一般社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
IN2015-96 (2016-01)

アニメ動画の音声とキャスト情報を用いた声優認識

榮田　基希†　　服部　峻††

†,†† 室蘭工業大学 ウェブ知能時空間研究室　〒 050–8585 北海道室蘭市水元町 27–1
E-mail: †12024022@mmm.muroran-it.ac.jp, ††hattori@csse.muroran-it.ac.jp

あらまし　主にアニメ，ゲーム，映画，音楽などの娯楽メディアから音声が流れた時，どこかで聞いたことがあると感じることがある．視聴中のアニメ動画の音声が誰なのかを調べようとするならば，一度エンディングのスタッフロールまで飛ばしたり，タイトルやキャラクター名などで Web 検索を掛けたり，余計な労力を要することになる．そこで本稿では，アニメ視聴中に音声が流れたら，その音声の声優名を認識し，リアルタイムで自動的に画面に表示する声優認識システムを提案する．本システムは視聴中のアニメ動画，及び，声優データベースに格納された音声波形データを用いて類似度を計算して声優認識を行う．さらに，視聴中のアニメ動画のタイトルを用いて検索された Web 上のキャスト情報（キャラクター名と声優名のペアから成るテキスト情報）で声優を絞り込む．
キーワード　声優認識，音声認識，キャスト情報，Web テキスト抽出

Voice Actor Recognition Using Voice and Cast Information of Anime Video

Motoki EIDA† and Shun HATTORI††

†,†† Web Intelligence Time-Space (WITS) Laboratory, Muroran Institute of Technology
27–1 Mizumoto-cho, Muroran, Hokkaido 050–8585, Japan
E-mail: †12024022@mmm.muroran-it.ac.jp, ††hattori@csse.muroran-it.ac.jp

Abstract　When we hear a voice from amusement media such as animes, games, movies, and music, we sometimes feel like that we have heard the voice somewhere. To check whose voice it is in a playing anime video, we have to carry extra burdens of skipping the anime video to the staff roll of the ending once and searching the Web by the anime title and/or character name. This paper proposes a Voice Actor Recognition system that recognize a voice actor's name from her/his voice in a playing anime video and displays the name automatically in real time. The system uses the sound waveform data of a playing anime video and each voice actor's sound waveform data stored in a voice actor database to calculate their similarity. And moreover it filters by cast information (textual information of pairs of a character name and its voice actor's name) on the Web searched by the title of a playing anime video.
Key words　Voice Actor Recognition, Speech Recognition, Cast Information, Web Text Extraction

1. まえがき

近年日本には様々な娯楽メディアがあり，我々はそれらを普段の生活の中で目や耳にする機会が多くなっている．情報通信機器の普及で多くの人が，パソコンやモバイル端末などの機器で番組や動画の視聴，ゲームなどが今では手軽にできる．このような娯楽に触れる機会が多くなって来ると，どこかで聞いたことがある音声が流れて来ることがある．

その音声の発生源がアニメ動画の場合，誰の音声であるかを知るためには，エンディングのスタッフロールまで飛ばしたり，Web で作品のタイトル名やキャラクター名で検索したりするなどの余計な労力を掛ける必要が出て来る．例えば，あるユーザが適当なアニメを視聴していた際，そのアニメの中に出て来たキャラクター A の音声がユーザの聞いたことのある音声であったとする．そこで，そのユーザがキャラクター A の声優について調べようとするならば，エンディングまで飛ばしたり，アニメタイトルやキャラクター名で Web 検索して，そのアニメの公式サイトやウィキペディアなどを探そうとするであろう．しかし，知りたいキャラクター A が作中の目立たない配役だった場合，Web で検索を掛けても中々出て来ないことも考えられる．また，主要なキャラクターではない場合，キャラクター名を記憶していない可能性もあり，エンディングのスタッフロー

ルが流れてもわからないだろう．その上，脇役であった場合，スタッフロールには男の子 B，男の子 C というようにキャラクター名を不明瞭に表記していることもあり，どの場面に出て来たキャラクターかわからないことも考えられる．

そこで本稿では，アニメ視聴中に音声が流れたらリアルタイムに声優名を自動的にアプリケーション内の画面に表示するシステムを提案する．アニメのキャラクターと声優名を関連付けて映像として表示することができる，ユーザ側に「このキャラクターはこの声優だ」と強いイメージを植えつけやすいシステムになると考えた．このシステムを実現するにあたって，データベースにあらかじめ登録してある各声優の音声波形データと視聴中のアニメ動画から流れる音声波形データを使って類似度の計算を行い声優を判定する．また，本稿では YouTube やニコニコ動画など動画サイトで視聴中のアニメのタイトルが特定されて既にわかっている状態を想定する．アニメのタイトルが特定されていることで，そのタイトルに基づいて Web 検索されたキャスト情報で声優を絞り込み，声優認識の精度が上がると考える．まとめると，音声で声優認識するだけではなく，視聴中のアニメ動画が持つアニメタイトルを用いて，Web からキャスト情報を自動で取得してデータベースにある声優を絞り込むようにすることで精度が上がると考える．最終的には図 1 のように，認識した声優名を画面に表示するだけでなく，その声優のプロフィール情報や他の出演作品の情報などを余計な労力を掛けずに提供できるシステムを目指していく．

図 1　最終的なシステムイメージ図

2.　提案システム

2.1　システム概要

アニメ動画に流れる音声から声優名を認識するため，声優に限定しない一般の話者認識や声紋による個人認証などの従来研究 [1–3] を参考にして，それぞれの声優の声の特徴には音声波形の数値の軌跡が異なっているという仮説を立てた．本稿における声優認識システムは図 2 に示す処理を繰り返すことで声優名を認識する．提案システムでは，アニメ動画から流れる音声データを取得して波形表示するのに，Android 標準 API の Visualizer [4] を用いる．Visualizer とは音声の可視化のことであり，音声波形を表示するラインの頂点座標は，左上を基点とする Android 端末上の座標系で表されている．

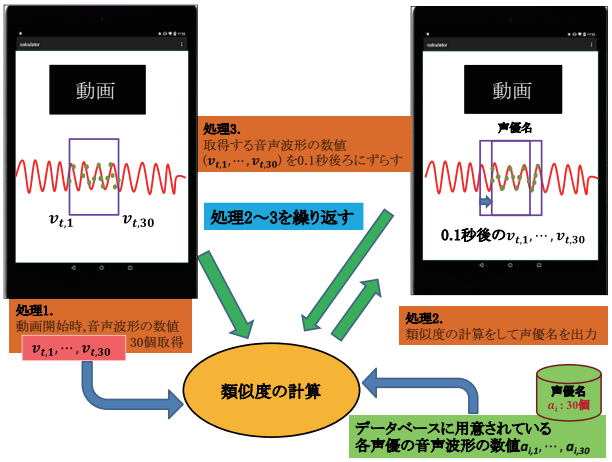

図 2　提案システムの処理の流れ

処理 1 では Android 端末で動画を流して音声波形を表示する．再生位置 t 秒において，新しく生成される音声波形の数値（以下，$\boldsymbol{v_t}$）を約 0.1 秒ごとに 30 個取得する．次の処理 2 で，$\boldsymbol{v_t}$ とあらかじめデータベースに用意されている各声優 i の音声波形の数値（以下，$\boldsymbol{a_i}$）30 個を使って類似度の計算を行って，一番類似度の高かった声優名を画面に出力する．最後に処理 3 で $\boldsymbol{v_t}$ に格納されていた一番古い音声波形の数値を取り出し，約 0.1 秒後の次の再生位置で出て来る新しい数値を格納していく．以後，処理 2 と処理 3 を繰り返す．

2.2　声優データベース

本節では前述に記述しているデータベースの詳細について説明する．データベースに入っている要素を図 3 に示す．中身には，1 列目に声優名，2 列目以降には音声波形の数値 $\boldsymbol{a_i}$ がある．今回は 40 人分の声優データを用意した．つまり，40 人分の声優名と 40 人分の音声波形の数値 $\boldsymbol{a_i}$ が 30 個ある．この $\boldsymbol{a_i}$ は，微妙な誤差はあるが約 0.1 秒毎に記録したものである．よって 1 人につき約 3 秒分の数値が用意されている．データベースに入っている $\boldsymbol{a_i}$ は正規化されていない．

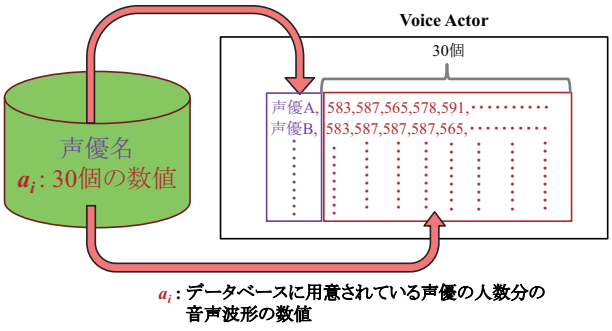

図 3　声優データベースの要素

2.3 音声波形の数値の正規化

類似度の計算をするにあたって，図 3 のデータベースに入っている音声波形の数値や視聴中のアニメ動画から得られた音声波形の数値は Android 端末上の座標であるため，0 を中心に振動する音声波形へと正規化する処理を行う．動画再生時に一番初めに生成される音声波形の数値（以下，startPoint）が基軸になると考えられ，この startPoint を用いて正規化を行った．

$$\boldsymbol{v_t} = (v_{t,1} - \mathrm{startPoint}, \cdots, v_{t,30} - \mathrm{startPoint})$$

$$\boldsymbol{a_i} = (a_{i,1} - \mathrm{startPoint}, \cdots, a_{i,30} - \mathrm{startPoint})$$

2.4 声優認識アルゴリズム

本節では，声優認識の為の類似度の計算方法，その類似度に基づく判定方法，及び，精度を上げるためのキャスト情報を用いた絞り込み方法について説明していく．

2.4.1 声優認識に用いる類似度の計算

図 2 の類似度の計算が行われる処理の詳細について説明する．まず，類似度の計算の為に $\boldsymbol{v_t}$ と $\boldsymbol{a_i}$ の要素を 30 個ずつ用意する．その詳細を図 4 に示す．本稿では類似度の定義として，ユークリッド距離とコサイン類似度，相関係数の 3 種類を用いる．$\boldsymbol{v_t}$ を取得して正規化した音声波形の数値を順番ごとに $v_{t,1}, v_{t,2}, \cdots, v_{t,30}$ と置き直すことにする．同様に，$\boldsymbol{a_i}$ を取得して正規化した音声波形の数値を順番ごとに $a_{i,1}, a_{i,2}, \cdots, a_{i,30}$ と置き直すことにする．以下の式で類似度を算出する．

（1） ユークリッド距離に基づく類似度

$$\boldsymbol{v_t} = (v_{t,1}, \cdots, v_{t,30}), \quad \boldsymbol{a_i} = (a_{i,1}, \cdots, a_{i,30})$$

$$d(\boldsymbol{v_t}, \boldsymbol{a_i}) = \sqrt{(v_{t,1} - a_{i,1})^2 + \cdots + (v_{t,30} - a_{i,30})^2}$$

$$= \sqrt{\sum_{j=1}^{30} (v_{t,j} - a_{i,j})^2}$$

$$\mathrm{sim}(\boldsymbol{v_t}, \boldsymbol{a_i}) = \frac{1}{d(\boldsymbol{v_t}, \boldsymbol{a_i}) + 1} \tag{1}$$

（2） コサイン類似度

$$\boldsymbol{v_t} = (v_{t,1}, \cdots, v_{t,30}), \quad \boldsymbol{a_i} = (a_{i,1}, \cdots, a_{i,30})$$

$$\mathrm{sim}(\boldsymbol{v_t}, \boldsymbol{a_i}) = \frac{\sum_{j=1}^{30} v_{t,j} \cdot a_{i,j}}{\sqrt{\sum_{j=1}^{30} v_{t,j}} \sqrt{\sum_{j=1}^{30} a_{i,j}}} \tag{2}$$

（3） 相関係数

$$\boldsymbol{v_t} = (v_{t,1}, \cdots, v_{t,30}), \quad \boldsymbol{a_i} = (a_{i,1}, \cdots, a_{i,30})$$

$$\mathrm{sim}(\boldsymbol{v_t}, \boldsymbol{a_i}) = \frac{\sum_{j=1}^{30} (v_{t,j} - \overline{v_t})(a_{i,j} - \overline{a_i})}{\sqrt{\sum_{j=1}^{30} (v_{t,j} - \overline{v_t})^2} \sqrt{\sum_{j=1}^{30} (a_{i,j} - \overline{a_i})^2}} \tag{3}$$

式 (1) から (3) のいずれかを用いて，声優データベースに用意されている声優の人数分の類似度が求められる．算出された類似度をそれぞれ比較していき，一番類似度の高い声優が約 0.1 秒の区間毎の声優と判定される．この流れを図 4 に示す．しかし例外として，動画が開始された直後の約 3 秒間は $\boldsymbol{v_t}$ の値が 30 個たまりきっていないため類似度の計算はされない．

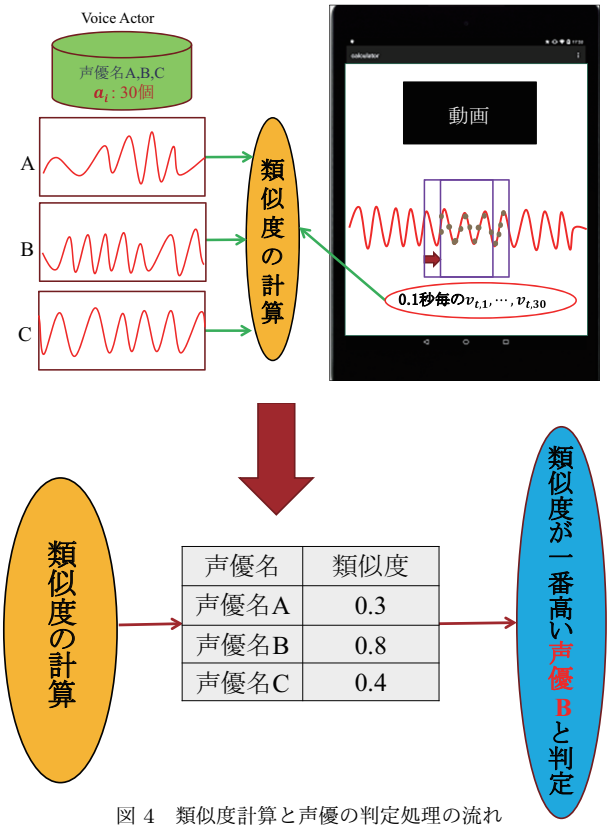

図 4 類似度計算と声優の判定処理の流れ

2.4.2 パラメトリック声優認識

前節の方法で声優認識をすると，約 0.1 秒の区間毎に声優名が判定されて出力される．そこで，約 0.1 秒毎に行う類似度計算及びランキングを連続 N 回分まとめてから声優認識し，その N 回の P%以上をある声優が 1 位をどの声優よりも多く獲得したら，約 $0.1 \cdot N$ 秒の区間はその声優の音声であると判定されるように定義づける．また，どの声優も N 回中 P%以上 1 位を獲れなかった場合には「なし」と判定する．

- N 回 ： 約 0.1 秒毎に行われる声優認識の回数
- P% ： N 回中何回 1 位を獲れば声優認識の解として採用されるかを定める割合

例として N が 10 回，P が 60%のパラメータの場合のシステムの処理を図 5 に示す．図 5 を見ると 0.1 秒区間毎の 1 位の回数が，声優 A が 6 回，声優 B が 2 回，声優 C が 2 回と声優認識されている．この例の場合，声優 A が 10 回中で 60%以上 1 位を獲得していて，どの声優よりも一番多く 1 位を獲得しているので，この 1 秒区間は声優 A であると認識される．

次に，同じパラメータ設定で 0.1 秒区間毎の 1 位の回数が，声優 A が 4 回，声優 B が 3 回，声優 C が 3 回の例を図 6 に示す．この場合，誰も 10 回中 6 割以上 1 位を獲得していないので，この 1 秒区間は誰でもないと判定されて「なし」となる．

最後に N が 10 回，P が 40%のパラメータの場合に 0.1 秒区間毎の 1 位の回数が，声優 A が 4 回，声優 B が 4 回，声優 C が 2 回の例を図 7 に示す．声優 A と声優 B の両者とも 10 回中 4 割以上 1 位を獲得しており，その回数も同じであるため，優劣が決まらない．そこで，1 位を獲った回数が同じ声優が複数存在した場合，今までは 0.1 秒毎に算出していた類似度を，決定戦まで勝ち進んだ声優に対してのみ各々10 回分足した合計で比較する．声優 A の場合 0.1 秒毎の類似度を 10 回足すと 2.5315 であり，声優 B の場合 0.1 秒毎の類似度を 10 回足すと 1.2521 である．よって，声優 A の方が声優 B よりも類似度の合計が大きいので，この 1 秒区間は声優 A であると認識される．

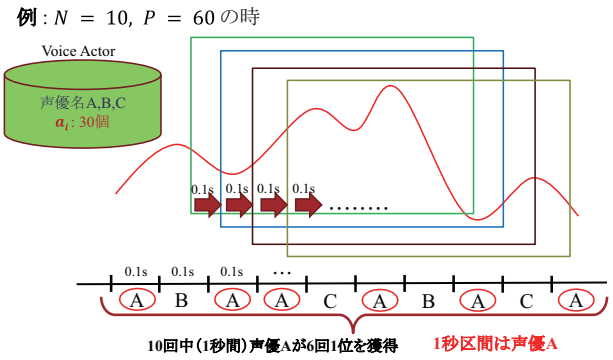

図 5　パラメトリック声優認識の処理の例

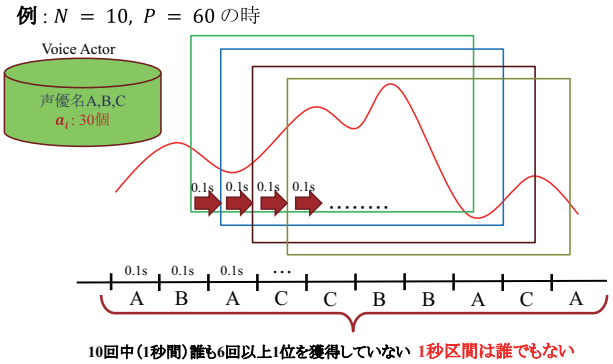

図 6　パラメトリック声優認識で判定「なし」となる場合

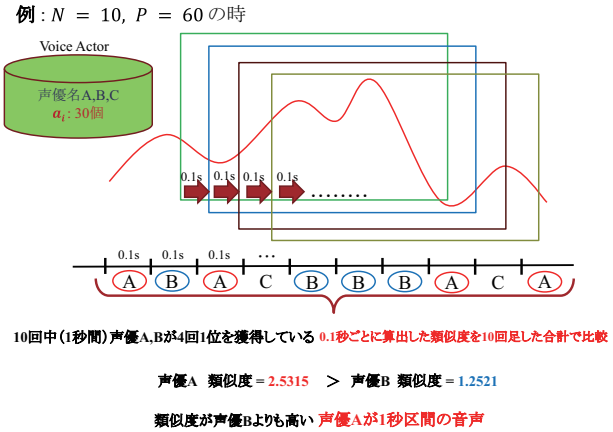

図 7　1 位を獲った回数が同じ声優が複数存在した場合

2 つのパラメータ（N 回と P%）を持つパラメトリック声優認識の処理の流れについて，以上の 3 種類の場合分けを含むフローチャートを図 8 に示す．

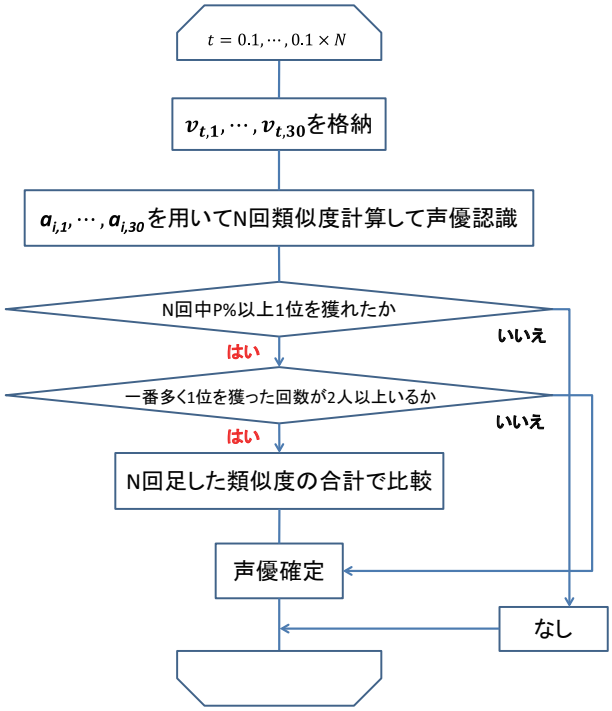

図 8　パラメトリック声優認識のフローチャート

2.5　キャスト情報による声優データベースの絞り込み

音声波形を用いて声優を認識するだけでなく，視聴中のアニメのタイトルを基に Web 検索して取得したキャスト情報（キャラクター名と声優名のペア）を活用することで，声優認識の精度を向上させる手法を提案する．キャスト情報は人手で作成することも考えられるが，Web 検索して自動取得する方法を採用し，ウィキペディアやアニメの公式ページから取得することを想定している．図 9 のように，ウィキペディアなどのソースコードを見てみると定型文が見られるので，自然言語処理を使ってウィキペディアのソースコードからキャスト情報を抽出し，声優データベースの絞り込みを行う．

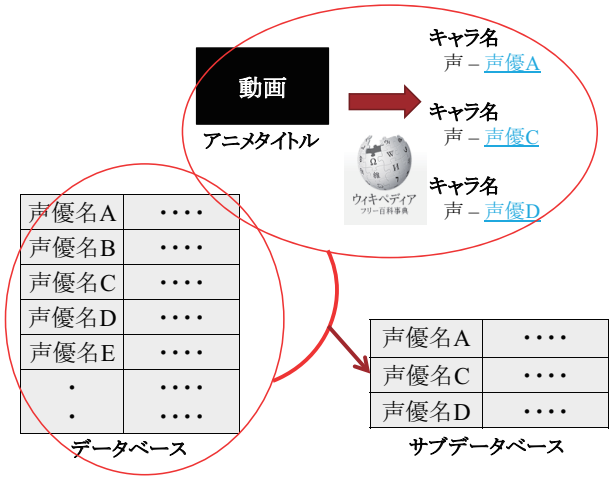

図 9　キャスト情報による声優データベースの絞り込み

3. 評価実験

本章では，3 分のアニメ動画 1 件を用いて，本システムの声優認識の精度に関して評価実験を行う．評価実験用のアニメ動画 1 件に出て来るキャストであるキャラクターと声優のペアは 2 組であるが，このアニメ作品シリーズには全部で 16 名の声優が出演している．また，声優データベースには男性 20 名，女性 20 名の声優 i の名前と音声データ a_i が入っており，評価実験用のアニメ動画 1 件に出て来る声優 2 名は確実に含まれている．このフルの声優データベースに加えて，アニメタイトルで検索したウィキペディアからテキスト抽出したキャスト情報で絞り込まれた声優 16 名が入っているサブデータベースと，評価実験用のアニメ動画 1 件に出て来るキャストだけに絞り込んだ声優 2 名が入っているサブデータベースの 3 種類を用いる．

3 種類の類似度計算と様々なパラメータ設定で声優認識した結果がテキストファイルとして出力される．このテキストファイルを PC 上に実装した評価システムに流すと，2 つのパラメータ（N 回と P%）に応じて認識精度を出力する．声優認識の精度を測る指標として，再現率と適合率の以下の式を用いる．

$$\text{再現率} = \frac{\text{システム認識した正解合計時間}}{\text{正解の声優名の時間}}$$

$$\text{適合率} = \frac{\text{システム認識した正解合計時間}}{\text{システム認識した声優名の時間}}$$

3.1 類似度計算の比較

本システムの評価実験では以下の項目について注目する．

- 3 種類の類似度計算の評価
- キャスト情報を取得した場合としない場合
- パラメータ N 回と P%の最適化

まず，3 種類の類似度計算のうち，本システムではどの類似度計算が最適なのかを比較する．比較する際，キャスト情報を取得している状態で，パラメータを $N = 1$ 回の場合（パラメータ P は関与しない）に固定している．3 種類の類似度計算それぞれの再現率と適合率，F 値を表 1 と図 10 に示して比較する．

表 1　類似度計算の種類に依る声優認識精度の比較 (1)

類似度の計算	再現率	適合率	F 値
ユークリッド距離	0.029	0.020	0.024
コサイン類似度	0.057	0.039	0.046
相関係数	0.060	0.041	0.049

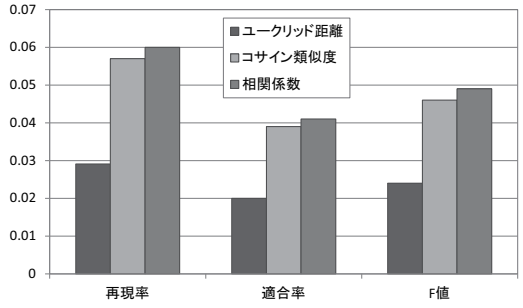

図 10　類似度計算の種類に依る声優認識精度の比較 (2)

表 1 や図 10 の結果から読み取れるようにユークリッド距離を用いるよりも，コサイン類似度や相関係数を用いた方が全体的にシステムの声優認識の精度が高いことがわかる．このような結果が出たのは，音声波形データ間の類似度がユークリッド距離では適切に表現されていないからであり，本システムで用いる類似度の計算にユークリッド距離は好ましくないとわかる．

3.2 キャスト情報の有無の評価

次に，キャスト情報を用いる場合と用いない場合とで比較評価を行う．比較する際に類似度は相関係数を用いて，パラメータを $N = 1$ 回の場合に固定している．キャスト情報の有無それぞれの再現率と適合率，F 値を表 2 と図 11 に示す．

表 2 や図 11 から，キャスト情報を用いて声優認識した方が良い精度を出していることがわかる．声優データベースに入っている声優の候補を絞ることができれば，候補の数を減らすことができるので声優認識の精度の向上につながると考えられる．

表 2　キャスト情報の有無に依る声優認識精度の比較 (1)

キャスト情報	再現率	適合率	F 値
あり	0.06	0.041	0.049
なし	0.028	0.019	0.023

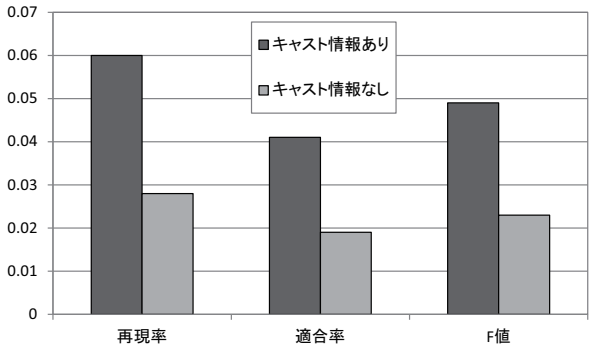

図 11　キャスト情報の有無に依る声優認識精度の比較 (2)

3.3 パラメータ N 回と P%の最適化

N 回と P%のパラメータが再現率と適合率，F 値に影響を与えることが想定されるため検証する．比較する際の前提条件として，類似度は相関係数を用いて，キャスト情報を取得している場合に固定している．上記の条件でパラメータを変動させて実験したところ，再現率と適合率にはあまり顕著な違いが見られなかったため，本稿では割愛する．そこで，様々な条件下での F 値がパラメータの変動に依って，システムにどのような影響を及ぼしているのか検証する．アニメ動画に出て来るキャストだけに絞り込んだ 2 名の声優データベースを用いるのはシステム上は現実的でないが，声優認識の精度の変動をわかりやすく考察するために用意した．

以下のパラメータ最適化の実験で使用する条件は，

- コサイン類似度で，キャスト情報の 16 名に限定している
- 相関係数で，キャスト情報の 16 名に限定している
- コサイン類似度で，アニメ動画に出て来るキャスト 2 名
- 相関係数で，アニメ動画に出て来るキャスト 2 名

の 4 パターンである．この 4 パターンの条件下でパラメータを

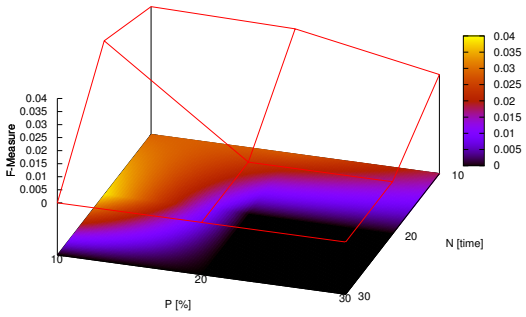

図 12　キャスト情報ありでコサイン類似度を用いた時の F 値

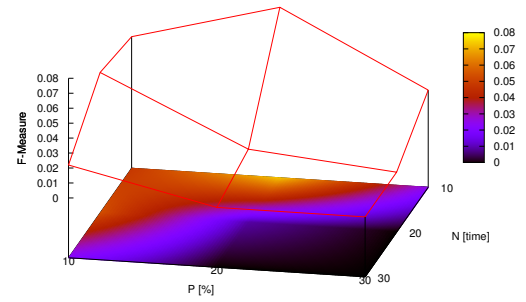

図 13　キャスト情報ありで相関係数を用いた時の F 値

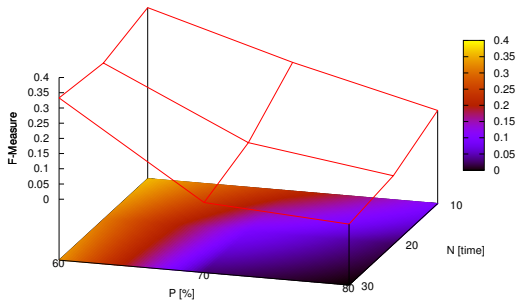

図 14　キャスト 2 人でコサイン類似度を用いた時の F 値

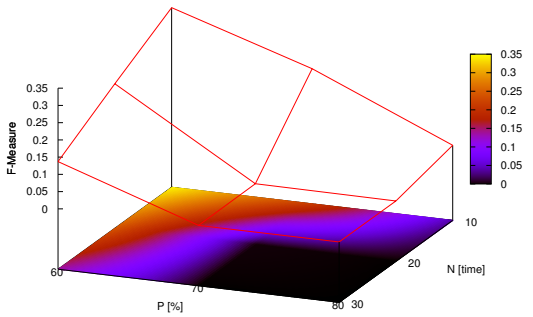

図 15　キャスト 2 人で相関係数を用いた時の F 値

変動させて比較する．図 12 から図 15 の全体を比較してみると，どの条件下でも 2 つのパラメータが小さい場合に F 値が高いことがわかる．これは各パラメータの値が小さいと，判定「なし」となる危険性も低くなるからである．また，F 値が高くなるか低くなるかは，パラメータ P%の変動に依って大きく変わることがわかる．これは N 回の声優認識が精確にされていないため，声優認識にばらつきが生じているのが原因ではないかと考えられる．0.1 秒毎の声優認識にばらつきがあると，N 回中 P%以上の閾値には届かないため，必然的に判定「なし」が多くなる．また，パラメータ P%と同様に，パラメータ N 回の方も少なからず影響を及ぼしている．図 12 と図 13，図 15 から N が大きくなるにつれて F 値が下がっているのがわかる．

4.　まとめと今後の課題

本稿ではアニメ動画から声優を認識するために，動画の音声データを Android 標準 API の Visualizer を用いて音声波形として出力させて，その音声波形から取得できる数値を用いた 3 種類の類似度計算に基づいて声優認識する手法を提案した．さらに声優認識の精度をより向上させるために，Web 上でキャスト情報を取得したり，2 種類のパラメータを設けたり，音声波形データの数値を正規化したり，様々な改善方法を検討した．その結果，キャスト情報を取得してデータベースに入っている声優の候補を出来る限り絞った方が声優認識の精度が向上することがわかった．また，類似度の計算において，ユークリッド距離を用いると著しく精度が低くなることがわかった．パラメータに関しては，N 回毎にばらついた声優認識結果が出ているために高い閾値 P%を設けると途端に精度が低くなることを確認した．全体の考察として，Visualizer で取得する音声波形データを使って声優認識するシステムの精度が低いと感じる．これは，Android 標準 API の Visualizer から取得できる音声波形データが合成波形であるからではないかと考えられる．

今後の課題として，音声の認識の精度の向上を目指していく．まず初めに，今後は Android の他の機能を使って音声データをフーリエ変換して周波数の情報も取り入れることが考えられる．また，本稿では Visualizer の音声波形データの数値の軌跡を用いた声優認識を行ったが，Android 搭載の dB を算出できる機能を使って dB の情報を使うことも考えている．次に，本稿の声優データベースには声優 1 名につき 1 種類の 1 つの音声波形データしか入っていなかったが，複数の種類の複数の音声データを入れておき，それらを組み合わせることで声優認識の精度向上を図る．最後に，本稿で用いた類似度計算だけでなく，他の類似度の定義を用いる方法なども検討していく．

文　　献

[1] 古井 貞熙, “話者認識の現状と展望,” 電子通信学会誌, Vol.67, No.5, pp.537–543 (1984).
[2] 小林 光, 田中 章浩, 木下 健太郎, 岸田 悟, “声紋による個人認証システムの構築,” 電子情報通信学会 ニューロコンピューティング研究会, 信学技報, Vol.108, No.480, pp.13–17 (2009).
[3] @y_benjo, “音声による既婚声優の判別問題,” 日本声優統計学会, 声優統計, Vol.2 (2013).
[4] Google Android – Visualizer, `http://developer.android.com/reference/android/media/audiofx/Visualizer.html`.