

言語情報を用いない音声による直接操作インタフェース

五十嵐 健夫¹

John F. Hughes²

¹東京大学

²ブラウン大学

はじめに

音声入力を利用したシステムでは主に音声の持つ言語情報を利用している。すなわち、音声認識エンジンが音声入力をテキストやコマンドといった離散的なシンボル列に変換し、それを入力としてシステムが動作する。このような方法は、連続音声認識によるテキスト入力や、自然言語によるエージェントとのインタラクションといった目的には適しているが、ユーザの発声から結果のフィードバックまでに時間がかかるために、画面をスクロールするといった連続的かつ機械的な操作には適していない。本論文では、音程や発声の有無といった、音声のもつ音響信号としての側面に注目することによって、機械的な直接操作を効率的に実現する手法を提案する。なお、本手法を説明したデモビデオが www.mtl.t.u-tokyo.ac.jp/~takeo から入手可能である。

関連研究

非言語情報を利用した例としては、まず従来型の音声認識の認識率向上のために利用したものが挙げられる[2]。また、会話型のインタフェースシステムにおいて、相槌のような非言語情報からユーザの状態を推定することによってインタラクションを支援する方法が提案されている[4]。後藤らは、音声入力中の有声休止をトリガーキーとして残りの単語を補完する「音声補完」を提案している[1]。音声によって機械的な直接操作を実現した例として SUITEKey システムがあげられる[3]。これは音声のみによってウィンドウシステムを操作するシステムで、たとえばマウスを動かす場合には「上へ移動...ストップ」というように間に時間を空けて発話すると、「ストップ」と発話されるまでの間マウスカーソルが移動を続ける。本論文では、このような連続的・機械的操作をより効率よく行う方法をいくつか提案する。

提案するインタラクションテクニック

(1) 発声の有無によるオン・オフ操作

本手法を用いたインタフェースでは、ユーザの声がオン・オフボタンとして動作する。すなわち、ユーザが連続的に声を出しつづけている間はボタンが押され続けているものとみなされ、逆に発声がない間はボタンから手が離されているものとみなされる。例えば、ユーザが「ボリュームアップ、あーーーー」と発声すると、「あーーーー」と発声している間ボリュームが上がりつづける。これによって、ユーザはシステムのフィードバックを連続的に観察しながら入力を行うことができる。なお、この手法を利用する際には、有声音だけでなく息をマイクに吹きかけるといった無声音を利用することができる。有声音にくらべて無声音は喉への負担が少ない他、他人への迷惑が最小限に抑えられるといったメリットがある。

(2) 音程によるパラメータの増減操作

本手法は、前出の手法を拡張するもので、連続的な発声の間に音程を変化させることによって補助パラメータの制御を可能とする。言い換えると、音声の一つのパラメータを制御するジョイスティックやスライダーなどと同様の動作をするものである。例えば、ユーザが「上へ移動、あーーーー」と発声して「あーーーー」の発声が続く間だけ移動が起こるような場合に、途中で音程をあげることで速度を上げたり逆に音程を下げることで速度を落としたりできる。移動速度に基づく自動ズームングなどと組み合わせることによってより効率的な動作を行うことも可能である。

(3) タンギングによる離散的パラメータの増減

前出の2手法はテレビの音量のような連続的なパラメータの制御に使われるものであるが、本手法はチャンネルの切り替えのような離散的パラメータの制御に使用するものである。例えば、ユーザが「チャンネルアップ、たったったっ」と発声すると、テレビのチャンネルが4つ上に切り替わる。この手法も前出の手法と同様、音響信号を見ているだけなので、声でなく、手を叩いたり指

を鳴らしたりすることでも同様の効果を得ることができる。

実装

発声の有無の検出は、スペクトルの合計を閾値にかけることで判断している。ノイズの影響を除去するため、低周波領域(375Hz以下)はカットして計算している。音程の変化は、1フレーム前(-12ms)の元のスペクトルと、1フレーム後でかつ周波数を上下にシフトさせた($\pm 43\text{Hz}$)スペクトルとの間の内積を比較することで検出している。すなわち、元のスペクトルと後で周波数を上にシフトさせたものとの内積が、周波数を下にシフトさせたものとの内積よりも大きい場合には、音程が上がったと判断し、逆の場合には下がったと判断される。ユーザの発声する音程が一定の場合には、上昇分と下降分が同程度出現するために、大局的には一定に保たれる。以上のように、本手法では、絶対的な音程を計算しているのではなく相対的な変化のみを検出している。タンギングは、入力音声ストリーム中に見られる短い発声部分を数えることで検出される。音声の有無、音程変化、タンギングの検出は、単純な実装ながら、複数のユーザに対して比較的頑健に動作することを確認している。

議論

通常の音声認識による入力と比較した場合の本手法のメリットとしては、1) 発話してから結果を待つことなく、システムの出力を見ながら連続的に操作を行うことができること、2) 日本語や英語といった特定の言語に依存しないこと、3) 声だけでなく拍手や指のスナップといった身体的な音も同様に利用できること、4) 実装がシンプルであること、が挙げられる。第4の点については、大量の語彙を聞き分けなければならない通常の音声認識に比べて、低レベルの音響信号を見るだけの単純なアルゴリズムであるため、ノイズに強く、話者に依存しない動作が実現しやすいと期待できる。

本論文で提案した手法は、身体的な障害、あるいは特定の作業のために両手が使えない場合の計算機インタフェースとして利用することを想定している。具体的な例としては、自動車の運転中のカーナビゲーションシステムの操作、CAVEのような没入型のバーチャルリアリティシステムで両手がふさがっている場合の操作、あるいはロッククライミングや乗馬など両手がふさがっている状況

での携帯端末の利用などが考えられる。また、これ以外のアプリケーションとして、エンターテインメント向けの利用が考えられる。例として、本手法を利用した簡単なゲームを実装してみた結果、非常に好評であった。個人的な利用だけでなく、劇場などでの観客参加型のエンターテインメントに応用することも検討している。

本手法の問題点としては、通常の会話と異なる不自然な発声が必要となる点が挙げられる。従来型の音声認識でも計算機に認識させるために多少不自然な発声が必要とされる場合があるが、本手法の場合は、音声を連続的に出しつづければならないなど、ユーザへの負担がさらに大きいものと考えられる。また、不自然な発声は、周囲の人間にとっても迷惑となるので、共用のオフィスや公共の場での利用は困難であろう。

本手法は、従来型の音声認識に基づく入力と組み合わせることによって一層の効果をあげるものと期待される。例えば、例にもあげたような、音声認識によってコマンド情報を入力し本手法によってパラメータを調節する、といった組み合わせが有効な使い方と考えられる。今後は、非言語情報を活用したインタフェースについて本論文で提案した3手法以外のものを考えていく他、各種のモダリティを組み合わせたインタフェースについて探っていきたいと考えている。

参考文献

[1]後藤真孝, 伊藤克亘, 秋葉友良, 速水悟, 音声補完:音声入力インタフェースへの新しいモダリティの導入, インタラクティブシステムとソフトウェア VIII, 日本ソフトウェア科学会 WISS2000, pp.153-162, 近代科学社, 2000.

[2]Hirose, Y., Ozeki, K., Takagi, K., Effectiveness of prosodic features in syntactic analysis of read Japanese sentences, Proceedings of ICSLP2000, Vol.3, pp.215-218, 2000.

[3]Manaris, B., McCauley, R., MacGyvers, V., An Intelligent Interface for Keyboard and Mouse Control--Providing Full Access to PC Functionality via Speech, Proceedings of 14th International Florida AI Research Symposium (FLAIRS-01), 2001).

[4]Tsukahara, W., Ward, N., Responding to Subtle, Fleeting Changes in the User's Internal State. Proceedings of CHI 2001, pp.77-84, 2001.