

イメージセンシングを用いたリアルな機器制御の検討

齊藤 公博*

Study of realistic equipment control using image sensing technology

Kimihito SAITO*

The interface transfer method based on the gesture recognition is described. The method means the concept of connection between an existing control device and a gesture recognition hardware. The combination of an existing well-designed interface and a hands-free system allows us to suffer convenient and simple life. As an example of existing device, we introduce a drone as a control target with the LeapMotion as gesture recognition device. We also investigated the realistic three-dimensional image capture for making model of a control target in a virtual world by Unity. The advantages and outlook of the methods are discussed.

Keyword Image sensing, Equipment control, Gesture recognition, Interface transfer method

1. 本研究の背景と目的

テクノロジーの発展により、従来は人が不可能であったこと、大きな労力を必要とし実現できなかったことが可能になってきた。遠く離れた場所への移動手段や通信手段、巨大な複雑なものから原子サイズに迫る小さなものを加工する手段、見ることや聞くことのできない情報を得る手段、また膨大な計算や判断を必要とする情報処理まで、将来人々が手にできる能力はこれからも増えてゆくと考えられる。それに伴い、その手段・機能を人がどのように取得し操るかというインターフェイス技術の発展も大事であると思われる[1,2,3]。

著者らは、このような機器と人とのインターフェイスを、触れる必要のない自然な動作の認識（ジェスチャー認識）により置き換えてゆくことができないかという問いを基に、インターフェイス変換という考え方で実現してゆく検討を行っている[4,5,6]。

近年のカメラ技術の進展は著しく、性能面では過去のフィルの解像度や感度を超えるセンサーや、kHzを超えるフレームレートが実現され、大きさや価格面では、小さなレンズ付きセンサーを複数搭載したスマートフォンが普及してきているという状況である。この技術を最大限に用い、すでに実現されている機器のヒューマンインターフェイスの機能をジェスチャー認識に置き換えることから進めるという考え方をインターフェイス変換と呼んでいる。

昨年までの紀要では、対戦型ゲームのジェスチャー制御から始まり、ロボットアーム、ラジコンヘリ、マウス、人型ロボットの制御を研究室内でやってきた。大きな課題はジェスチャー認識アルゴリズムの認識率向上と、制御対象の機能の意味（意義）づけである。今回の研究では、まずジェスチャー認識アルゴリズムに関して、製品化されている LeapMotion [7] を用い、現在の認識技術のレベルを把握することを目指した。また、制御機能の意味づけとして、操縦が安定でよりリアルな制御が可能となるドローンを対象とするもの、および、Unity により実現する仮想空間内に実際の人物の3D キャプチャーを用いたアバター [8] を配置することでリアリティを実現する検討を行った。

2. ジェスチャーを用いたドローン制御

第一の実験は、従来同様の領域積分を用いたカメラ画像でのジェスチャー認識、および LeapMotion を用いたドローンの制御である。

前回のラジコンヘリの制御では、ジェスチャーによる動作自体は確認できるものの、ヘリコプター自体の制定性が十分でないために操縦者の思うような動きがなかなか実現できないという課題があった。近年は、小型で比較的安価であっても、センサーを用いて制定性の良いホバーリングを行えるドローンがあり、今回の制御対象にすることにした。

ジェスチャーの入力に関しては、従来通りのカメラ入力

*近畿大学工業高等専門学校
総合システム工学科 電気電子コース

から素朴な画像処理を行って制御対象へ信号を送るもののほかに、LeapMotion を用いて左右の手によるジェスチャーを認識して制御対象へ送る 2 つの方法を検討した。

2. 1 インターフェイス変換用ハードウェアの作製

制御対象として ドローン: Parrot MANBO FLY PF27078 とコントローラ: Parrot Flypad MiniDrone コントローラを用いた。

コントローラの機能としては、左右のスティックで上下、前後、左右、左右旋回の 8 方向(+静止)の制御ができるようになっている。それぞれ何段階かのスピードが可能のようであるが、今回は可変抵抗のスティックへ並列にフォトカプラを接続して、各 8 つの機能をデジタルで動作させるように改造した。前回ヘリコプターの制御では、特に上下動作に対してアナログの制御が必要(ホバーリングが不安定であったため)でアナログのフォトカプラを用いたが、今回はデジタル制御で十分安定な動作が可能であった。

フォトカプラは、Arduino の DIO により ON/OFF を行い、Arduino は PC から USB で接続されている。システムの構成と接続基板を図 1, 2 に示した。

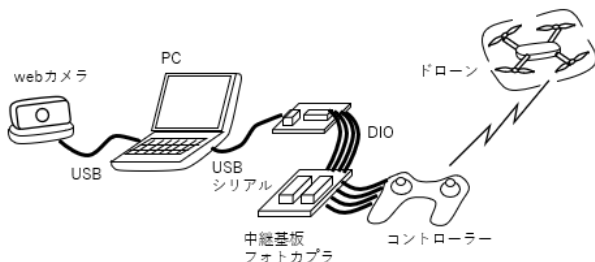


図1 システム構成図

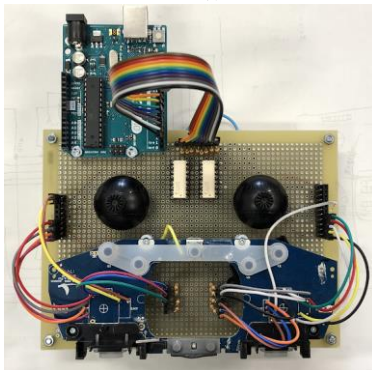


図2 コントローラへの接続基板

PC から Arduino へ USB を介して、各 DIO に対応する文字を送信して制御を行う。Arduino にはシリアル入力による文字に対応したデジタル出力を行うようプログラムが書き込まれている。

2. 2 カメラ画像処理を用いたジェスチャー認識制御

ジェスチャー認識として 2 つの方法を検討したが、一つは従来通りの web カメラ等の画像処理による、領域積分で

ON/OFF を行う方法である。背景の緩やかな変化による誤作動を防ぐため、時間微分画像の絶対値を領域を区切って積分するようにした。この場合には変化している時間のみ ON となるため、0.2sec 程度のホールド時間を設けることである程度自然な動作を可能にすることができた。PC 上では Python をベースとし、OpenCV によるカメラ入力と画像処理、シリアル出力による前節の Arduino への信号送信を行う構成である [9,10]。

地元イベント、および本校学園祭でデモを行ったが、少し慣れてくるとかなり思うような動作をさせることが可能である。



図3 ジェスチャーによるドローン制御のデモ

2. 3 LeapMotion によるドローンの制御

LeapMotion (LeapMotion 社) は 2 個の赤外線カメラと赤外線 LED を用いて左右の手指の位置を判別できる小型 (5cm x 3cm 程度) のセンサーである。C 言語や Python などによるプログラミングが可能であるが、今回は websocket を通して両手の位置を取得する方法で認識を行った。Python のプログラムでは、

```
ws = create_connection("ws://localhost:6437");  
res = ws.recv();
```

とすることで ast 形式が取り込まれ、左右の手の 3 次元の位置がリアルタイムで取得できる。



図4 LeapMotion による制御の様子

左右の手の領域を、ドローンの8つの動作方向に対応させ、検出時に Arduino へ文字としてシリアル通信で送信する。

実際に動作させている様子を図4に示した。前述の素朴な画像処理に比べて背景除去などの必要もなく、連続した高精度な制御が可能であった。

3. Unity を用いたリアルな対象制御の検討

前回までに、カメラ画像処理によるジェスチャー認識を用いて、Playstation および Unity [11] での仮想的な人物の制御の検討を行ってきたが、今回は制御対象として実際の人物の3D キャプチャー像を配置することで、より現実感があるインターフェイスのモデルを構成することを試みた。

前回と同様、ジェスチャー入力にはカメラ画像を用いた素朴な領域積分とし、UDP にて Unity へ動作を送信、Unity 内で対象となるオブジェクト（キャラクター）を動かすことを目標とした。

3. 1 Structure Sensor による 3D 人物画像の取得

実際の人物の3D キャプチャーは、Occipital 社、iPad 用赤外線3Dスキャナー Structure Sensorを用いた [12]。手を広げて直立した状態で、360deg 全身のキャプチャーを行う。その後、itSeez3D アプリ [13] にて 3D データに変換する。



図5 3D キャプチャー画像(Maya 読み込み)

次に、アニメーション作成と Unity 取り込みのために変換(Meshlab など、ply ファイルから)、あるいは、itSeez3D での取り込み時に FBX ファイルを作成する。

3. 2 Maya による 3D アニメーションの作成

Maya [14] はオートデスク社による3次元コンピュータグラフィックスソフトウェアであり、業界等(Blender 等と並び)でも広く用いられている。

FBX 形式の3D データを Maya で取り込み、アニメーションを作成する。リグ付け(骨格)を行い、いくつかのキーフレームの動作を作成し、補間によってアニメーション

を作成する。ジェスチャー認識で行いたい動作をいくつか作成したのち、Unity で取り込む FBX ファイルとしてセーブする。

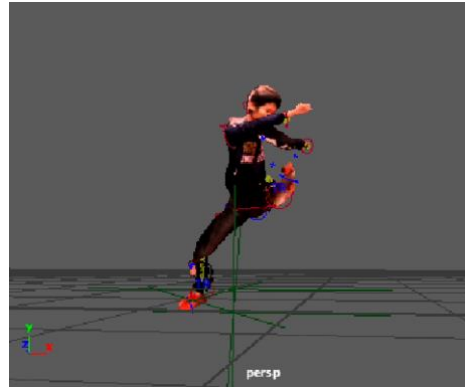


図6 Maya によるアニメーション作成例(一部)

Unity 上で Asset として取り込み、オブジェクトのスク립トを編集して、外部からの要求で動作をさせる。図7に、実際にジェスチャーにより Unity 上のアニメーション動作をさせている図を示す。



図7 ジェスチャーにより Unity 上の3D キャラクターを動かす

4. まとめと今後の展望

人と触れる必要のない、装置との自然なインターフェイスを目標に、撮像デバイスを用いた既存のインターフェイスとの変換という考え方を基にした検討を進めている。

今回の報告では、(1) LeapMotion を用いたドローンのジェスチャー制御、(2) カメラ画像を用いた Unity 上でのリアルな3D キャラクター制御を試みた。

近年のドローン自体の性能(制定性)の向上もあり、カメラ画像での意図した制御も可能になってきた。また、LeapMotion を用いたドローンの制御は、より高い精度で可能であることが分かった。既存のジェスチャー入力装置の性能向上を実感するとともに、本研究の考え方の実現性も少しずつ実感している。例えば Tello-python[15]のように、python 言語で直接 wifi によりドローンを制御することができるような製品もあるが、ドローンの特化した認識装置を用いずに既存の安価なイメージセンサで高精度・高機

能な認識を実現する方法も考えてゆきたい。

Unity 上でのリアルな制御対象として、3D 人物キャプチャーを用いる準備が出来た。ジェスチャー入力とリアルな人物動作の対応を検討し、インターフェイスとしての実用性を高めてゆく手段やモチベーションとしてさらに進めてゆく。

本成果を報告するにあたり、令和元年度当研究室の専攻科・竹岡寿明君、卒業研究生・梅田草太君、杉山貴哉君、富永隆功君、西園京太君、および矢田康明君に感謝致します。

参考文献

- 1) 越後 他、IT Text 人画像処理、オーム社 2007
- 2) 畠 他、動き情報と情報圧縮を用いたロバストなジェスチャー認識手法、電子情報通信学会論文誌 D II , J81-II, 9, pp1983-1992 (1998)
- 3) T. Mashita, et. al., "Pointing Gesture Recognition and Indicated Object Detection," Proc. 10th Intl. Conf. Human Computer Interaction (HCI2003), 2, pp.731-735 (2003)
- 4) 齊藤、平成 29 年度近畿大学工業高等専門学校紀要
- 5) 齊藤、平成 30 年度近畿大学工業高等専門学校紀要
- 6) S. Kobayashi, "Optical Gesture Recognition System," Siggraph 97
- 7) <https://www.leapmotion.com/ja/>
- 8) E. Castronova, "On Virtual Economies," CESifo Working Paper Series No. 752
- 9) Python, Numpy, Scipy マニュアル等
- 10) OpenCV マニュアル: <http://opencv.jp/opencv-2svn/c/> など
- 11) <https://unity3d.com/jp>
- 12) <https://structure.io/>
- 13) <https://itseez3d.com/media.html>
- 14) <https://www.autodesk.co.jp/products/maya/overview>
- 15) <https://www.ryzerobotics.com/jp/>