

深度センサを用いたユーザの向きの変化に頑健な ジェスチャ認識

都竹尚紀^{†1} 梅澤猛^{†1} 大澤範高^{†1}

深度センサを用いた徒手によるジェスチャ認識は、デバイスを保持または装着する煩雑さがなく、仮想空間でのインタラクションや家電操作インタフェースとして有用である。しかし、一般に深度センサは正対しての使用を想定しており、それ以外の角度では認識率が低下する傾向がある。センサ正面とユーザの向きの角度に応じて座標変換を行う手法も提案されているが、角度によっては認識率が大きく低下するという課題が残る。そこで本稿では、Dynamic Time Warping 法によるジェスチャ認識において、角度に応じた個別のジェスチャテンプレートを登録することで認識率を向上させる手法を提案する。様々な角度でのジェスチャ認識実験を行い、提案手法が認識率の向上に有効であることを確認した。

Improving Gesture Recognition with Depth Sensor by Considering Direction of User

NAOKI TSUTAKE^{†1} TAKESHI UMEZAWA^{†1}
NORITAKA OSAWA^{†1}

In interfaces of electrical appliances and interactions in a virtual reality environment, hand gesture recognition based on information of image and/or depth sensors outside a human body is useful because it does not require a user to hold or to wear a device. However, gesture recognition fails sometimes when body movement of a user is not fully tracked. Although a method was proposed which transforms coordinates of tracking data based on an angle between the user and a face of the sensor, a recognition rate depends on the angle. This paper proposes a method that uses multiple gesture templates for different angles to improve gesture recognition rate in Dynamic Time Warping technique. An experiment shows that the proposed method improves rates of gesture recognition.

1. はじめに

キーボード、マウス、リモコンなどの様々な入力デバイスが存在するが、仮想空間 (VR) での操作には不適な場合がある。また、電気機器を操作する度にデバイスを手に取るのは煩雑である。それに対し、ジェスチャは身体動作を基にしており、より自然な動きで操作が可能である。加速度センサなどを身体に装着、保持していなければならないことはユーザの負担となる。深度センサによって徒手によるジェスチャ認識が可能であり、Kinect[1]などの安価な深度センサの登場により多くの場面で徒手によるジェスチャ認識を利用可能になっている。KinectはMicrosoft社のデバイスでRGBカメラに加え、深度センサが搭載されており、Kinect用のソフトウェア開発ツールキットであるKinect for Windows SDK[2]を用いることで、図1の身体20ヶ所部位の三次元位置情報を取得することが可能である。この身体部位はビッグデータからの機械学習により推定されている。しかしKinectはユーザが正面を向いていることを想定しているため、ユーザがKinectに対し横や後ろを向いた際に認識がうまく出来ない問題がある。図2がその様子を示して

おり、横向きの際は身体の奥側がオクルージョンによりずれて認識されてしまう。また後ろ向きの場合、右手が反対に認識されてしまっていることがわかる。

センサが正対していない様々なケースを次に示す。

- ・ ラップトップ PC の前にセンサを配置して人の動きを認識する場合、モニターを見るのをセンサが妨害してしまうので脇に設置する
- ・ ヘッドマウントディスプレイ (HMD) を利用した VR インタラクションにおいて、ユーザは VR 環境の映像に即して向きを変えるため、ユーザがセンサに対して正面を向いているとは限らない
- ・ 家電をジェスチャで操作するインテリジェントルームへの利用を考えた場合、ジェスチャは家電に向けて行うため、取り付けられたセンサと向きが一致するとは限らない

このようにセンサが正面にない様々なケースが考えられる。本稿ではセンサによって横向きや後ろ向きを認識するためのシステムについて検討した。

^{†1} 千葉大学大学院融合科学研究科
Graduate School of Advanced Integration Science, Chiba University

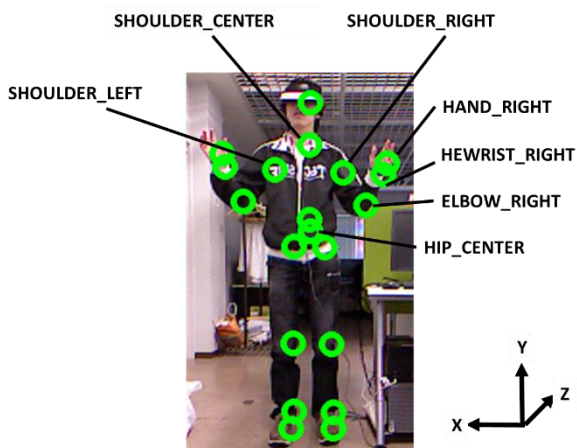


図 1 Kinect による骨格認識
Figure 1 Skelton tracking by Kinect.

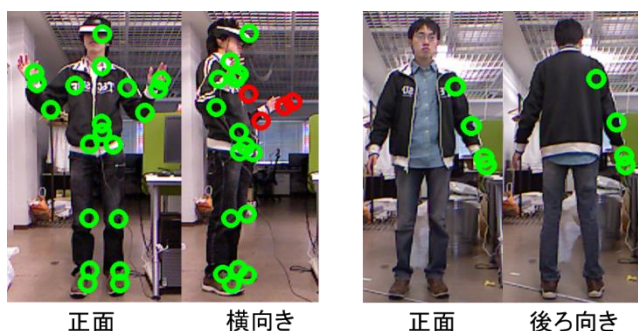


図 2 横, 後ろ向きにおける誤認識
Figure 2 Wrong recognition in side and back.

2. 関連研究

深度センサに対してユーザが正対していない場合でのジェスチャについては、いくつかの研究がなされている。長谷川ら[3]は上半身の部位の位置情報からユーザの向きを取得し、その角度より座標変換を行う手法を提案した。しかし、横向きの場合は奥の肩のオクルージョンによりユーザとセンサとの角度が大きくなるほど、正しいデータを取得することができない。武田ら[4]は横向きのジェスチャを手の深度情報から取得しているが、動的なユーザの方向の変化については言及していない。

今までの我々の研究[5]では長谷川ら[3]の手法を改良し、Kinect で得た手前の肩と肩の中心(SHOULDER_CENTER), 重心 (HIP_CENTER, 原点) の座標を用いて角度を取得する手法を提案し、斜め方向, 横方向の認識率の向上を行ったが, 正面方向の認識率が低くなってしまった。

本稿では深度センサとして Kinect を用いて, 我々の既存の手法[5]をさらに改良し, 高精度のジェスチャ認識を可能とする手法を提案し, 評価する。

3. 提案手法

3.1 向き取得の改良

2 節で述べたとおり, 今までの我々の手法[5]では手前の肩と肩の中心を使った場合, 斜め, 横方向の認識率は向上した。しかし, 手前の肩のみを利用するために正面方向では頻繁に手前の肩が左肩と右肩で入れ替わってしまい, 角度がずれてしまう。そのため, 正面方向での認識率は長谷川らの両肩を用いる手法[3]に比べ低下した。そこで, 今までの我々の斜め, 横方向のジェスチャを認識できる手法[3]と長谷川らの正面から斜め方向のジェスチャを認識できる手法[5]のそれぞれの利点を活かすため, 両方の手法でそれぞれジェスチャ認識を行い, 最終的に出力するジェスチャを次のように決定する。

・両方の手法で認識した場合

同一のジェスチャである場合はそのジェスチャを出力し, 異なるジェスチャを認識した場合は, 誤認識を防ぐためジェスチャと認識しない。

・片方の手法のみが認識した場合

認識したジェスチャを出力するジェスチャとする。

・両方共認識なかった場合

ジェスチャを認識しない。

これにより認識率を向上させるとともに, あいまいなジェスチャに対しての誤認識を防ぐことが出来ると考える。

次に, 角度の求め方と座標変換について述べる。ここでは座標変換のために, 各座標において図 1 の HIP_CENTER を原点としたユーザ座標系を作り, 相対座標に変換した後, 座標変換を行う。x軸, y軸, z軸はそれぞれ図 1 に示されている Kinect for Windows SDK の座標系を踏襲し, 右手系としている。

・長谷川らの両肩を用いて角度を取得する手法[3]

原点の座標を引いて相対座標とした左肩(SHOULDER_LEFT)の座標を $L = (L_x, L_y, L_z)$, 右肩(SHOULDER_RIGHT)の座標を $R = (R_x, R_y, R_z)$ としたとき, ユーザの向き θ は,

$$\theta = \tan^{-1} \frac{L_y R_z - L_z R_y}{L_x R_y - L_y R_x}$$

で表すことができる。

・手前の肩と肩の中心を用いて角度を取得する手法[5]

原点の座標を引いて相対座標とした左肩の座標を $L = (L_x, L_y, L_z)$, 右肩の座標を $R = (R_x, R_y, R_z)$, 肩の中心を $C = (C_x, C_y, C_z)$ としたとき, ユーザの向き θ は,

$$\theta = \tan^{-1} \frac{L_y C_z - L_z C_y}{L_x C_y - L_y C_x} \quad (L_z \leq R_z)$$

$$\theta = \tan^{-1} \frac{R_y C_z - R_z C_y}{R_x C_y - R_y C_x} \quad (L_z > R_z)$$

で表すことができる。

その後、取得した角度を用いて、変換前の座標を ${}^t(x \ y \ z)$ 、変換後の座標を ${}^t(x' \ y' \ z')$ としたとき、座標変換は、

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

で得られる。

また、ユーザが横を向いた場合、Kinect はユーザの身体と接触した箇所を部位と認識してしまうため、図 3(b)のように本来原点であるべき点と異なる点を原点と認識してしまう問題がある。その問題に対処するため、我々はユーザを半径 0.2m の円柱と考え、原点をユーザの向きに合わせて変換を行うことで、原点のずれを低減する。

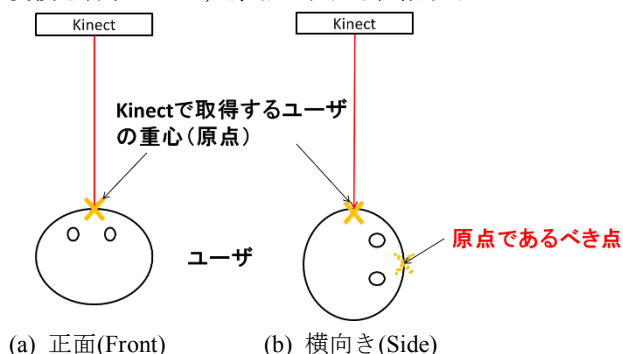


図 3 座標変換の問題点

Figure 3 Problem of coordinate transformation.

3.2 ジェスチャ認識

ジェスチャ認識部には Dynamic Time Warping(DTW)法[6]を利用する。これは2つの波形の相違度を測定する手法であり、あらかじめジェスチャテンプレートを作成し、そのテンプレートとの相違度が閾値を下回った場合ジェスチャと認識する。ジェスチャの特徴量には右手、右手首、右肘の三次元位置情報を利用し、長さは30フレームとしている。

閾値の取得方法は村田ら[7]の手法を利用する。複数人で同一のジェスチャを行い、それぞれのジェスチャ間でDTW値を測定する。その後、中間となるジェスチャをテンプレートとして取り出し、そのジェスチャからDTW値が最も大きいジェスチャとの差を閾値とする。今回、学習・識別プログラムにはオープンソースのDTW[8]をC++に改良して利用した。

3.3 向きに合わせたジェスチャテンプレートの作成

従来手法[3][5]ではKinectに対して正面のデータをテンプレートとして作成していたが、正面のデータだけでは向

きによる座標のずれに対応することは難しい。そこであらかじめ左斜め、右斜め方向のジェスチャテンプレートを作成することで、向きによるジェスチャのずれに対応する。

座標変換を行いやすくするため、図4のように角度を設定し、図中の矢印の方向へのジェスチャテンプレートを作成する。それぞれを左から90° template, 45° template, 0° template, -45° template, -90° templateとする。

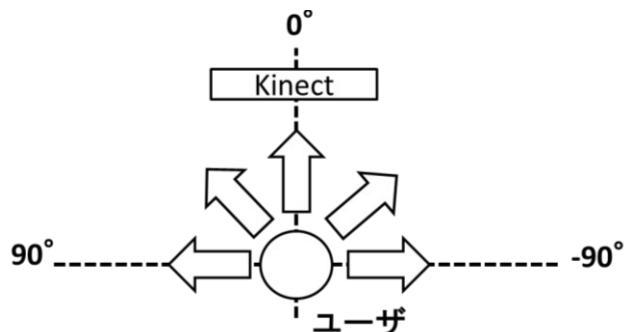


図 4 ユーザの角度設定

Figure 4 Setting of the user's direction.

4. 実験

4.1 想定環境

3人の被験者で、図4の5種のジェスチャを3.3で述べた5方向のテンプレートでそれぞれ作成する。その後0°，±45°，±90°でジェスチャを4回ずつ試し、それぞれのテンプレートでの認識回数を測定した。

ジェスチャは長谷川らの研究[3]で個人差の少ないジェスチャとして挙げられたものである。

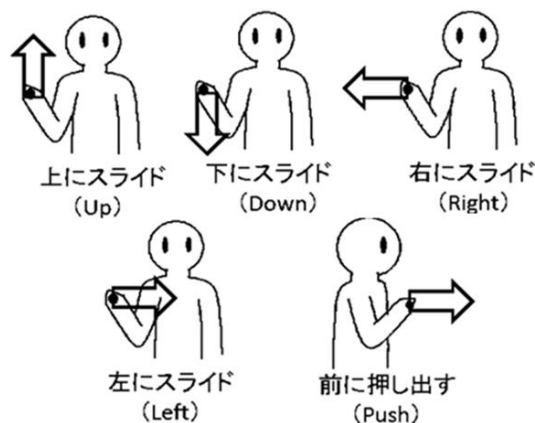


図 5 ジェスチャー一覧

Figure 5 Using gestures for experiment.

4.2 結果

Up, Down, Right, Left, Push, 全ジェスチャの認識率をそれぞれ表1の(a)~(f)に示す。縦軸がテンプレートで横軸がユーザとKinectのなす角度、数値はジェスチャ認識率のパーセンテージである。

表 1 ジェスチャの認識率

[縦軸：利用したジェスチャテンプレート，
横軸：ユーザと Kinect がなす角度，数値：認識率]

Table 1 Recognition rate of gestures.

(a) Up					
Template\Angle	90°	45°	0°	-45°	-90°
90° template	42	100	83	92	100
45° template	67	100	100	100	100
0° template	42	100	100	100	100
-45° template	25	100	75	100	100
-90° template	0	42	67	92	100

(b) Down					
Template\Angle	90°	45°	0°	-45°	-90°
90° template	58	50	100	100	100
45° template	67	100	100	0	100
0° template	0	92	100	58	100
-45° template	17	33	58	100	100
-90° template	8	33	58	50	100

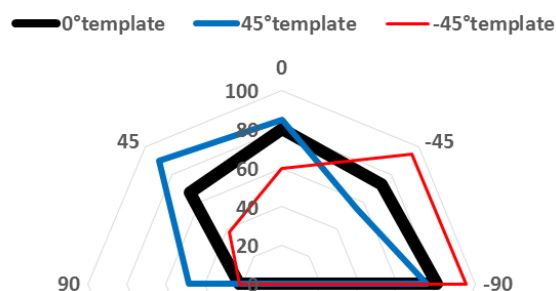
(c) Right					
Template\Angle	90°	45°	0°	-45°	-90°
90° template	17	17	25	42	67
45° template	75	100	100	92	83
0° template	58	67	100	100	100
-45° template	67	17	50	92	100
-90° template	8	0	8	75	100

(d) Left					
Template\Angle	90°	45°	0°	-45°	-90°
90° template	58	42	67	100	92
45° template	33	92	92	83	92
0° template	8	75	92	100	92
-45° template	0	42	75	92	92
-90° template	0	33	83	100	100

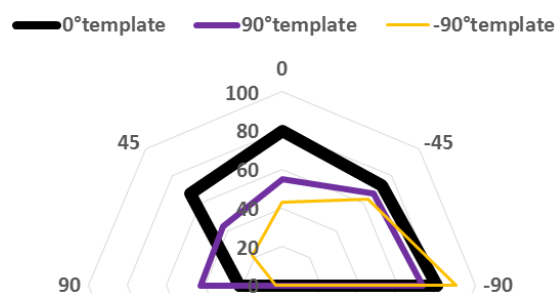
(e) Push					
Template\Angle	90°	45°	0°	-45°	-90°
90° template	33	8	0	0	8
45° template	0	58	33	0	0
0° template	0	0	8	8	8
-45° template	0	0	42	92	83
-90° template	0	0	0	0	50

(f) 全ジェスチャ					
Template\Angle	90°	45°	0°	-45°	-90°
90° template	42	43	55	67	73
45° template	48	90	85	55	75
0° template	22	67	80	73	80
-45° template	22	38	60	95	95
-90° template	3	22	43	63	90

また，全体のジェスチャにおいて，正面方向と斜め方向との認識率，正面方向と横方向との認識率をレーダ図で表したものを図 6(a)，(b)に示す．レーダの外周が角度で，内側の数値が認識率(%)である．



(a) 0° template と ±45° template の比較
(Comparison 0° template and ±45° templates)



(b) 0° template と ±90° template の比較
(Comparison 0° template and ±90° templates)

図 6 実験結果

Figure 6 Experimental results.

4.3 考察

図 6(a)より，全ジェスチャにおいて正面のテンプレート (0° template) のみを利用する場合に比べ，斜め方向のテンプレート (±45° template) を組み合わせることで，45°，-45°，90°，-90° の認識率がそれぞれ 23%，22%，26%，15%増加したことがわかる．また図 6(b)より，横方向のテンプレート (±90° template) を組み合わせることで，90°，-90° の認識率がそれぞれ 10%，20%増加したことがわかる．(a)と(b)を比較すると，斜め方向のテンプレートに比べ，横方向のテンプレートの認識率が低かった．これは横方向ではテンプレートが正しく取得できないことに起因していると考えられる．

さらに，90° の認識率が他の角度に比べて低い結果となった．これは 90° ではジェスチャを行う右手が手前に来るため，ジェスチャ動作により右肩の座標がずれてしまったためであると考えられる．

また，各ジェスチャを見ると，表 1 より Push ジェスチャの認識率が低い結果となっており，特に正面のテンプレートで正しく認識ができていなかったことが見て取れる．

これより Push 動作が正面のテンプレートに適した動作ではないということがわかる。今回は簡単な動作での評価にとどまっているが、動作によっては正面よりも斜め方向の方がテンプレートに向いている場合もあると思われる。さらに、一度のジェスチャで2種のジェスチャが認識するケースも見受けられた。そこで、ジェスチャごとに認識したテンプレートの数を測定し、最も認識したテンプレートの数が多いジェスチャを最終的なジェスチャにすることで、誤認識を防ぐことができると考えている。

5. おわりに

本稿では深度センサから得られた骨格情報を座標変換し、さらにジェスチャ認識に斜め、横方向のテンプレートを利用すること手法を提案した。そして、手を用いた5種のジェスチャにおいて、正面のみのジェスチャテンプレートをを用いる場合に比べて斜め方向で22.5%、横方向で20.5%認識率を向上させることに成功した。

次に今後の課題について述べる。

5.1 複数 Kinect への拡張

1 台の Kinect で横向きのユーザのジェスチャを認識可能であることを述べてきたが、後ろ向きの場合にはジェスチャがユーザの背に隠れてしまうため、骨格情報を取得することは難しい。そこで複数の Kinect を利用することでユーザのセルフオクルージョンに頑健な認識にする方法が考えられる。

我々の以前の研究[5]では図 7 のように Kinect を配置し、HMD にマーカを取り付け、そのマーカを認識した Kinect をメインの Kinect とし、メインの Kinect から骨格を取得していた。しかし、様々な用途に対応するためにマーカを用いないユーザの顔部分の肌色認識でのメインの決定が必要であると考えられる。

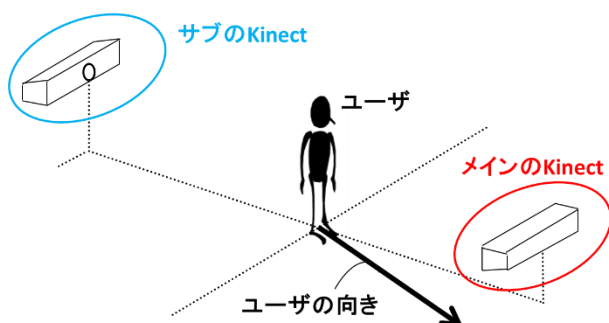


図 7 複数 Kinect を用いたジェスチャ認識

Figure 7 Gesture recognition with multiple Kinects.

5.2 ジェスチャの種類、動作

今回は図 4 のジェスチャを右手でのみ行ったが、今後は両手を用いた複雑なジェスチャを行うことが出来るようシ

ステムを拡張することを考えている。Tarik ら[9]により両手を用いたジェスチャのテンプレートを作成する時に、動きが大きい部位に重み付けを行なう手法が提案されており、この手法を統合することでより複雑な両手での動作の認識率を高めることが可能になると考えている。

参考文献

- 1) Microsoft Kinect, <http://www.xbox.com/ja-JP/kinect>
- 2) Kinect for Windows SDK, <http://www.microsoft.com/en-us/kinectforwindows/develop/>
- 3) 長谷川秀太, 赤池英夫, 角田博保: 姿勢を考慮したハンドジェスチャを利用する機器操作の提案・評価, 情報処理学会研究報告 ヒューマンコンピュータインタラクション (HCI), Vol.2012-HCI-147, No.24, pp.1 - 6, 2012.
- 4) 武田泰幸, 浅野秀胤, 寺林賢司, 梅田和昇: 距離画像センサを用いた操作者の姿勢を考慮に入れたジェスチャ認識システムの構築, 日本機械学会ロボティクス・メカトロニクス講演会 2012 講演論文集, 2A2-L01, 2012.
- 5) 都竹尚紀, 梅澤猛, 大澤範高: 複数の深度センサを利用したユーザの向きに依存しないジェスチャ認識手法, 全国大会講演論文集 2013(1), pp.107-109, 2013.
- 6) Sakoe, H. and Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition, *IEEE Trans. on Acoustics, Speech and Signal Processing*, No.26, pp 43 - 49, 1978.
- 7) 村田 龍吾, 富田 勝義, Le Van Nghia 木村 朝子, 大島 登志一: 3 次元仮想空間とのインタラクションに適したジェスチャインタフェース・ツールの開発, 情報処理学会研究報告 ヒューマンコンピュータインタラクション (HCI), Vol.2012-HCI-147, No.26, pp.1 - 8, 2012.
- 8) waxin' lyrical, <http://www.waxinlyrical.com/KinectGestures>
- 9) Arici, T., Celebi, S., Aydin, A. S. and Temiz, T. T.: Robust gesture recognition using feature pre-processing and weighted dynamic time warping, *Multimedia Tools and Applications*, pp.1 - 8, 2013.