

インタラクティブな新しいデジタルサイネージシステムの提案

下川和辰[†] 鈴木正祥[†] 清水哲也[†]

サレジオ工業高等専門学校 情報工学科[†]

1 はじめに

近年、繁華街のいたる所や電車やバスに置いてあるディスプレイなどに、商品の宣伝や天気情報などを伝える、多くのデジタルサイネージが見られる。デジタルサイネージは企業からは「動く広告」として注目されるようになり、1979年、東京の新宿駅東口に施工された日本初の大型街頭ビジョン「アルタビジョン」を始め、2002年にJR東日本が山手線で開始した、「トレインチャンネル」や、夏や冬などの季節や時間帯、温度などを基に適切な商品を消費者に勧める高輝度液晶タッチパネルディスプレイを利用したピーディーシー株式会社の次世代販売機などがある。しかし、街などにあるデジタルサイネージは単にユーザへ情報を伝えるのみで一方通行のため、必ずしもユーザがほしい情報が手に入るとは限らない。そこで、デジタルサイネージをユーザが”操作”して情報を得ることで、ユーザがほしい情報を容易に手に入れることができると考えた。本研究では、Kinectを利用したモーション入力とマイクを利用した音声入力の2つのインタラクティブな要素を加えて、自分が欲しい情報が簡単に得られるような、新しいデジタルサイネージのシステムを提案する。

2 新しいデジタルサイネージの提案

提案するデジタルサイネージでは動画を再生している時を通常のデジタルサイネージとし、画面の前に人が立ち止まると、身体の入力とGoogle社のWebSpeechAPI[1]を使った音声入力を使ってコンテンツの検索、コンテンツの移動やズームを行う。コンテンツの検索は一問一答方式を採用する。また、音声処理、身体処理、全体処理は別々のプログラムで動作している。図1に今回提案するシステム概要図を示す。



図1 提案するインタラクティブなデジタルサイネージの概要図

Proposal of interactive new digital signage
[†]Kazutoki SHIMOKAWA, Masayoshi SUZUKI, Tetsuya SHIMIZU
 Computer Science & Technology, Salesian Polytechnic

音声処理では、音声データを前処理し、Google社が提供しているWebSpeechAPIに転送する。音声認識の結果を基にデータベース検索をする。

身体処理では、Kinectを使って骨格の認識、左右の手のX、Y、Z座標を取得させる。また、コンテンツ移動などはここで処理を行ってから全体処理に出力する。APIは、PrimeSense社が中心となって開発したオープンソースの「OpenNI[2]」を利用する。

全体処理は、音声処理と身体処理の処理結果をまとめている場所である。音声処理から返答された情報を元にコンテンツを表示、身体処理でズームや移動のコマンドが認識されたら、画像のズームや移動など処理する。

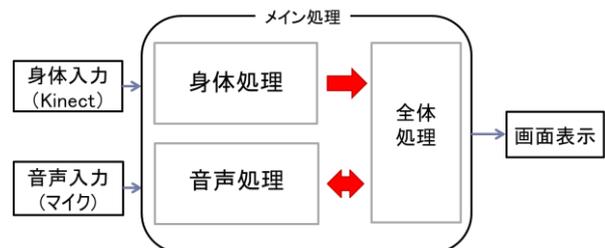


図2 提案するインタラクティブなデジタルサイネージの全体的な処理

提案するデジタルサイネージは、人（骨格）が認識されなくなると、通常のデジタルサイネージに戻るようになっている。

3 WebSpeechAPIの特徴と問題点

今回提案するシステムではWebSpeechAPIを利用しているが実装実験の結果としてWebSpeechAPIの問題点が浮き彫りになった。ここでは、WebSpeechAPIの特徴と問題点を示し改善策を提案する。WebSpeechAPIの主な特徴は、ウェブブラウザ上で動作し、数行のJavaScriptで簡単に音声認識が実装できることである。しかし、Google社が提供しているChromeウェブブラウザのみでしか利用できないが、33カ国の音声認識に対応しており、キーワードを認識できるように人の声の音声パターンを学習したりする必要がなく、簡単に音声認識が使える。また、認識結果は複数の候補が出力される。問題点としては、イントネーションの区別や、検索頻度の少ないワード（造語）などの認識の精度が低い点があげられる。

4 デジタルサイネージでの WebSpeechAPI の有効性の検証

今回のデジタルサイネージの検索では、限られたワードを検索することを仮定している。音声認識がうまく動作しないとインタラクティブではない。音声認識が成功しない原因の1つとして被験者の滑舌の問題がある。そのため WebSpeechAPI の候補機能を使って滑舌の問題が解決できるか調べ、認識率から WebSpeechAPI がデジタルサイネージで有効であるかを調べる。

4.1 仮説

音声認識の結果をデータベースに検索するとき、必ずしも滑舌が良いわけではないので、自分ではそのように言っているにもかかわらず、実際に認識されない場合がある。この問題が起こると、データベースにあるキーワードと一致しなくなるため、検索に引っかからない現象が起きる。そこで、WebSpeechAPI の出力結果の候補機能を使って滑舌の補完ができるのではないかと考えたので検証する。

4.2 実験

あらかじめデータベースには検索されるキーワードとコンテンツ情報が入っている。ここでは、学園祭に関する情報が入っている。実験方法としては、あらかじめデータベースに入っている単語を録音し、それを WebSpeechAPI に音声認識をさせてデータベースに適合するかを調べる。単語数は 272 単語とし、被験者は 10 人とする。WebSpeechAPI の出力結果の候補を含まないデータベース検索と、出力結果の候補を含んだデータベース検索との結果を比較する。認識率はデータベースにヒットしたものを数えるとする。

4.3 結果

実験結果を表 1 に示す。表 1 から、候補を含む場合では認識率が上がっていることがわかる。そこで、なぜ認識率が向上したかについて考察する。

表 1 音声認識の認識結果(被験者 10 人の平均)

	候補を含まない	候補を含む
認識できた数	170.90	214.80
認識率	62.83%	78.97%
計	272	

4.4 考察

ここでは、候補を含まないで認識させ、認識不可だったものと、候補を含むと検索に成功したものを抽出して考える。

候補ありで検索した結果で一番多かったのは、

滑舌の修正が多かった。例えば、「モダンス公演」が最初の認識では「戸田公園」となっていた。また、「5CS」が最初の認識では、「0CS」となっていた。これらは、候補ありの認識で認識されている。これは WebSpeechAPI の結果が検索したいワードに近い音を最初に出力してしまったために、このような現象が起きていると考えられる。

次に空白の修正がある。例として「物理情報研究室」と音声認識させると、「物理情報」と「研究室」の間に半角のスペースが空いてしまう。WebSpeechAPI が自動的に単語に分けて複数のキーワードとして認識してするのではないかと考えられる。元々 WebSpeechAPI は Google 社の Android や Chrome ウェブブラウザの検索用に作られているので複数のワード検索と誤動作してこのような動作をするのではないかと考えられる。

これ以外に、漢字の修正がある。これは、同音異義語の区別がつかないため、最初の結果に違う漢字が出力されてしまうことがある。その場合でも多少であるがカバーできると考えられる。

しかし認識結果から、候補を使った認識率は全体の 16.14%しか改善していないので、あまり認識率が向上してないことがわかる。

4.5 結論

候補ありで検索した結果、微量であるが音声認識が向上し、限りなく同じような発音であるという条件での滑舌の補完が確認できた。また、認識率から WebSpeechAPI がデジタルサイネージで有効であることもわかった。

5 まとめ

本稿では、Kinect を利用したモーション入力とマイクを利用した音声入力で、自分が欲しい情報が簡単に得られるような新しいデジタルサイネージを提案し、WebSpeechAPI の候補機能を使って滑舌の問題が解決できるか調べ、WebSpeechAPI がデジタルサイネージで有効であるかを調べた。今後は SQL でのパターン検索を使用し複数の結果を画面上に表示させ、ユーザビリティ向上をはかっていきたい。

参考文献

- [1]. Web Speech API Specification
<https://dvcs.w3.org/hg/speech-api/raw-file/tip/speechapi.html>
- [2]. OpenNI (Ziv Hende1)
<https://github.com/OpenNI>