

SEANA: 利用者の動作を強調する 音の拡張現実アプリの開発

吉田 起^{1,a)} 西村 竜一^{1,b)} 入野 俊夫^{1,c)} 河原 英紀^{1,d)}

概要: 本研究では、音の拡張現実システム SEANA を開発した。提案システムは、利用者の動作をスマートフォン (Android 端末) で認識し、その動作と対応づけられた音を外部のスピーカーから鳴らす機能を有する。私たちの日常生活に動作を誇張した音を付与することで、エンターテインメント等での応用を期待することができる。利用者の動作の認識には、パターン認識アルゴリズムであるサポートベクタマシン (SVM) を用い、本研究では、その有用性を調査した。特徴量には、Android 端末で測定が可能な加速度センサ、ジャイロセンサの各出力値を用いた。評価実験では、SVM のカーネル関数を変更した場合の識別率を比較した。

SEANA: Development of Auditory Augmented Reality App. for Emphasizing the Action of the User

ISAMI YOSHIDA^{1,a)} RYUICHI NISIMURA^{1,b)} TOSHIO IRINO^{1,c)} HIDEKI KAWAHARA^{1,d)}

Abstract: We have developed a mobile app. for realizing the sound-based augmented reality system, called "SEANA". The system can recognize actions of the smartphone (Android terminal) users, and has a function to play sound effects according to the recognized actions from the external audio speakers. By applying an exaggerated sound in our daily life, we can expect that it brings new applications of the entertainment field. We have adopted the support vector machine (SVM) as a pattern recognition algorithm for the action of the user. In this paper, we investigated its usefulness. As for the feature vectors for the SVM, we used the output values of the acceleration sensor and the gyro sensor given from the Android terminal. In the evaluation experiments, we compared the identification rate of changing the kernel functions of the SVM.

1. はじめに

本研究では、私たちの行動に音の拡張現実を提供する SEANA (Sound Effect AR with Natural Act) を提案する。スマートフォン (Android 端末) を用いて、利用者の動作を認識し、動作に対応した音を外部スピーカー再生することで、利用者の行動を誇張することができる。

本システムの応用先としては、エンターテインメント分野や、「気付き」が必要な私たちの日常生活を検討している。

エンターテインメント分野では、利用者の動作に対して、誇張された音によってフィードバックを返す応用を想定している。図 1 に、チャンバラごっこ (戦闘シーン) での提案法の利用イメージを示す。プレイヤーは剣を持ったつもりで、手を振りかざし、システムはその動作に対応し、誇張された効果音を外部スピーカーから再生する。例えば、任天堂から発売されている家庭用ゲーム機 Wii では、Wii リモコンを所持しているプレイヤーの動作に対し、振動や音でフィードバックを与えることが可能である。しかし、これは特定のハードウェアに依存した実装であった。一方で、提案システムは、ユーザが所有しているスマートフォンを端末とするため、動作データの取得には新たな機器の導入を必要としない。

また、「気付き」の応用では、歩行音などの誇張表現の利

¹ 和歌山大学システム工学部
Faculty of Systems Engineering, Wakayama University, 930
Sakaedani, Wakayama, Wakayama 640-8510, Japan
a) s165061@center.wakayama-u.ac.jp
b) nisimura@sys.wakayama-u.ac.jp
c) irino@sys.wakayama-u.ac.jp
d) kawahara@sys.wakayama-u.ac.jp



図 1 利用イメージ 1 (チャンバラごっこを誇張する場合)

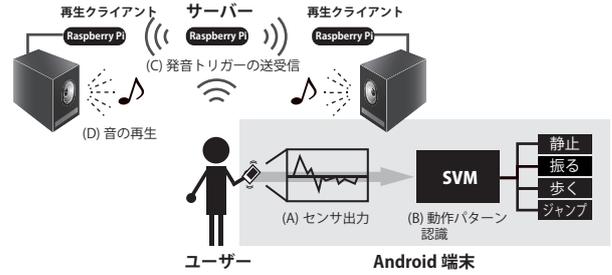


図 4 提案システムの処理フロー

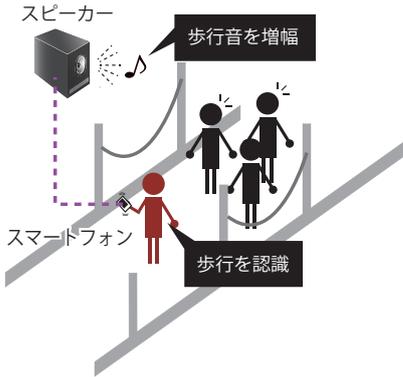


図 2 利用イメージ 2 (歩行音を誇張する場合)

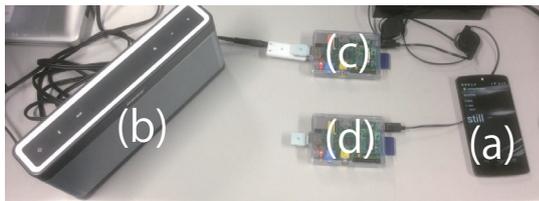


図 3 提案システムのハードウェア構成

用を考えている。音の誇張表現に関し、従来から存在する例としては、子ども向けの「音が鳴るサンダル」(子どもが歩くと、それに併せて「ピコピコ」と音が発生する)が挙げられる。このサンダルには、音が鳴る楽しさを提供に加え、まわりの人間(主に大人)に対して、子どもの存在を知らせるという機能が存在する。別の例では、ハイブリット電気自動車に、走行中のモーター音をわざと誇張する機能を付与する事例がある。これは、誇張された音で、自動車の存在を周りに気付かせ、危険を事前に回避することを意図している。

本システムでは、スマートフォンとネットワーク結合されたスピーカー群を組み合わせることで、上記のような利用者の動作を誇張する音を周囲に提示することを可能とする。例えば、図 2 に示すように、人で塞がっている道を通りたい場合などで、自分の存在を周囲に気付かせる事ができる。将来的には、医療介護の現場で、要介護者の行動を音を使って誇張し、周囲の人間に要介護者の状態を知らせるような利用を期待することができる。

2. 提案システムの概要

提案システムである音の拡張現実システム SEANA は、以下のハードウェアから構成される(図 3)。

- 動作パターン認識アプリを起動している Android 端末
- 誇張音を再生するスピーカー
- b) と接続する Raspberry Pi (音の再生クライアント)
- システム全体の通信を管理する Raspberry Pi (サーバー)

a) は、利用者が所持する端末である。b) と c) は一組をセットで使用する。なお、開発したシステムでは、複数のセットを空間上(部屋等)に配置することが可能である。再生音を誇張表現を高めるためには、同一の空間内に複数のセットを配置する。提案システムは、a), d) 及び b) と c) のセットを同一の無線 LAN ネットワークで接続する。

本システムの処理の流れ(図 4)は、大きく以下になる。

- a) の Android 端末内でユーザの動作に対するセンサ出力を取得
- (B) サポートベクタマシン(SVM)を用いて動作パターンを認識
- (C) 認識結果の情報を d) を経由して、c) に対して音再生のトリガー(以下、再生トリガー)として送信
- (D) c) が再生トリガーを受信、b) から誇張音を再生

2.1 RaspberryPi を用いた音再生機構

空間内にスピーカーを配置する際や利用時の手軽さを考慮した上で、スピーカーと Android 端末は、無線通信(WiFi)で情報を交換する仕様とした。開発の当初、Bluetooth Audio 等を利用して、WAV や FLAC 形式等の音響信号そのものを Android 端末から直接スピーカーに送信する手法を検討した。しかしながら、送信データ量が大きいため、音の再生がはじまるまでに遅延が発生することになった。また、Android 端末内部ではパターン認識による動作認識機構を常に動作させる必要があるため、端末内部の CPU 等負荷を削減する必要があった。よって、提案システムでは、送信する情報量を抑えることができる「再生トリガー」による情報交換方式を採用することにした。

無線通信を担うとともに、スピーカーを駆動するハードウェアとしては、Linux ボードコンピュータの Raspberry Pi を採用した。Raspberry Pi は、小型であり、広く一般に組み込み機器としても利用されている。よって、将来的には、Raspberry Pi とスピーカーを一つのハードウェアに統合することで、空間内に分散配置する際の手間を抑えることができる。また、安価なため、導入コストが低いというメリットもある。

提案システムでは、Raspberry Pi を c) と d) の2つの用途で利用している。c) は、スピーカー b) と有線のオーディオケーブルで接続され、音を再生するクライアントである。d) は、システム全体の通信を管理し、再生トリガーを中継する。

Android 端末から再生トリガーを受信したサーバ d) は、同報通信により、すべての再生クライアント c) に再生トリガーを同時中継することができる。なお、スピーカー b) から再生する音響信号は、すべての再生クライアント c) の内部ストレージに同一のものが保存されている。

3. SVM を用いた動作パターンの認識

提案システムでは、ユーザの動作のパターン認識に、サポートベクタマシン (SVM) [1] を用いた。以下では、開発した動作パターンの認識機構の構成と、SVM の学習に用いる動作の記録データの収集方法について述べる。

動作パターンの初歩的な識別手法として、加速度等のセンサー取得値の閾値超過を判定する手法がある [2]。しかし、実際に閾値超過によるシステムを試作したところ、複雑な動作に対応することが困難であることが判明した。隠れマルコフモデル (HMM) を利用した手法 [3] は、動作パターンの時間方向への伸縮に柔軟に対応できる利点があるが、リアルタイム性に劣ることがわかった。これは、HMM が、一つの動作のセグメント単位で処理するアルゴリズムであり、動作の開始から終了までの区間を切り出してから処理することを原則としているためである。それに対して、SVM はフレーム単位認識を高い精度で実行することが可能なため、高いリアルタイム性を期待することができる。

提案システムでは、SVM の実装には、Chih-Jen Lin らにより開発された LibSVM を用いる。ただし、実際には、Yuan-Ching Spencer Teng らによる Libsvm を Android OS で利用できるライブラリである LibsvmAndroidClassification [5] を用いた。

3.1 特徴量

本研究では、Android 端末を利用者ごとに一台所持し、その加速度センサとジャイロセンサ (角速度センサ) の出力を SVM の特徴量とした。行動認識の特徴量に関しては、複数のセンサを身体の様々な箇所装着して取得する手法 [4] が提案されている。しかし、センサを複数に増やす

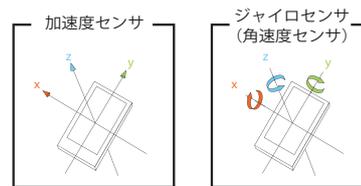


図 5 実験に用いた Android 端末の加速度センサとジャイロセンサ

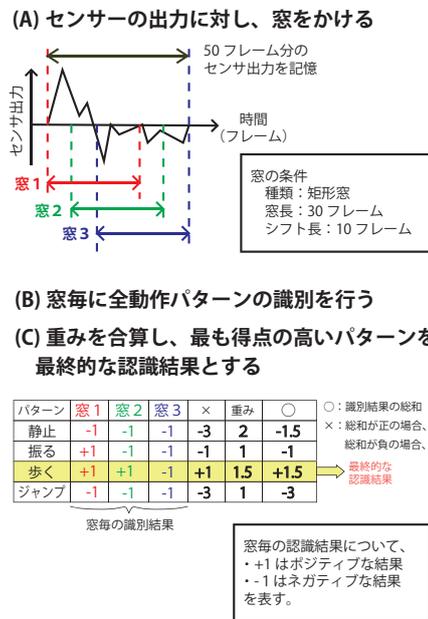


図 6 並列 2 値識別器を用いた多数決による出力決定法

ことは、利用者の利便性の低下や導入コストの増加を招くために難しい。今回、利用者が一台の Android 端末のみを所持した状態に着目し、その際に取得できるデータの活用を検討する。

図 5 に、計測に用いた Android 端末の加速度センサ及びジャイロセンサの各軸の向きを示す。加速度、角速度共に x,y,z の各軸方向、すなわち計 6 軸の計測値を用いた。今回用いたハードウェア (Google Nexus5) では、各センサから 15fps で値を取得することができる。計測の時間長は、2 秒 (合計 30 フレーム) とした。

特徴量には、計測した値そのものである RAW データと平均値を用いた。平均値は、取得した 30 フレーム分の計測値を各軸毎に平均したものである。従って、RAW データの場合は 180 次元、平均値では 6 次元の特徴量で識別処理を実行することになる。

3.2 マルチクラス識別への拡張

SVM は基本的には 2 値識別器であるため、複数のクラスを識別するときにはマルチクラスに拡張する必要がある。本研究では、クラスごとに 2 値識別器を用意し、それらを並列とした場合の多数決による出力決定法を採用した (図 6)。

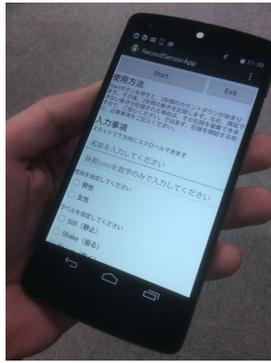


図 7 実験使用機材 (Google Nexus5)



図 8 マニュアル

図 9 動作記録の様子

認識時には、最初に、過去の計測値を含めた 50 フレーム分の特徴量を連結する。次に、図 6(A) に示すように、矩形窓を用いた切り出し処理を行う。この際、矩形窓の窓長は 30 フレーム、シフト長を 10 フレームとした。30 フレームの長さを持つ特徴量系列を 3 つ得ることができる。3 つの特徴量系列をそれぞれ入力とし、識別対象のクラスごとにポジティブ/ネガティブを出力する並列 SVM で 2 値識別処理を行う。出力がポジティブの場合をスコア+1、ネガティブをスコア-1 とする。動作のクラスごとにスコアの合算を行い (各クラスごとに 3 つの出力を合算することになる) 最も高いスコアを有したクラスを最終的な出力とした。なお、ポジティブの頻度が同じ場合も考慮し、経験的に頻繁に出現することがわかっている行動のクラスに対しては、図 6(C) のようにスコアに重みを付与している。

3.3 動作データの収集と識別器の構築

本研究では、独自に動作を計測したデータの収集を行った。動作のパターン (認識時にはクラスとして用いる) は、10 種類 (静止、(静止状態から) 歩く、(静止状態から) 走る、歩き続ける、走り続ける、振る、突く、しゃがむ、ジャンプ、蹴る) である。協力者は、11 名 (20 代の男性 5 名、女性 6 名) である。10 種類の動作を 2 秒間、各 10 回記録するように依頼した。今回の実験では、環境を統一するために、すべての協力者の動作を同一の Android 端末 (Google Nexus5, 図 7) で記録した。動作記録時には、協力者は利き手に Android 端末を持つ。依頼をする際、協力者に動作を指定したマニュアル (図 8) を示し、「どのような動作をしてほしいのか」を説明するようにした。記録中は、図 9 に示すように、常に確認できる位置にマニュアルを設置するようにした。

ここまで説明した動作の記録には、新規に開発した専用 Android アプリ (以下、収集アプリ) を用いた (図 10)。収集アプリは、以下の通り、協力者が自ら操作することで容易に動作を記録することができる。

- (1) アプリを起動。「入力画面」を表示
- (2) (A) に名前や身長などの属性を入力。(B) で今から記



図 10 開発した収集アプリの画面

録する動作を選択。

- (3) (C) の Start ボタンをクリック。「記録画面」に遷移。
- (4) 3 秒間のカウントダウンの後に、動作を 2 秒間記録。

記録したセンサ出力の一部を、図 11 及び 12 に示す。図 11 は、協力者 A の「静止」時の加速度を示す。図 12 は、同じく協力者 A の「ジャンプ」時の加速度である。各グラフの横軸は時間 (フレーム) 縦軸は計測値を示す。

記録した計測値に対して、教師データとなるラベルを付与した。図 13 に示すように、各動作クラスの識別器を構築する際、クラスと協力者が収集アプリ上で選択した動作が同じのとき、ポジティブ (+1) とした。対して、選択したクラスと異なる動作の記録値は、すべてネガティブ (-1) とした。しかし、この方法そのままでは、ネガティブの事例がポジティブに対して多くなり、データ数の不均衡が生じる。そこで、ネガティブデータに対しては削減を行い、量を 10 分の 1 とするようになった。

4. 識別実験によるカーネル関数の選択

4.1 実験の目的

SVM では、カーネル関数の選択が識別器を構築する際の検討事項となる。LibSVM では利用できるカーネル関数には複数の種類があるため、今回、予備実験の性能調査を

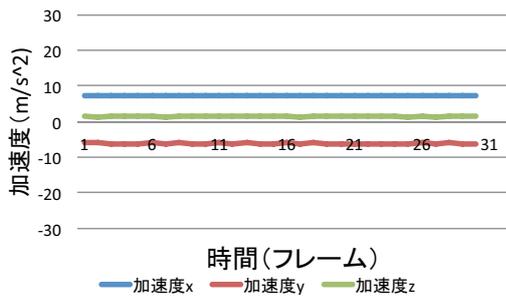


図 11 「静止」時の加速度

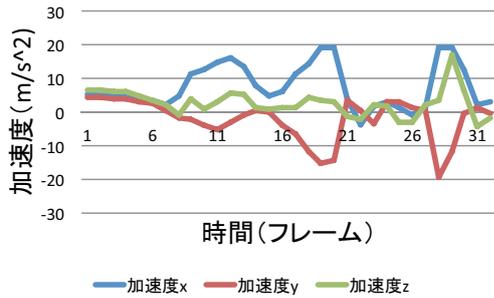


図 12 「ジャンプ」時の加速度

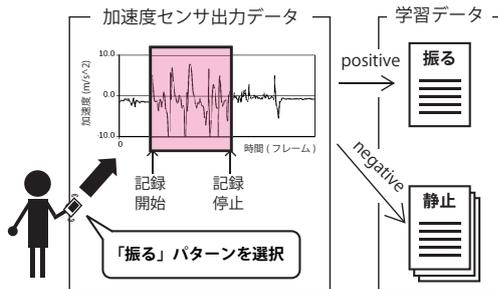


図 13 計測データに対するラベルの決定方法

通じてカーネル関数を選択することにした。また、特徴量として、センサーの計測値(加速度、角速度)のRAWデータを利用する時と平均値を比較することも本実験の目的とする。

比較の対象としたカーネル関数を以下に示す。なお、多項式カーネルにおいては、次元数を2次元から10次元まで変化させ、調査した。

- 線形カーネル (linear kernel)
- 多項式カーネル (polynomial kernel)
- ガウシアンカーネル (radialbasis kernel)

4.2 実験条件

今回の実験は、識別器の構成を決定するための予備の実験であり、識別器の学習と評価に同じデータを用いるクローズド条件にて行った。これにより、識別器のポテンシャルを確認することができる。なお、提案システムでは、マルチクラスへの対応のため、先述のように並列SVMを用いたスコア計算を行っている。しかし、本実験ではカー

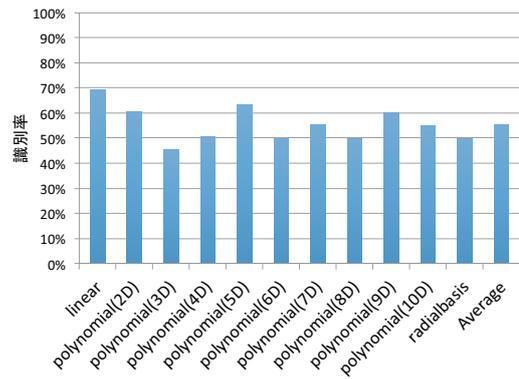


図 14 カーネル関数毎の識別率 (RAW データ)

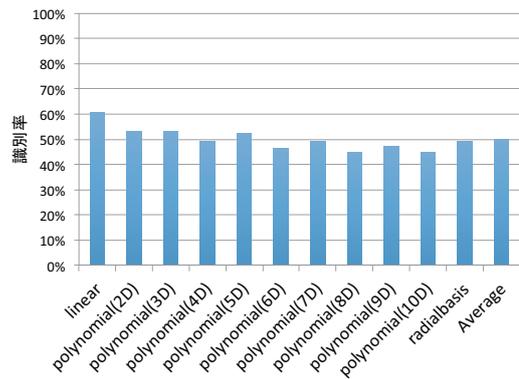


図 15 カーネル関数毎の識別率 (平均値)

ネル関数の選択によるSVM単独での識別能力を調査することが目的である。このため、マルチクラスに拡張した結果は用いず、各クラスの単独の識別器の性能を調査した。

4.3 実験手法

各クラス毎に構築したSVMをそれぞれ使用し、識別結果を記録した。各識別器において、評価データと識別器のクラスが同じときは、ポジティブを出力し、クラスが異なるときはネガティブを出力する事が期待される。それぞれのクラスに対し、正しくポジティブを出力した場合は点数を+1し、ネガティブのときは加算しないようにした。合計した点数を評価データ数で平均し、カーネル関数毎の識別率とした。

4.4 実験結果

実験結果を図14及び15に示す。各グラフの値は、入力したすべての評価データに対する、カーネル関数毎の識別率の平均(以下、平均識別率)である。また、図14は、RAWデータの特徴量として利用した場合、図15は平均値を特徴量とした場合を示す。

この結果より、RAWデータと線形 (Linear) カーネルの組合せが、最も高い識別率 69.2%を示した。しかし、図16のように、動作のクラスによっては、他のカーネル関数を用いた方が識別率が高くなる場合がある。その中には、

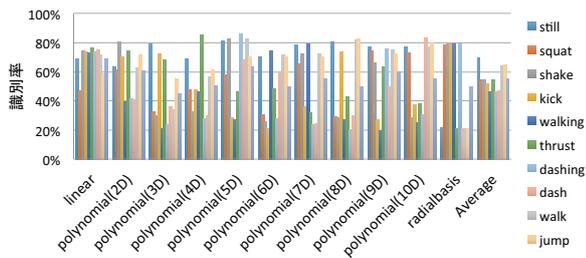


図 16 カーネル関数による識別率変化の詳細 (RAW データ)

80~90%の識別率を示すケースも存在した。よって、動作のクラスによってカーネル関数を使い分けるような工夫が必要であることが確認された。

また、RAW データと平均値の特徴量の違いを比較すると、RAW データが 5.3%ポイント高い性能を示した。従って、提案システムには、特徴量として RAW データを利用する事が妥当であると判断できる。

5. まとめ

本研究では、Android 端末を用いた音の拡張現実システムの開発を行った。このシステムは Android 端末を持った利用者の動作に合わせた誇張音を、外部のスピーカーから再生することができる。

利用者の動作パターンの識別には SVM を利用した。SVM の特徴量とカーネル関数を選択する際の基準とするため、カーネル関数ごとに变化する識別率を実験で求めた。実験の結果、線形カーネルのとき、特徴量に RAW データの利用が最良であることが示唆された。ただし、動作のクラスによっては、線形カーネル以外で高い性能を示すため、並列 SVM の特性を生かして、異なる SVM 識別器の組合せ等を検討する必要がある。

今回は、SVM の構成を決定するための予備的な評価までの実施となったが、マルチクラスでの評価実験を今後予定している。認識精度の向上は継続的に検討すべき課題である。例えば、ゲーム等で活用する場合、利用者の動きと関係のない誇張音は、ユーザ体験に悪影響を及ぼす。屋外で使用する場合は、騒音問題に発展することも危惧される。よって、システム全体の開発と並行して、基盤となる動作識別手法の改良が早急に必要である。

参考文献

- [1] V.N. Vapnik, *The Nature of Statistical Learning Theory*, Springer (1995).
- [2] 大内 一成, 土井 美和子, 加速度と音で日々の生活行動を認識する ActivityAnalyzer, *インタラクション* 2011, pp.255-258 (2011).
- [3] 野津 拓人, 澤田 秀之, 橋本 周司, 加速度センサを用いた HMM によるジェスチャー認識, *情報処理学会第 56 回全国大会講演論文集*, no.4, pp.21-22 (1998).
- [4] 村尾 和哉, 寺田 努, 加速度センサの定常性判定による動作認識手法, *情報処理学会論文誌*, vol.52, no.6, pp.1968-1979

- (2011).
- [5] Yuan-Ching Spencer Teng, LibsvmAndroidClassification <https://github.com/spencerimp/LibsvmAndroidClassification> (2014 年 12 月 19 日アクセス確認).