

Kinect を用いた、障害者支援を目的とした指文字、手話の文字変換

松元 勇斗[†] 築地 立家[‡]東京電機大学大学院 理工学研究科 情報学専攻^{†‡}

1. 研究の背景と目的

1.1 背景

Kinect のハックによって開発されたアプリケーションの中でも取り分けて有名なのが、日本人の鷺尾友人氏によって開発された「kinect-ultra」である。

この Kinect-ultra は Kinect を使ってウルトラセブンになりきることを目的としたアプリケーションである。

Kinect-ultra は、ポーズの検出や認証に難しいアルゴリズムや処理を用いることなく、Kinect 独自の動作認証システムであるスカルトンジョイントを使い、手や腕、肩のベクトルを変換するだけでポーズの検出と認証を行っている。

この認証方法を用いることで、Kinect を使ったモーションキャプチャやポーズ認証が実用的となり、個人の開発レベルでもモーションキャプチャが可能となった。

Kinect-ultra が世に登場して以来、世界中の様々な機関や団体が Kinect を用いた動作処理の研究を行い、様々な用途を目的としたアプリケーション開発が進んでいる。

1.2 目的

本研究では、Kinect を社会貢献に向けて利用できないかを研究する。その一環として、Kinect のカメラ、赤外線センサを活かし、指文字及び手話を読み取り、文字やに変換して表示する、目や耳が不自由な人を支援するアプリケーション開発を目標とする。

また、Kinect を用いた手話の文字化はヨーロッパやアメリカといった、いくつかの海外の大学の研究チームが現在研究を行っているが、国によって手話が全く違う点やこの様な社会貢献目的での研究が普及していないことを考え、独自の方法で研究を行い、日本語用の手話を用いることで他国の研究との差別化を図る。

独自の研究方法とは、Kinect のハードの使用上難しいと言われている指の認証を OpenCV や OpenGL といったライブラリを用いて行う。また、スケルトンの位置と指の本数、ベクトルの変換によるポーズ認証により、手話の動作認証、手話の文字化を実証する。

今回の研究では、指の認証と指文字の検出、指文字を表示するソースコードを独自で開発し実験を行った。

これにより、他の研究チームにはない研究結果を導けると考え、日本固有の手話表現を文字化できると考えた。

2. 手話と指文字

手話とは、手指動作と非手指動作を同時に行う視覚言語である。非手指動作とは、顔の部位（視線、眉、口、顎など）を使った動作である。

今回の研究では、この非手指動作のほとんど使われない日本語対応手話と呼ばれる手話の文字化を目指す。

また指文字とは、指や手を使い単語を一文字ずつ表現する手法である。手話の一要素であり、手話の弱点である語彙の少なさを補う為に用いられる。

指文字は固定されたポーズなので今回の様な Kinect を用いた実験に使用するに適している。

3. ポーズ・指文字の文字変換

本研究では、手話の文字変換にポーズの検出と表示処理を用いる。

具体的には、手話の動作の際に生じるポーズを検出するためには、ジョイントの位置やジョイントを結んだ線のベクトル、距離などを使い検出を行う。

例えば手話で「こんにちは」と表現するポーズは、手を目の前にし、肘を 90° 曲げた腕を、そのまま前に倒す動作である（手でお辞儀をするような動作）。

この動作を文字化するのであれば、肩・肘・腕のジョイントの角度の変化、手首のカメラに対する向きの変化を取得し、「3点を作る角度が 180° かつ地面にほぼ直角に上方に向いている」状態から「3点がほぼ直線に並び、かつ地面に対し水平である」状態に変化した際に、「こんにちは」と表示する。

指文字の文字変換も同様に行う。検出方法としては、主に指の数、手のひらと手首の位置、手のひらから指先までの長さ(x軸とy軸)を用いて行う。

4. 指の検出

KinectSDK を用いた指の検出では、直接指のジョイント情報を取得することができないため、手のひらの関節の位置を参考にし、手領域のマスク画像を作成し用いることで指先の検出処理を行う。

実験では、指認証にスケルトンの位置や指の数といった情報を用いて、OpenGL ライブラリを使用し、ソースコードを作成した。その為、本研究では直接 kinect-ultra のライブラリは使用していない。

Character conversion of the finger character aiming at the disabled person support using Kinect, and sign language.

[†]Yuto Matsumoto

[‡]Tatsuie Tsukiji

^{†‡}Tokyo Denki University. Science-and-engineering graduate course. Informatics

5. 手指の検出

KinectSDK を用いた指の検出は、「可視画像座標系」「デプス画像座標系」「スケルトン座標系」の座標系の相互変換を用いて行う。

使うことで手の領域を抽出し、手の輪郭を調べ、手のひらから検出する。その後、ジョイントの位置を参考にし、手領域のマスク画像を作成することで指の位置を検出する。フローチャートを図1に示す。

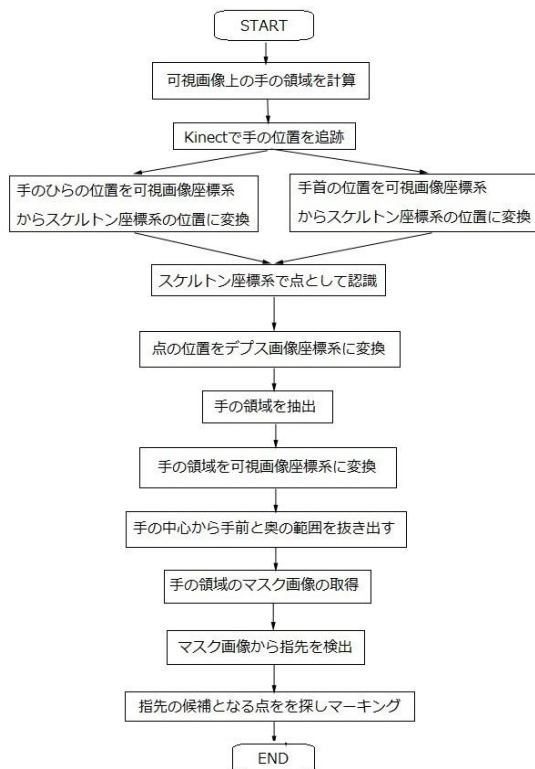


図1 指認証のフローチャート

6. 指文字の検出

日本語に対応した指文字のほとんどは、指文字の検出で用いた3種類の情報で表現することができ、今回の実験では表現が大きく違う3種類の指文字を検出させることで指文字の検出が可能であるか証明する。また、レスポンスの高速化のため可能な限り少ない処理で検出させることを目標とする。

実験で使用した指文字は、日本語の「あ」「し」「な」の3種類であり、特徴点抽出について表を用いて解説する。また、実際の検出の結果を図2に示す。

表1 指文字の認証情報

	指の数	手首の位置	手のひらから指先までの長さ	表示までの平均時間(ミリ秒)
あ	1	下	x軸が5cm以上	267
し	3	水平	x軸が5cm, y軸が8cm以上	250
な	2	上	y軸が重力方向に8cm以上	117



図6 指文字「あ」の検出

本研究の新しい点は、特徴点抽出の際に複雑な画像認証や処理を用いていない点である。

手話や指文字を使った会話は動作が早いので、複雑な処理でポーズを検出しようとするとはスムーズなコミュニケーションが行えない。

その点、本研究は検出から表示までのレスポンスに優れており、高いマシンスペックを要求することもない。

例を挙げると、フランスの研究機関が開発した手話認証システムでは、「bonjour(ボンジュール)」を表す動作を行ってから文字の表示まで約1秒から2秒程かかる。それに対し今回研究で開発した指認証は平均して1秒以内で表示に成功している。

7. 今後の課題

今回の実験で、指認証のような固定されたポーズや、体全身を使ったポーズの検出は可能であるとわかった。

しかし手話での会話に必要な一連の動作、つまり「モーション」の検出及び判定には、より高度で複雑な処理が必要であり、Kinect単体では難しいことがわかった。

今後は実用性化に向けて、顔の表情や口を検出し処理するため、Kinectを同時に2台以上使って動作と表情の検出が可能であるか実験が必要であり、音声認識を用いて、入力された音声解析し、CGで手話を表示させ、インプット、アウトプットを両立させることが望ましい。

8. 参考文献

- [1] 中村薫, 齊藤俊太, 宮城英人: 「Kinect for Windows SDK プログラミング C++編」, 株式会社ゲッシュ
- [2] 永田雅人: 「実践 OpenCV 映像処理&解析」, 株式会社カットシステム
- [3] Zhou Ren, Jingjing Meng, Junsong Yuan: 「Robust Hand Gesture Recognition with Kinect Sensor」
- [4] Paul Doliotis, Alexandra Stefan, Christopher McMurrough, David Eckhard, and Vassilis Athitsos: 「Comparing Gesture Recognition Accuracy Using Color and Depth Information」
- [5] 鷲尾友人: Kinect-ultra
<http://code.google.com/p/kinect-ultra/>