

[研究論文]

住空間にコントローラと情報を重畳するシステムの開発

金子将之¹・杉村博²・黄啓新²

1 博士前期課程 電気電子工学専攻

2 創造工学部 ホームエレクトロニクス開発学科

Development of a System for Superposing Controller and Information on Living Space

Masayuki KANEKO¹, Hiroshi SUGIMURA², Keishin KOH²

Abstract

In this paper, we develop a system to superpose information of web services and home appliances. The system has several functions which display information of web services and control home appliances. The system is controlled by natural actions which are gestures and voice in order to simple control apparatus and is assembled by three major modules which the sound recognition, the gesture recognition and a projector. We describe the entire process, the implementation and the evaluation of a prototype system.

Keywords: Web service, Home appliance controller, ECHONET Lite, Home network, Internet of Thing

1. はじめに

我々は家庭内の様々な場面において、Web サービスの情報を取得する、家電製品を制御するといった行動を取る。これらの行動はスマートフォン、タブレット、リモコンなど専用の端末を用いることが多い。特にリモコンは手で様々な操作が行える点や実装が容易であることから、多様な機器に搭載されており、ユーザはその場から移動することなく操作が行えるといったメリットがある。しかし、専用の端末は機器ごとに操作方法が異なるため、端末が増加するに従って、操作が複雑化してしまう問題点がある。

操作の複雑化を軽減するため、自然な動作で操作することが可能な PUI (Perceptual User Interface) の研究が行われており、ジェスチャ操作や音声認識操作をパソコンや家電製品の制御に利用する研究は盛んに行われている[1][2][3]。これらの研究はいずれも単体の操作方法のみを適用しているため、ユーザの利用シーンによってはデメリットが生じてしまう。例として、ジェスチャ操作は両手が塞がった状態では操作が困難であり、機能が増加す

るほど対応したジェスチャを覚える手間が生じてしまう。音声認識操作は連続した入力には不向きであり、テレビの音量調整、エアコンの風量調整、カーソルの移動といった操作は難しい。これらのデメリットは両者の操作方法を組み合わせることで解決することが可能であると考ええる。また、これらの操作方法に加えて、プロジェクタによって家庭内の様々な場所に情報を重畳させることによって、ユーザは場所に囚われることなく、Web サービス情報の取得や家電製品の制御が行えるのではないかと考える。

本研究では、ジェスチャ操作や音声認識操作といった自然な制御手法を用いて、住空間に Web サービスや家電製品のコントローラといった情報を重畳し制御するシステムを開発し、プロトタイプシステムを用いて評価実験を行うことで、システムの有用性を考察する。

2. システム概要

本システムは、リビングやキッチン、寝室などの住空間に Web サービスや家電製品のコントローラを重畳す

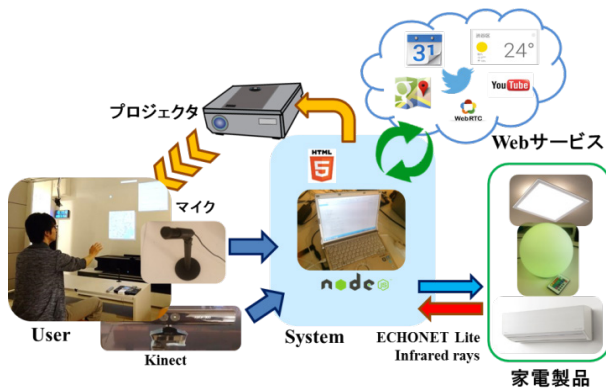


Fig.4 実装したプロトタイプシステム

3.1 Network controller モジュール

Managerの命令内容によってNetworkからWebサービスの情報を取得する。カレンダーはGoogle Calendar, 地図はGoogle Maps, 動画はYouTubeのAPIを利用する。TwitterはTwitter APIを利用し, 天気予報にはlivedoorのWeather Hacksを利用する。テレビ電話はブラウザ間でデータの送受信が可能なWebRTCを用いてビデオチャットとして実装している。

3.2 Gesture controller モジュール

プロトタイプシステムでは, Microsoft社のKinect for Windowsを用いてC++で機能を実装した。Kinectから取得したスケルトンデータをXYZ座標と手を握った状態のGrab, 手を開いた状態のReleaseの2つのジェスチャー情報に変換し, Managerに送信する。

3.3 Speech controller モジュール

音声認識にはGoogle Inc.が提供するWeb Speech APIを利用する。マイクから入力された音声は, Googleのサーバで文字列に変換される。文字列がSCDB内のキーワードと一致した場合は, Managerに一致したキーワードを送信する。

3.4 SCDB モジュール

システムが取得できるWebサービス, 制御できる家電製品のキーワードを文字列で管理する。Webサービスの取得における音声対応表をTable 1に示す。Webサービスの取得では, スケジュール又はカレンダーといった曖昧な表現も含めてデータベースに格納する。家電製品の制御における音声対応表をTable 2に示す。家電製品の制御では, キーワードを1stと2ndに分けて管理する。1stが対象の家電製品の名称で, 2ndが対象の家電製品に対する制御内容である。

Table 1 Web サービス取得における音声対応表

キーワード			Web サービス
スケジュール	カレンダー	予定	Google Calendar
Tweet	つぶやく	Twitter	Twitter
マップ	地図	経路	Google Maps
天気予報	天気		Weather Hacks
YouTube	動画		YouTube
テレビ電話	電話	通話	WebRTC ビデオチャット

Table 2 家電製品制御における音声対応表

キーワード		制御命令
1st	2nd	
照明	点けて	照明を点ける
	消して	照明を消す
	常夜灯	照明を常夜灯モードにする
	全灯	照明を全灯モードにする
エアコン	点けて	エアコンを点ける
	消して	エアコンを消す
	冷房	エアコンを冷房運転にする
	暖房	エアコンを暖房運転にする
	自動	エアコンを自動運転にする
	強風	風量を強風にする
	弱風	風量を弱風にする
	26℃	設定温度を26℃にする

3.5 Superpose controller モジュール

プロジェクトから重畳する情報は, ブラウザにHTML5とJavaScriptで作成したページを表示することで実装した。上部に認識結果を文字列として表示する枠を備える。ユーザの操作によって呼びだされたWebサービス又は家電製品のコントローラは正方形のパネル形状で住空間に重畳する (Fig.5)。また, ユーザの手の動きに合わせてハンドアイコンが移動する。

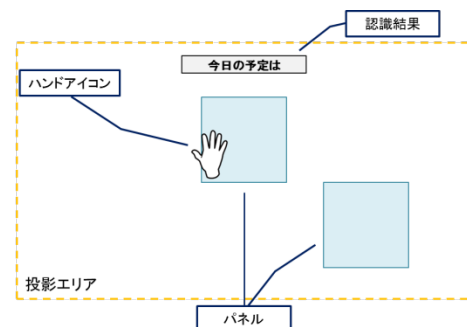


Fig.5 情報の重畳イメージ

3.6 HA controller モジュール

家電製品の制御は、遠隔制御方法として広く普及している赤外線リモコン(Infrared rays = IR)と経済産業省が推奨し、今後の更なる普及が期待される通信プロトコルである ECHONET Lite(EL)を用いる。IR 対応家電製品の制御には、Wi-Fi 機能の付いたオープンソースな赤外線デバイス IRKit [5]を利用し、HTTP で制御する。

4. 評価実験と結果

プロトタイプシステムの評価を行うため、2 種類の評価実験を行った。1 つ目の実験は、Web サービスの表示と家電製品の制御における音声認識操作の認識率と反応速度の評価を行う。被験者は自分の口とマイクの距離が 30cm 離れる地点で椅子に座り、マイクに向かって用意した命令を言ってもらい。命令内容は Web サービスの表示命令が「カレンダー」、「Twitter」、「天気」、「東京から幕張までの経路」の 4 つで、命令の発言終了タイミングから情報が表示されるまでの時間を計測し、10 人の被験者にそれぞれ 20 回ずつ計 200 回の実験を行った。命令通りに情報が表示された場合は掛かった時間を記録し、音声認識されなかった場合は×、間違えた認識をした場合はその認識結果を記録した。家電製品の制御命令は「照明を点けて」、「照明を消して」、「エアコンを点けて」、「エアコンを消して」の 4 つで、命令の発言終了タイミングから家電製品が動作するまでの時間を計測し、5 人の被験者にそれぞれ 10 回ずつ計 50 回の実験を行った。命令通りに家電製品が動作した場合は掛かった時間を記録し、認識されなかった場合は×、誤認識の場合はその認識結果を記録した。2 つ目の実験は被験者の手振りに対するシステムの反応速度の評価である。Kinect から 1m, 2m, 3m, 4m 離れた地点に立ち、被験者が手振りを開始したタイミングから画面のハンドアイコンが反応するまでの時間を計測した。ハンドアイコンが反応しない場合は、一度カメラから外れる、反対の手を振るといった動作を行い、反応するまで継続して時間を計る。それぞれの距離で 10 人の被験者に 20 回ずつ計 200 回の実験を行った。

音声の認識率の結果について Fig.6 に示す。認識率は成功回数を実施回数 200 回で割って算出する。Web サービスの表示命令における成功回数と認識率は「Twitter」が 196 回で 98.0%、「カレンダー」が 189 回で 94.5%、「東京から幕張までの経路」が 173 回で 86.5%、「天気」が 170 回で 85.0%であった。家電製品の制御命令は「照明を点けて」が 48 回で 96%、「照明を消して」が 46 回で 92%、「エアコンを点けて」が 47 回で 94%、「エアコンを消して」が 46 回で 92%であった。全体的に 85%以上の認識率があることから、言い直しによるユーザのストレスやアクセス遅延は発生しにくいと考える。

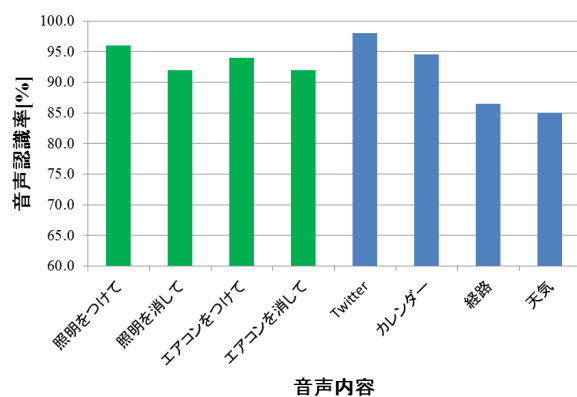


Fig.6 音声の認識率

Web サービスの制御における反応速度の結果を Fig.7 に示す。平均値はそれぞれ命令通りに情報を表示した実験の計測時間を合計し、その値を成功回数で割って算出する。全体の平均時間は 2 秒以下であることから、遅延を感じることなく Web サービスの情報にアクセスすることが可能であると考えられる。

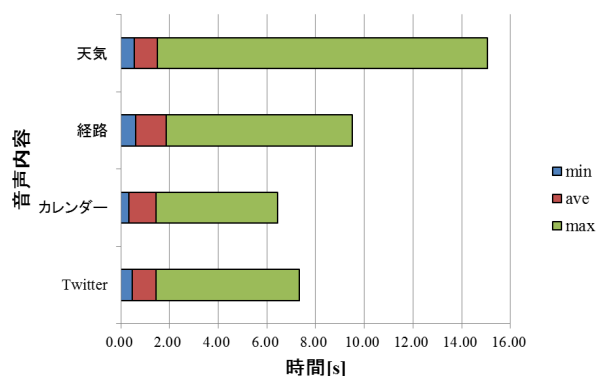


Fig.7 Web サービスの制御における反応速度

家電製品の制御における反応速度の結果を Fig.8 に示す。平均値は命令通りに家電製品が動作した実験の計測時間を合計し、その値を成功回数で割って算出する。全体の平均時間は 1.5 秒程度で、最大時間は 3 秒以内であることから、机のリモコンを手を持ち操作するよりも速い操作が可能であることがわかる。

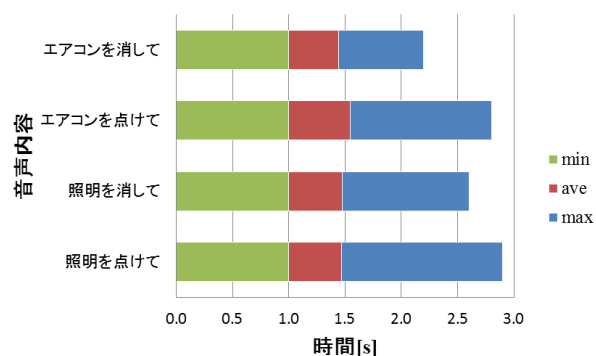


Fig.8 家電製品の制御における反応速度

ジェスチャの反応速度の結果を Fig.9 に示す. 平均値は距離ごとに全ての計測時間を合計し, その値を実施回数 200 回で割って算出する. 距離が 3m までの平均時間は 3 秒以内であり, 手を認識するまでの時間が短いことがわかる. しかし, 4m 地点での認識は 4 秒程度の時間が掛かってしまい, 被験者も遅延を意識したという結果となった.

4m 地点の遅延時間が 1~3m までの結果と比較して 1 秒以上遅い理由として, Kinect のスケルトン認識できる許容範囲が関係していると考ええる. Kinect を用いて手を握る, 開くといったジェスチャに対応するためには, スケルトン認識によって手の位置を把握する必要がある. 今回使用した Kinect のスケルトン認識が可能な許容範囲は 0.8m~4.0m であるため, 他の実験と比べて最大許容範囲 4m で実施した実験では, スケルトン認識が不安定になり, 結果の平均値に影響したのではないかと考える.

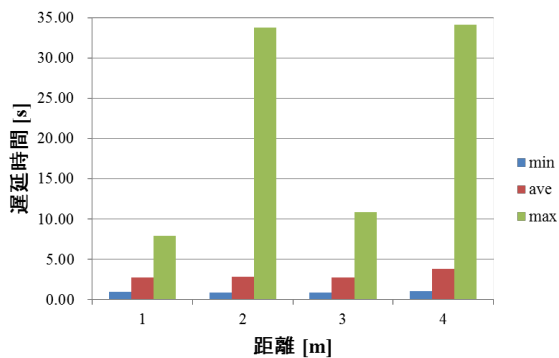


Fig.9 ジェスチャの反応速度

5. おわりに

本研究ではジェスチャや音声認識といった自然な制御手法を用いて, 住空間に Web サービスや家電製品のコントローラといった情報を重畳し制御するシステムを開発し, プロトタイプシステムを用いた評価実験の結果からシステムの有用性を示した. 今後は, 部屋に複数人いる場合の音声識別や投影先の形状から重畳する情報のサイズや量を調整する機能などを検討していきたい.

参考文献

- [1] Lee Sang-Hyuk: "A Kinect Sensor based Windows Control Interface", International Journal of Control & Automation, Vol. 7, No. 3, (2014).
- [2] Jun-ichi Iwashita: "Development of an Interface System Based on Hand Gesture Recognition for Electrical Appliance Operation in Daily Life Scene.", IEEJ Transactions on Electronics, Information and Systems, Vol.107, No.1, pp.676-685, (2010).

- [3] Noriyuki Matsubara: "Evaluating Habituation Effect on Conversational Voice Control in Home Network System", Technical report of IEICE, 111(107), pp.047-052, (2011).
- [4] Kinect, <https://dev.windows.com/en-us/kinect>, (2015).
- [5] IRKit, <http://getirkit.com/>, (2015).