

機械学習を用いたジェスチャー認識精度向上方法の研究

吉村拓哉[†] 橋本剛^{††}

近年、身体の動きを認識して遊ぶゲームが増加している。任天堂の Wii やマイクロソフトの Xbox 向けデバイスである Kinect を用いたゲームがそれである。すでに多くのゲームが、企業や個人によって開発されている。例えば、誰もがかめはめ波を撃てるようになる、“Kinect-kamehameha”と呼ばれるゲームがある。しかしそれらのゲームでは、ジェスチャーの認識精度が低いものがある。本研究では、機械学習を“Kinect-kamehameha”に適用し、認識精度の向上を目指す。また、Kinect を用いるゲームにおける SVM の有用性を示す。

A study of improve gesture recognition accuracy using machine learning

YOSHIMURA TAKUYA[†] and HASHIMOTO TSUYOSHI^{††}

In recent years, the game which recognizes a motion of the body is increasing. The game using Kinect which is a device for Xbox of Wii of Nintendo or Microsoft is it. Many games are already developed by the company and the individual. For example, there is a game called “Kinect-kamehameha”. Anyone will be able to shoot Kamehameha. however, the gesture recognition accuracy is low in some games. In this research, machine learning is applied and improvement in recognition accuracy aimed.

1. はじめに

近年、マイクロソフトから発売された Xbox ゲーム用コントローラ Kinect が注目を浴びている。Kinect はデバイスを手に持つことなく人間の身体や音声を認識でき、ゲームコントローラとして利用できる。対象物の認識は、Kinect 前部に取り付けられたレンズのひとつからレーザのパターンを広範囲に照射することで実現している。また、膨大な量の学習を行うことで人間の手や腕、肩、脚などの身体各部分や関節の座標を高い精度で取得¹⁾できる。Kinect を使うプログラムはすでに多数考案されている。しかし、すでに販売されている Kinect のゲームソフトや、個人により開発されているソフトウェアでジェスチャー認識の部分に問題があることも多い。例として Kinect-Kamehameha²⁾について考える。これは、Kinect を用いて「誰でもかめはめ波を撃つ体験ができる」ソフトでありソースも公開されている。かめはめ波を撃つためには特定の

ジェスチャーを行う必要があるが、このときのジェスチャー認識精度が低いため「かめはめ波を撃つ動きをしてもかめはめ波が撃てない」ことがある。

そこで本研究では、ゲームにおけるジェスチャー認識精度の向上を目指す。ジェスチャー認識の精度を向上させるための研究³⁾は多いが、今回は画像認識の分野で広く用いられている、SVM⁴⁾(Support Vector Machine) を用いる。SVM は教師あり学習を用いる識別手法のひとつであり、動作認識の為に研究⁵⁾に用いられている。Kinect-Kamehameha を題材として、SVM を用いたジェスチャー認識精度向上の方法を論じる。

2. Kinect-kamehameha

Kinect-kamehameha とは、鷲尾友人氏によって作成されたフリーソフトウェアであり、ソースが公開されている。鷲尾氏のプレゼンテーション⁶⁾において、「世界初のしょうもないものへの挑戦」としている。しかしながら、同プレゼンにあるように大きな反響があり、公表した動画には最初の一週間で 30 万回もの再生があったという。また、多くのコンテストで受賞している作品でもある。Kinect-kamehameha では、多くの人が子供の頃夢見たかめはめ波を撃つことができる。ただかめはめ波がでるだけではなく、髪が金髪に

[†] 松江工業高等専門学校専攻科 電子情報システム工学専攻
Advanced Engineering Faculty of Electronic and Information Systems, Matsue College of Technology

^{††} 松江工業高等専門学校 情報工学科
Department of Information Engineering, Matsue College of Technology

なり身体のまわりには金色のもやが見える。また、溜めを大きくすることで極大のかめはめ波を放つこともできる。このように Kinect-kamehameha では、まるで人気漫画の主人公のようにかめはめ波を撃つことができる。

3. ジェスチャー認識における問題点と解決策

Kinect によるジェスチャー認識は、開発用ライブラリである OpenNI に用意されている関数⁷⁾を用いて行える。ここで、OpenNI と PrimeSense のリリースしている NITE ライブラリをまとめて OpenNI と記述する。OpenNI では、表 2 に示す動きを認識することができる。

表 1 認識できる動き	
OpenNI	NITE
腕を前後に動かす動き	前後への動き
腕を左右に振る動き	左右への動き
腕を上げる動き	上げる動き
腕を動かす動き	下げる動き
	左への動き
	右への動き
	静止

一方、マイクロソフトのリリースした KinectSDK⁸⁾では、ジェスチャー認識用の関数は用意されておらず、独自に定義する必要がある。複雑な動きを識別するには、Kinect より取得した各関節の座標データをもとに、ジェスチャーを定義する必要がある。Kinect から取得できる各関節座標のイメージを図 1 に示す。

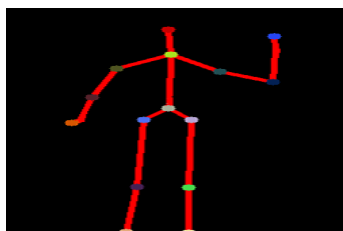


図 1 取得するイメージ

また、Kinect から取得する各項目を表 2 に示す。各項目で 3 次元座標データ (x,y,z) を取得できる。

ここで、例としてかめはめ波のジェスチャーについて考える。かめはめ波は「両手の距離が十分に近いか」や「腕がまっすぐのびているか」の条件により判定を行う。この際に、Kinect より取得した各関節の 3 次元座標データを判定用のパラメータとする。図 2 は、実際に Kinect-kamehameha をプレイした際の、かめは

表 2 取得できる各項目

上半身	下半身
HEAD	SPINE
SHOULDER-CENTER	HIP-CENTER
SHOULDER-LEFT	HIP-LEFT
SHOULDER-RIGHT	HIP-RIGHT
ELBOW-LEFT	KNEE-LEFT
ELBOW-RIGHT	KNEE-RIGHT
HEWRIST-LEFT	ANKLE-LEFT
HEWRIST-RIGHT	ANKLE-RIGHT
HAND-LEFT	FOOT-LEFT
HAND-RIGHT	FOOT-RIGHT

め波認識成否例である。(a) が失敗例、(b) が成功例である。

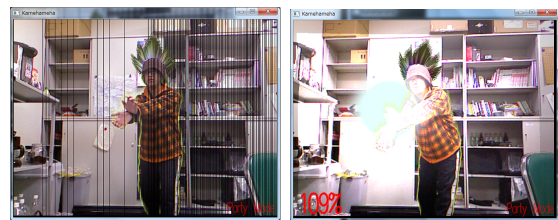


図 2 かめはめ波認識の成否例

かめはめ波を撃つ人間にとっては (a), (b) どちらも同じジェスチャーである。しかし、ジェスチャー判定に使われているパラメータと、実際に測定したパラメータの誤差によって認識失敗になる。ユーザとしては、だいたい同じジェスチャーを行えば認識成功としたい。そこで、ジェスチャー認識に用いるパラメータ部分で機械学習を行うことで、ジェスチャー認識の成功するパラメータ群、失敗するパラメータ群をマージン化し認識の精度を向上させる。

4. 機械学習

機械学習には、SVM と呼ばれるパターン認識手法を用いる。SVM は教師あり学習を用いる識別手法のひとつでもある。Kinect が各関節における座標データを取得するための学習にも使われており、画像認識の分野で広く使用されている。SVM の特徴はマージン最大化である。SVM では、学習データの中で最も他クラスと近い位置にいるものを基準として、そのユークリッド距離が最も大きくなる位置に識別境界を設定する。その学習結果に対して、問題となるデータを検証しどのクラスに分類されるかを判定する。今回は多種ある SVM ツールのうち libsvm⁹⁾を用いて学習を行う。

5. 学習方法

ジェスチャーの認識精度を向上させるために、まず学習に必要な教師データを作成する。なお、Kinectから取得した各関節座標データのうち、上半身のデータを学習データとして保存する。一定期間行うジェスチャーを無条件に正例であると定義し、正しくジェスチャーを行う。同様に、一定期間行うジェスチャーを無条件に負例であると定義し、わざと間違ったジェスチャーを行い保存する。この正例と負例について libsvm で学習を行い、元となる教師データを作成する。その後、判定したいジェスチャーを行い問題データとして保存する。問題データと教師データを libsvm にかけて、それが正例か負例かを識別する。正例であれば、教師データのうち正例データとして保存し教師データを更新する。負例についても同様に保存し更新する。以上を繰り返し、識別の精度を向上させる。学習データは、正例と負例でデータ数が同じとなるよう調整した。データ数は、正例が 20 回分、負例が 20 回分である。

6. 実験と考察

今回の実験では、学習によって特定のジェスチャーを正しく判別できるか否かを確認した。なお、実験は以下の環境で行った。

- OS: Windows 7 Enterprise
- デバイス: Kinect
- ライブラリ: Kinect for Windows SDK, libsvm

学習に必要な教師データを作成し、それが正しく判別できているか、問題となるデータを作成し実験を行った。正例と同じ動きをジェスチャー 1、負例と同じ動きをジェスチャー 2 とした。各ジェスチャーについて以下に示す。図 3 について、(a) の形からジェスチャーを開始し、(b) が判別成功となるジェスチャーである。図 4～6 について、(a) の形からジェスチャーを開始し、(b) が判別失敗となるジェスチャーである。

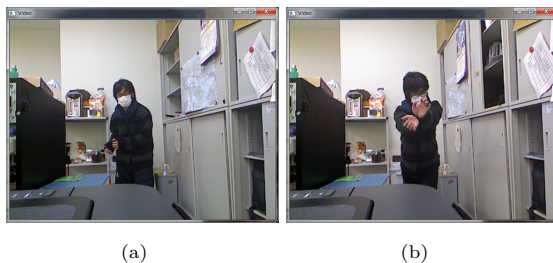


図 3 判別成功となるジェスチャー



図 4 判別失敗となるジェスチャー 1



図 5 判別失敗となるジェスチャー 2

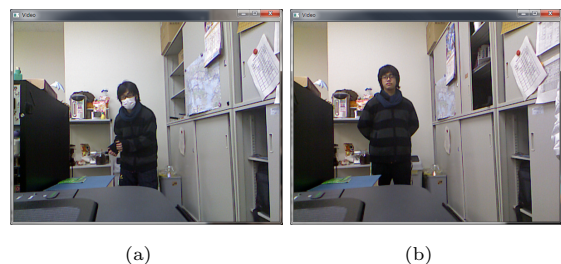


図 6 判別失敗となるジェスチャー 3

実験を行った結果、図 3 は正しいジェスチャーである、図 4～6 は異なったジェスチャーである、と判別できた。よって、今回の実験から、学習によってジェスチャーの判別ができるか否かを確認することができた。しかしながら、今回の実験では学習に使用したデータ数が少なく、またテストの回数も少ないため、更なる実験と学習を繰り返しテストの精度を確認する必要がある。また、実験では同一人物による教師データと問題データの作成を行ったため、複数の人間によってデータを作成し、どのような差異がでるかテストする必要がある。

7. おわりに

今回は, Kinect により取得した座標データと, libsvm を用いて教師あり学習を行った. その結果, 認識したい特定のジェスチャーと, 明確に違うジェスチャーの判別に成功し, SVM における学習方法の基礎と KinectSDK についての挙動を把握することができた. 今後, ジェスチャー認識の精度向上をはかるために, 現在フリーで配布されている DTW¹⁰⁾ と呼ばれるライブラリを用いる. 2次元ベクトルを用いてユーザが行う特定のジェスチャーを識別できるようになるため, これに用いられている手法も参考に Kinect-kamehameha を改良し, より精度の高いものを作成する. また, 精度が何%単位で向上したかを数値化することで, Kinect を用いるゲームにおける有効性を検証していく.

参 考 文 献

- 1) J.Shotton, A.Fitzgibbon, M.Cook, T.Sharp, M.Finocchio, R.Moore, A.Kipman, A.Blake : *Real-Time Human Pose Recognition inParts from Single Depth Images*, CVPR,(2011)
 - 2) 鷺尾 友人 : *kinect-kamehameha kamehameha by Kinect*,
<http://www.slideboom.com/presentations/377555/Kinect-Hacks-for-Dummies-in-Japanese/>,(2011)
 - 3) Louis-Philippe Morency, Ariadna Quattoni, Trevor Darrell : *Latent-Dynamic Discriminative Models for Continuous Gesture Recognition*, CVPR,(2007)
 - 4) V. Vapnik : *The Nature of Statistical Learning Theory*, Springer,(1995)
 - 5) 松永武文, 尾下真樹 : *SVM を用いた動作認識の為の動作状態の自動推定*, 情報処理学会研究報告 グラフィックスと CAD, Vol. 2008, No.109, pp.31-36, 飯塚,(2008)
 - 6) 鷺尾 友人 : *KinectHacksForDummiesja*,
<http://code.google.com/p/kinect-kamehameha/>,(2011)
 - 7) 中村 薫 : *KINECT センサープログラミング*, pp.54-210,(2011)
 - 8) Microsoft : *Kinect SDK for Developers*,
<http://www.microsoft.com/en-us/kinectforwindows/>,(2011)
 - 9) R.E. Fan, P.H. Chen, and C.J. Lin : *Working set selection using second order information for training SVM*, J. Machine Learning Research, vol.6, pp.1889-1918,(2005)
 - 10) Steve : *Kinect SDK Dynamic Time Warping Gesture Recognition*, <http://kinectdtw.codeplex.com/>,(2011)
-