

身体動作認識技術を用いた機器制御に関する研究

企画情報室 コンピュータ応用チーム 岩崎 健太, 村山 大策

要 旨

近年着目されている人間の身体動作を認識する技術を利用して機器制御を行うことを課題とし、前年度までに要素技術の研究、システムの実装を行ってきた。しかし、様々な機器の制御のためにそれぞれ専用システムを開発することはコストを要するため、ユーザにとっては導入の敷居が高い。本研究では、身体動作を利用した機器制御を、汎用的なシステムとして低コストで実現することを目指し、Android 端末を利用したプロトタイプシステムを実装した。実装したシステムについて簡易的なユーザビリティ評価を行った結果、システムの設定操作がわかりやすいという意見があった一方、認識に関するフィードバックが不足しているという意見もあり、今後のシステム改良、実用化に向けての知見が得られた。

1. 諸言

センシング技術の進歩に伴い、比較的安価なセンサデバイスで、人間の体全体の位置や、手、指の位置を非接触に認識することが可能になっている。それに伴い、人間の身体の位置の変化をジェスチャ入力として認識し、コンピュータ上で利用するという技術が注目されている。このジェスチャ入力は、直観的に入力を行うことができ、入力装置にも直接触れる必要がないため、ユーザの位置的制約が小さく、安全衛生面においても優れているという特長がある。

前年度に実装した簡易的なシステムでは、Kinect¹⁾で非接触に取得した身体位置の座標データからジェスチャを判定し、DC モータ駆動の玩具の制御を行った²⁾。このシステムでは、使用するジェスチャの定義や、そのジェスチャを行ったときにどう制御するかは、全て開発時に決定していた。しかし、ジェスチャ入力の応用的利用に関しては、まだ十分に知見が得られておらず、完成したシステムに対して、改良が必要になることも考えられる。例えば、ジェスチャにどのような制御内容を対応付ければ良いかは、機器、センサデバイス、ユーザ等によって異なるため、実際の使用環境において長期間運用することで初めてわかる、といったことが考えられる。また、センサデバイスは、近年、次々と新しく高精度な製品が登場しているため、あるセンサデバイスに特化したシステムを実装してしまうと、新たなセンサデバイスに対応させるには、大きな改良が必要となる。このように、ジェスチャ入力は発展途上の分野であることから、ユーザにとってはシステム導入の敷居が高く、エンジニ

アでなければ気軽に試すことも難しい。

そこで、本研究では、ユーザがカスタマイズ可能で、センサデバイスに極力依存しない、汎用的かつ低コストであるシステムを提案する。

2. 提案手法

2.1 Android 端末の利用

汎用的かつ低コストであるシステム実現のために、スマートフォンやタブレット端末として広く利用されている Android 端末に着目した。この Android 端末が、ジェスチャ入力を行うユーザと制御対象の機器との仲介役となり、制御内容の決定、ジェスチャの管理、画面表示等を担う。

この方法を取ることで、以下の利点が得られる。

- (1) Android 端末は、多くの人が既に所持しているものを使用可能であり、新たに購入したとしても比較的安価であるため、低コストである
- (2) Android 端末上のアプリケーションを操作することで、ユーザがプログラミング等の知識を持っていなくても設定のカスタマイズが可能
- (3) Android 端末が、ジェスチャのフィードバックを提示する表示装置の機能も兼ねることができると、別個にモニタが不要

2.2 2次元座標データフォーマット

本システムで利用することを想定しているセンサデバイスは、Kinect や Leap Motion³⁾ 等、複数があり、今

後も、新たなデバイスが登場する可能性が高い。これらは全て異なる仕様、異なる特徴を持つが、いずれも、なんらかの身体位置の2次元座標あるいは3次元座標を利用するものである。これら座標を、汎用的な2次元座標フォーマットとして統一化することができれば、フォーマットに従った座標データを通信することで、センサデバイス間の差異を軽減できると考えた。

2.3 無線通信

Kinect等のセンサデバイスは、非接触に、身体位置の座標を取得するものであり、ユーザの位置的制約を小さくできることが利点の一つである。その利点を活かすため、センサデバイスとAndroid端末との通信には無線通信を採用する。これにより、ユーザが必ずしも機器のすぐ近くにいないとしても、ジェスチャ入力により制御を行うことが可能になる。

3. システム実装

3.1 システム構成

図1のような構成でシステムを実装した。

使用したハードウェアは以下の通りである。



図1 システム構成

センサデバイス : Kinect
Leap Motion

パソコン : Windows 7 Ultimate
Core i5 2500K 3.30GHz
RAM 8GB

Android 端末 : SHARP 製 SH-04E
Android 4.1.2

マイコン : PIC18F14K50
LPC11U24

3.2 センサデバイス概要

3.2.1 Kinect

Kinectは、Microsoft社が2010年に発売した、人間の身体位置を非接触に取得できるセンサデバイス（図2）である。ユーザから40cm~4m程度の距離に設置し、ユーザの方に向けて使用する。身体位置は、頭部、胴、両手、両足等、20か所の3次元座標が得られる。ジェスチャ認識は標準機能としては付属しておらず、開発者が身体位置座標をもとに実装する。



図2 Kinect

3.2.2 Leap Motion

Leap Motionは、Leap Motion社が2012年に発売した、人間の手、指の位置を非接触に取得できるセンサデバイス（図3）である。卓上に上向きに置き、上部空間の一定範囲内にある手、指の3次元座標を取得する。なお、Leap Motion自身にも独自のジェスチャ認識機能が付属しているが、手や指に特化した独自のものであり、本システムでは、2.2で述べたように、2次元座標フォーマットの使用による汎用化を目指しているため、使用していない。



図3 Leap Motion

3.3 ソフトウェア実装

3.3.1 Android アプリケーション

[開発環境]

- ・ Eclipse 4.3.0
- ・ Java
- ・ Android SDK

Android アプリケーションは図4の流れで処理を行う。

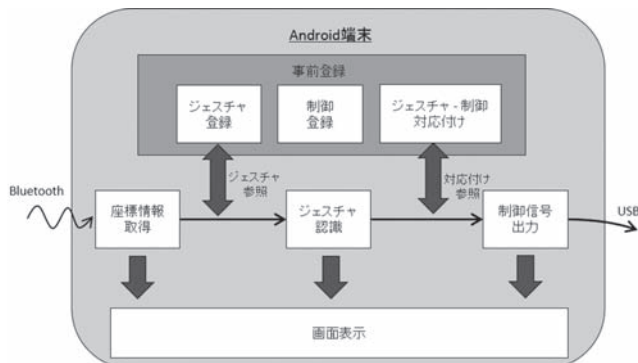


図4 Android アプリケーションの機能

以下に、主な機能の詳細を述べる。

(1) ジェスチャ登録

本システムでは、汎用性を高めるため、既定のジェスチャのみを使用可能とするのではなく、ユーザが追加および削除できる仕様とした。

ジェスチャの登録は、直観的かつ手軽に行えるよう、図5の様に、Android 端末の画面上において、指で軌跡を描く動作により行う。軌跡は、新しいところほど濃く、古いところほど薄く表示することで、視覚的に順序を判断できるようにしている。入力後は「登録」ボタンを押し、名前を付けることで登録が完了する。ここで、ジェスチャに名前を付けるのは、制御との対応付けの際の識別のためである。登録したジェスチャを一覧表示する機能も有しており、その一覧表示機能内でジェスチャを削除することが可能である。

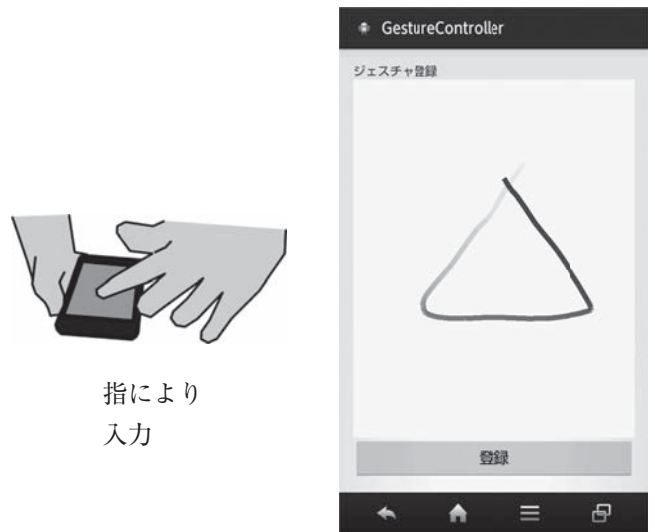


図5 ジェスチャ登録画面

(2) ジェスチャ制御対応付け

登録したジェスチャ動作を行った際に、マイコンに対してどのような制御命令を送るかをジェスチャごとに設定する。

設定は、一般的な Android アプリケーションでも採用されているリストとドロップダウンリストを使用したものであり、図6のようにジェスチャ名に対応した制御内容をそれぞれ選択することで設定する。



図6 ジェスチャ制御対応付け画面

(3) 画面表示

Android 端末の画面には、図7のように、座標の軌跡、認識したジェスチャ名、対応した制御内容を、それぞれ表示する。画面表示は、ユーザが、自身の操作が正しく行えているかをフィードバックし、操作性を高めるために必要である。



任意のセンサデバイスから
2次元座標を送信

図7 ジェスチャ画面表示

(4) ジェスチャ認識

ジェスチャの認識には、\$1 Recognizer⁴⁾を使用した。このアルゴリズムは、ジェスチャの大きさや角度の違いがあったとしても、比較的柔軟に対応して認識することができるため、ユーザやセンサデバイスを限定しないシステムに適していると考えられる。また、身体動作によるジェスチャ認識においては、ユーザに対して即座のフィードバックがあることが望ましいが、2013年時点の汎用的なAndroid端末でも十分高速に動作したことから、速度面においても実用上問題ないアルゴリズムであるといえる。

(5) 制御信号出力

認識したジェスチャ、および、予め設定しておいた制御との対応付けにより、Android 端末は制御内容を決定し、その内容に応じて、USB 接続されたマイコンに制御用のデータを送信する。マイコンへ送信するデータは、USB HID クラス上で独自フォーマットにより通信する。USB 通信はAndroid 端末側を親であるUSB ホスト、マイコン側を子であるUSB デバイスとして行う。Android

端末に多く採用されている micro-B タイプの USB 端子に接続して使用する場合には、Android 端末を USB ホストとして認識させるためのアダプタを使用する。

図8は、Android 端末である汎用的なスマートフォンから、アダプタ、USB ケーブルを介して PIC18F14K50 マイコンを制御している様子であり、LED の点灯によりその動作を確認している。

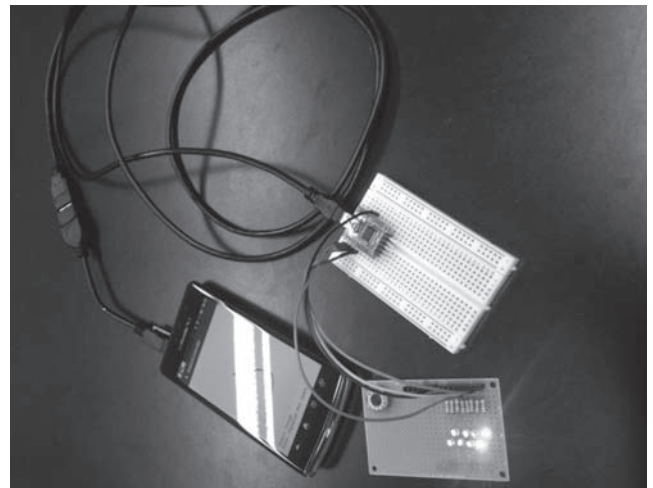


図8 制御動作確認

なお、USB 通信部分の実装には、Android SDK の USB Host API を使用しているため、使用する端末は Android 3.1 以上のものに限定される。

3.3.2 マイコン用ソフトウェア

機器に対して電気信号を出力するマイコン用のソフトウェアとして、以下の2種類で実装を行った。

(a) PIC18F14K50

[開発環境]

- ・ MPLAB X IDE 1.51
- ・ C 言語
- ・ Microchip Libraries for Applications

(b) LPC11U24

[開発環境]

- ・ mbed
- ・ C++

本システムでは、Android 端末とマイコンとを USB 接続して通信するため、USB デバイス機能を搭載したマイコンとしてこれらを選定した。

実装内容は、USB ホストである Android 端末から送信される独自フォーマットの制御データを受け取り、その内容に応じて基本的な出力処理を行うというものである。一般的なマイコン出力として、デジタル出力の ON, OFF, 切り替え, および PWM 出力の値変更, UART 通信による任意のシリアルデータ送信を行えるように実装した。

3.3.3 センサデバイス用ソフトウェア

センサデバイスから Android 端末に対して座標データを送信するためのソフトウェアを、Kinect と Leap Motion についてそれぞれ実装した。

(1) Kinect

[開発環境]

- ・ Visual Studio 2010
- ・ C#
- ・ .NET Framework 4.0

[座標データ]

右手座標を対象とし、正面から見て、左右方向の位置を X 座標、上下方向の位置を Y 座標として送信

(2) Leap Motion

[開発環境]

- ・ Processing 2.1.1
- ・ Java

[座標データ]

最も奥にある指を対象とし、左右方向の位置を X 座標、上下方向の位置を Y 座標として送信

なお、それぞれ 3 次元の座標データを得ることが可能であるが、手前、奥方向の座標は使用せず、2 次元座標として送信する。

4. ユーザビリティ評価

実装したシステムの動作を確認し、システムに対するユーザからの意見を得るため、本研究所職員の協力のもと、簡易的なユーザビリティ評価を行った。なお、実験協力者は日常的にスマートフォンを使用しており、簡単な電子制御の知識を持つ。

4.1 方法

実験協力者にはシステムを自由に使用させ、終了後、インタビューを行うという方法で評価を行った。なお、

使用するセンサデバイスの取得座標、取得精度により評価が異なることが考えられるため、Kinect, Leap Motion の 2 種類のセンサデバイス、および、画面タッチジェスチャ操作により座標を送信する Android アプリ(異なる Android 端末から使用)、の計 3 種類で行った。3 つ目の Android アプリを用いたのは、センサデバイスの性能に影響されず、確実に座標を送信することができた場合のシステム動作を確認するためである。また、簡略化のため、マイコンの制御内容は 8 つのピンのデジタル出力の切り替えのみに限定し、それぞれのピンに対応した LED の点灯により制御結果を表示することとした。

4.2 結果

以下にインタビューから得られた実験協力者の意見を記す。

- ・「ジェスチャの登録方法はシンプルでわかりやすい」
- ・「ジェスチャと制御との対応付けもわかりやすい」
- ・「画面表示だけではジェスチャの認識結果がわかりづらい。スマートフォンなので、音や振動でも知らせてくれた方がいい」
- ・「センサデバイス使用時に、ジェスチャが認識開始しているのかわからず、意図しないジェスチャを行ってしまうことがある。」

また、実験協力者の使用の様子を観察した結果、Kinect, Leap Motion のいずれも、おおそ座標は正しく認識されていたが、稀に座標が飛ぶ、座標にぶれが生じるなどの事象が見られた。

4.3 考察

Android アプリケーション上でのジェスチャ登録、ジェスチャ制御対応付けの設定はわかりやすいとの意見が得られた。これは、本システムが一般的な Android アプリケーションの GUI (グラフィカルユーザインタフェース) に沿って実装しており、実験協力者が普段から Android スマートフォンを使用していることから一般的な Android アプリケーションの操作に慣れていたためであると考えられる。スマートフォン操作に慣れていないユーザでは異なった結果となることも考えられるが、平成 25 年度現在、スマートフォン利用者は非常に多数を占めており、今後も増加することが想定される。そのため、将来的にはスマートフォン操作がより一般化し、多くのユーザにとって抵抗なく使用可能となること

が予想される。

実験協力者より、ジェスチャに対するフィードバックが不足しているとの意見が得られた。フィードバックの必要性については認識しており、画面表示という形式で提供していたが、それだけでは十分ではないという結果であった。これは、実験協力者はスマートフォンを卓上に置いて使用し、ジェスチャの動作に集中しつつ、制御先のLEDの点灯を確認していたため、スマートフォンの表示を注視しづらい状況にあったことが原因として考えられる。このような状況は他のユーザでも十分に起こりうると考えられるため、今後、画面表示の強調、あるいは、音声や振動を追加し、視覚だけでなく聴覚や触覚も含めたフィードバックを与えることが必要であると考えられる。

実験協力者からは、ジェスチャ認識開始のタイミングがわかりづらいという意見も得られた。ユーザが意図した通り、ジェスチャを開始できている場合には、システムが認識中であることのフィードバックをユーザに与えることで解決可能である。しかし、意図しないタイミングでジェスチャを開始した場合には、根本的な解決にはならない。そのため、認識アルゴリズムの見直し、あるいはセンサデバイス用ソフトウェアにおいて、ユーザが開始位置を指定できるような仕組みを付与することなども検討している。ただし、この問題については、認識の確実性を高めようとする中で、操作の自然さが損なわれることがあり、慎重に判断する必要がある。

センサデバイスで認識される座標については、まだユーザの思い通りに行かない場合があるが、本システムはセンサデバイスの置き換えにも柔軟に対応できるようになっているため、今後、センサデバイスの高精度化により解決することを期待している。

5. 結言

ジェスチャ入力を用いて汎用的に機器制御を行うためのシステムを、Android 端末を用いて実装した。実装したシステムは、2種類のセンサデバイスで同様に動作することを確認した。実験協力者からは、各種設定は容易であるという意見を得られた一方、フィードバックがわかりづらい、ジェスチャ認識開始がわかりづらいという意見も得られた。今後は、今回得られた意見をもとにシステムの改良を行い、多くの被験者を対象にして評価を行うことを考えている。また、マイコンから出力される制御信号の電気的変換を考慮し、既存の機器を実際に制

御することを検討している。

参考文献

- 1) 岩崎健太, 村山大策: 京都市産業技術研究所研究報告, No3, p.110 (2013).
- 2) <http://www.microsoft.com/en-us/kinectforwindows/> (2014年5月30日).
- 3) <https://www.leapmotion.com/> (2014年5月30日).
- 4) J. O. Wobbrock, A. D. Wilson, Y. Li: Proceedings of the 20th annual ACM symposium on User interface software and technology, p.159, (2007).