4 つのステータスに分類する。

表 1 の各ステータスは、S(Stable) は、視聴者全員の平均値が低く、安定しており、重力以外の加速度が検出できていない状態で誰も笑っていないことを表す。P(Particular) は、人によっては笑っている状態で、加速度センサに大きな値が記録されている。A(All) は、視聴者全員に様に加速度センサに大きな値が検出されている状態である。N(Noise) は、平均値が低く、個々の視聴者によって、異なる行動をしていることを表している。

$$\sigma = \sqrt{\frac{(A_{Mag(n)} - \bar{\mu})^2 + \dots + (A_{Mag(n)} - \bar{\mu})^2}{n - 1}} \quad (3)$$

$n = \text{users}$

表 1 笑いの分類アルゴリズム

Table 1 Classification algorithm of laughing

	平均加速度 > 0.02[G]	
	高い (動的)	低い (静的)
標準偏差 > 0.02[G]	(P) 人によって笑いが起こる (A) 全員笑う	(N) 該当なし (S) 安定

3.2 スマートフォンアプリケーションとデータ収集サーバープログラムの試作

3.1 節で提案したアルゴリズムによる笑いの評価の実験をおこなうために、映像再生と同時に加速度値を取得する iOS および Android デバイス向けの実験用スマートフォンアプリケーションとデータ収集用のサーバープログラムを試作した。本稿では、iOS デバイス用のアプリの基本機能と分析のために使用したサーバー側のプログラムの仕組みを以下に述べる (図 2)。

実験に使用した iOS デバイスは、2010 年に販売された 3.5 インチディスプレイの iPod Touch (iOS6.1.3) である。試作したアプリ上で映像再生を開始すると、再生と同時に 0.1 秒間隔で、現在時刻、3 軸の加速度値、端末の向き、現在の動画再生時間の取得、取得した加速度から重力加速度 (g_x, g_y, g_z) と瞬間的な加速度 (a_x, a_y, a_z)、 G_{Mag} 、 A_{Mag} を算出し、全部で 14 つのパラメータをデバイス内に CSV ファイルとして保存する。

デバイス内で保存された CSV ファイルは、動画視聴終了後にサーバーの役割をおこなう Mac Book Pro (Mac OSX

ver.10.8.4) 上で動作している。Ruby 言語 (ruby 2.0.0dev) と Web アプリケーションフレームワーク Sinatra (ver. 1.4.3) で開発したサーバー側のプログラムに HTTP 通信経由で CSV ファイルのアップロードがおこなわれる (図 2)。

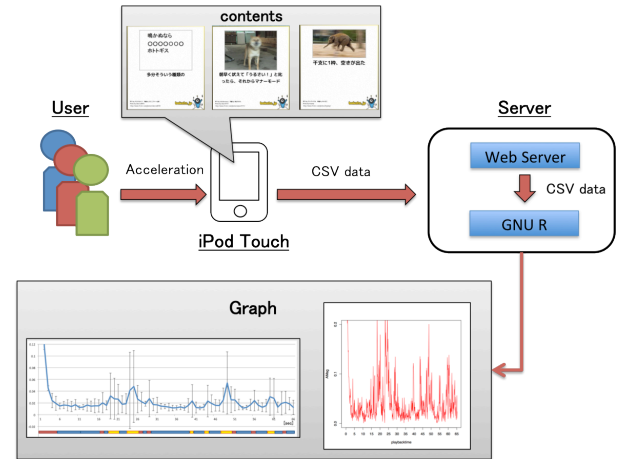


図 2 実験システムの構成

Fig. 2 Configuration of experiment system

また、サーバー側のプログラムは、アップロードされた CSV ファイルをサーバー内に保存し、統計解析ソフト GNU R に CSV ファイルのデータを渡すことで、 G_{Mag} 、 A_{Mag} の各種グラフを自動生成する仕組みも実装している。

3.3 L-PoD による笑い分類実験

試作した iOS 向けアプリケーションを使って、提案するアルゴリズムによる動画コンテンツの笑いの評価の実験をおこなった。

被験者は、日常的にスマートフォンを使っている大学生男性 8 名である。1 グループ 4 名の 2 グループ (A,B) に分け、被験者は椅子に座った状態 (座位) で、自然な形で自由にデバイスを持ち、各グループごとに異なる笑い誘発刺激をおこなう動画コンテンツ (静止画で構成された約 65 秒の MPEG 動画) を視聴する (図 3)。なお、詳しい実験方法については、参考文献 [11] にて、報告をおこなった。

3.4 A_{Mag} による笑い検出

実験終了後にビデオカメラで撮影した実験中の被験者の振る舞いの映像と、デバイスから取得した各被験者の A_{Mag} のグラフを確認しながら、両グループの被験者が笑ったかどうかを検証する作業をおこなった。その結果、実験中に笑った被験者はグループ A が約 2 名であり、グループ B は 3 名だった。図 4 は、両グループの中で、一番笑った被験者の A_{Mag} のグラフである。

この被験者は、ビデオ記録から声を出して笑っているのを確認しており、グラフ上でも 15~25 秒と 45~50 秒の間

に 0.2[G] を超える A_{Mag} を記録している. この被験者の A_{Mag} のグラフを見た場合, 図 4 左のように笑いの情動が確認しやすい被験者は 0.2[G] を超えている. しかし, 0.1[G] 以下の A_{Mag} の値から, 映像での確認を通して, 笑っていることが判断できる被験者も存在した. このことから, 閾値の設定によって, 幅広い笑いを対象とするか, より確実な笑いを対象にするかを選択することが可能である.



図 3 実験の様子
Fig. 3 An user example in experiment

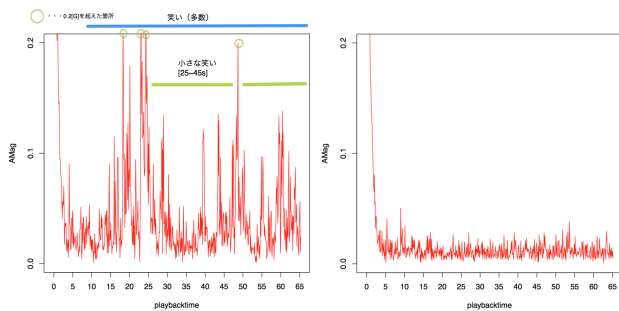


図 4 笑いの確認できた被験者 (左) / 笑いを確認できなかった被験者 (右)

Fig. 4 Laughter(left) / non Laughter (right)

3.5 評価アルゴリズムによる分類

笑った被験者が多かった「グループ B」の被験者のデータから, 笑いの分類アルゴリズムによる集散的笑い評価をおこなった (図 5). 折れ線グラフ下の 4 色 (赤・青・黄・灰) のバーが, 赤色が All, 青色が Stable, 黄色が Particular, 灰色が Not Applicable を表している.

結果として, おおまかではあるが, Stable と Particular の判定には成功している. All に関しては, フィルタの特性上, 実験開始直後の画面操作時の動作の部分に課題が残るが, 75% (4 人中 3 人) の反応で, All と判定されている.

4. Node.js を用いたリアルタイム行動分析

4.1 Node.js を利用したリアルタイム通信の実現

3.2 節で試作したシステムは, 一つの LAN 内にて, Ruby による軽量 Web サーバーである Sinatra フレームワーク

で構成されたサーバープログラムにクライアント側のスマートフォンアプリケーションから, 加速度情報を記録した CSV ファイルをアップロードする機能とサーバー側のグラフ生成機能によって, 分析をおこなった. 実験的な用途としては確実に動作するが, より実用的な環境を想定したスケーラビリティを考慮して, より深くテストされるべきであろう.

将来的な実用を想定した実装としては, リアルタイムなクライアント側とサーバー側の双方向通信によって, ユーザーの振る舞いを取得するものを目標としている. 今回は, WebSocket のサーバーサイド JavaScript での実装である Node.js を用いて, Web アプリケーション版 L-PoD を試作し, Web ベースの分散非同期接続での実装可能性を検討し, LAN/WAN 環境においてベンチマークを行った.

4.2 L-PoD Web アプリケーションの試作

Node.js による Web アプリケーションの開発手順は, クライアント側の HTML5 ベースで YouTube コンテンツを表示する Web ページ (index.html) とサーバーサイド側の Node.js のプログラム (server.js) から構成される (図 6).

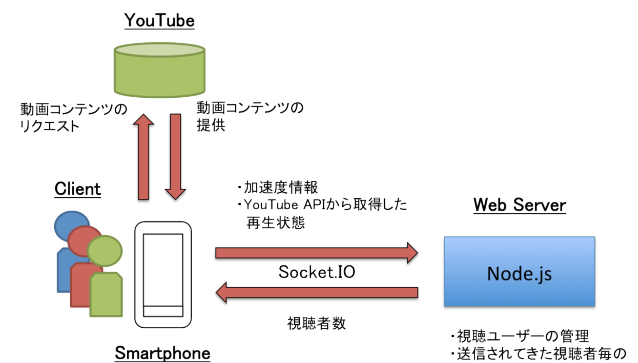


図 6 L-PoD Web アプリケーション版システム構成図

Fig. 6 System structure of web version

クライアント側は, 従来の iOS アプリケーションではなく, iOS および Android 上で動作する Web ブラウザを使用する. 使用する Web ブラウザは, JavaScript によって, Web ブラウザから内蔵の加速度センサにアクセスし, 加速度を取得する devicemotion イベントに対応している必要があり, これに対応した Android 版の Firefox (Android OS 4.0.4) と iPhone 版 Safari (iOS ver.6) を使用する.

クライアント側では, Web ブラウザから加速度センサの値を取得する機能, サーバー側に 0.1 秒間隔で加速度情報等を送信, サーバー側から送られてきたデータを表示・利用するための機能, 動画を再生するための機能の 4 つを実装した (図 7).

これにより, 通常のブラウジングによる HTTP 通信と異なり, Node.js のリアルタイム通信技術 socket.IO によ

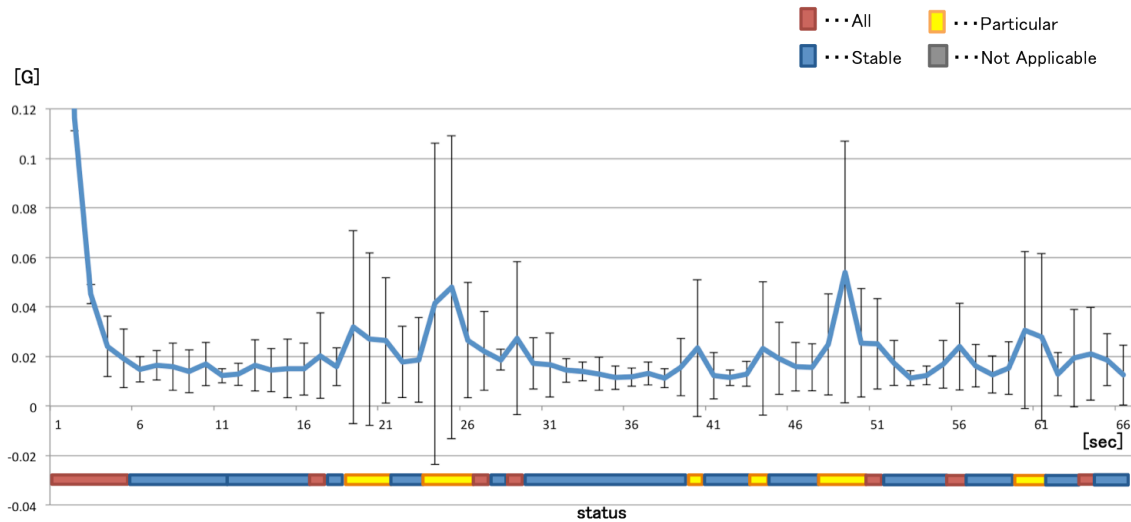


図 5 笑いの分類アルゴリズムによる集会的笑い評価

Fig. 5 A result of classification algorithm by collective laughing evaluation

て、3軸の加速度センサのリアルタイム値（浮動小数点）を高速に送信する事ができる。また、Node.js による分散非同期接続状況を確認する為に、クライアント側の画面上に現在接続中の総クライアント数とこれを可視化した棒グラフが表示される。

動画を再生する機能には、動画共有サービス「YouTube」で公開されている動画を再生する iframe 埋め込み版 YouTube Player API[12] を使用する。JavaScript による HTML5 ベースの動画プレーヤーの制御によって、Web ページ上

で動画の再生を行うのと同時に、3.2.3.3 節で説明したネイティブの iOS アプリケーションと同じく、現在の動画再生時間と動画の再生状況（未完了、再生終了、再生中、一時停止、バッファリング中、頭出し済みの全 5 種類）を取得することができる。

また、デバイスから取得した加速度値から、図 1 よりローパスフィルタ・ハイパスフィルタによるフィルタリング済みの計測値、式 (1) より G_{Mag} 、式 (2) より A_{Mag} の値を算出する。

サーバー側では、3 章と同様の Mac Book Pro を使用し、Node.js を使用したサーバーサイド JavaScript によって、ユーザーの管理全般を担う機能を実装している。クライアント側から非同期で送られてくる加速度情報を受信し、クライアント毎にテキストファイルで保存する。

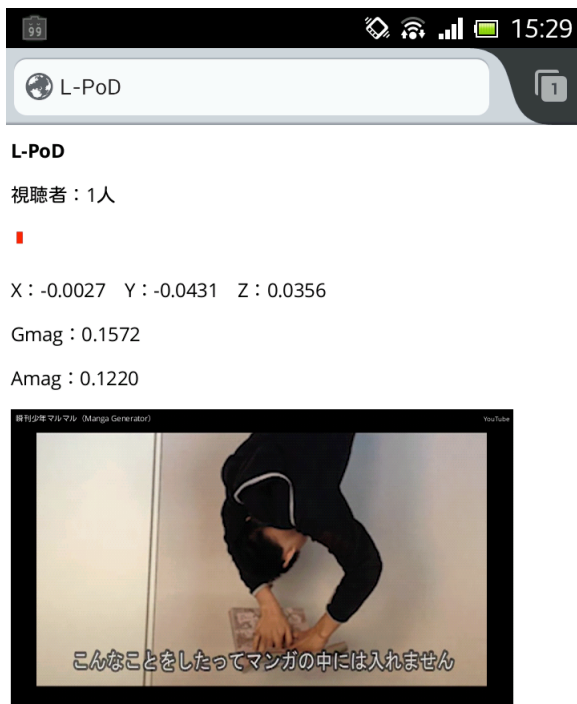


図 7 L-PoD Web アプリケーション

Fig. 7 L-PoD for Web application

4.3 L-PoD Web アプリ版の検証

本 Web アプリケーションを LAN, WAN 環境で使用した結果について述べる。LAN 環境において動画再生終了時まで、0.1 秒間隔でサーバー側に加速度情報と動画の再生時間、再生状況を送信させたところ、遅延無く送受信が確認できた。最大クライアント数は 5 台程度。同様に WAN 環境でのテストとして、Softbank および docomo による 3G 回線を経由して、ルーターのポートフォワーディング機能を使い、NTT フレッツ光経由・NTT ぷららの接続を確認した（図 8）。動画のバッファ状況・回線状況によって、再生遅延が生じるが、加速度センサデータについては、概ね指定の周波数（10Hz）での情報の取得に成功していることが、サーバー側で記録されたテキストファイルにより確認できた。

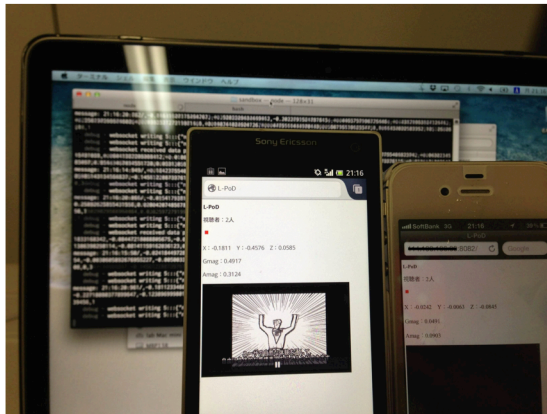


図 8 スマートフォンを使用した WAN 環境でのテスト
Fig. 8 Photo at WAN testing using Smartphones

5. 今後の展望

本実験により、今後より一般化しつつあるスマートフォンデバイスでの動画コンテンツ視聴において、デバイス非装着でユーザーにとって自然な視聴環境で自然な笑いを検出できた意義は大きい。特にデバイスの持ち方を指定せず、加速度センサによって不利な座位での検出に成功した点、および、Web ブラウザでの利用が可能になった点は価値が大きい。

従来の L-PoD はインターネット動画共有サイトのための、専用再生アプリ側に実装する事を想定していたが、今後は、JavaScript によるインタラクティブコンテンツについても、「笑いと楽しさ」という主観的な現象を、自然な体験と統合した簡易な方法で物理評価できるようになり、エンタテインメントシステムでの高精度なコンテンツの評価にも寄与可能になると考える。

具体的には、笑いの傾向を分類のような無意識な行動分析と併せて、facebook における「いいね！」ボタンのような、同意共有や、スマートフォンのシェイクアクションによる、多段階の非同期リコメンド値の共有などが実現できる。おすすめの動画の紹介、ネットワークを活用したテレビグジスタンス、VR コミュニケーションの可能性を大きく広げるものである。

L-PoD アルゴリズムはシンプルなフィルタリングとマグニチュードの算出によって構成されているが、Node.js との連動により、クライアントサイドにおけるリアルタイム化を行う事ができた。従来は、動画コンテンツ視聴中に中間データをサーバー側に送信することで、PoD の算出を現在の動画コンテンツ視聴後に処理することを想定していたが、クライアントデバイス側での演算と連携する事により、分析速度を飛躍的に向上させることができる可能性がある。

参考文献

- [1] 濱崎雅弘, 武田英明, 西村拓一: “動画共有サイトにおける大規模協調的な創造活動の分析”, 情報処理学会研究報告 [音楽情報科学], 一般社団法人情報処理学会, Vol. 2009, No. 11, pp.1-6, 2009.
- [2] 加藤匠, 山下泰介, 田中健司, 早川貴泰, 白井暁彦: “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”, エンタテインメントコンピューティング 2011, 2011.
- [3] 濱崎雅弘, 松尾豊, 武田英明, 西村拓一: “ソーシャルマッチングのための紹介支援システムについての考察”, 知能と情報: 日本知能情報ファジィ学会誌, 日本知能情報ファジィ学会, Vol.20, No.4, PP. 578-590, 2008.
- [4] にこぐらふ: 動画の「笑い」を可視化してみた。 , <http://www.nicograph.jp/>, 2013 年 8 月 10 日アクセス
- [5] 福嶋政期, 橋本悠希, 野澤考司, 梶本裕之: “笑い増幅器: 笑い増幅効果の検証”, ヒューマンインタフェース学会論文誌, Vol.12, No.3, pp.199-207, 2010.
- [6] 倉野大二郎, 松村耕平, 角康之: “マルチモーダルデータを用いた映像閲覧者の興味推定”, 情報処理学会 インタラクシオン 2013, 2013.
- [7] 加藤匠, 白井暁彦, 田中健司, 早川貴泰, 服部元史: “スマートフォンの高精度加速度センサを用いた抽象的動画作品視聴時のユーザ動作分析と作品のクオリティ向上手法の提案”, NICOGRAPH2011 春季大会, 2011
- [8] 加藤匠, 横田真明, 山下泰介, 服部元史, 白井暁彦: “振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, 2010.
- [9] 山下泰介, 白井暁彦, 服部元史: “感圧センサを使った感覚運動インタラクションのための自然なプレイヤ分析アルゴリズムと評価”, NICOGRAPH2011 春季大会, 2011.
- [10] 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, 2011.
- [11] 北田大樹, 鈴木伸之介, 白井暁彦: “スマートフォンの加速度センサを用いた微小不随意運動検出による動画視聴時の笑い評価手法”, 第 18 回日本バーチャルリアリティ学会大会論文集, 2013.
- [12] YouTube Player API Reference for iframe Embeds, https://developers.google.com/youtube/iframe_api_reference?hl=ja#Playback_status, 2013 年 8 月 10 日アクセス