

# Kinect の利用 ・ FFT

中川 夏綺

2016年11月2日

IS Report No. 2016113001

---

**IS** Report

Medical Information  
System Laboratory

## Abstract

非接触で人のゆらぎを計測可能な装置として Kinect がある。Kinect によって骨格座標の 3 次元データを取得し、周波数解析を行うことでゆらぎに含まれる成分を調べることができる。これにより人の状態とゆらぎの特徴について検討することが可能となる。

キーワード: Kinect, 高速フーリエ変換, 短時間フーリエ変換, 3 次スプライン補間

---

---

# 目次

---

第1章 はじめに . . . . .	2
第2章 Kinect . . . . .	3
2.1 Kinect の概要 . . . . .	3
2.2 補間方法 . . . . .	5
第3章 Kinect データの活用 . . . . .	6
3.1 高速フーリエ変換 . . . . .	6
3.2 短時間フーリエ変換 . . . . .	6
3.2.1 窓関数 . . . . .	7

---

---

## 第 1 章 はじめに

---

体の重心のゆらぎから平衡感覚の異常や運動失調の症状の特徴を調べる研究などが進められている。このような背景の下，体のゆらぎを計測する装置の一つに Kinect がある。

Kinect を用いることで体に直接マーカー等を装着することなく，体を認識し骨格の追跡をリアルタイムで行うことができる。これにより様々な状況下で体の各部位のゆらぎの特徴を見ることができ，特徴を見るための解析手法の一つとして，重心動揺の研究でも多く用いられている高速フーリエ変換が挙げられる。これにより，ゆらぎにどのような周波数成分が含まれているかを調べることができる。また，周波数成分の時間変化を調べるためには短時間フーリエ変換を行う。

本稿では Kinect の概要とデータの処理方法，解析方法 (高速フーリエ変換，短時間フーリエ変換) について述べる。

---

## 第 2 章 Kinect

---

### 2.1 Kinect の概要

Kinect とは Microsoft 社から発売されたセンサーデバイスである。本体には RGB カメラ，近赤外光を用いた距離センサー，マイクロフォンアレイが備わっている。Kinect の機能として，以下が挙げられる。

- 体の各部位 (25 点) の位置を取得
- 人物の抽出
- フェイストラッキング
- ジェスチャーコントロール

Kinect の外観を Fig. 2.1 に示す。また，Kinect は 25 箇所の骨格位置を検出できる。骨格位置は Fig. 2.2 に示す。

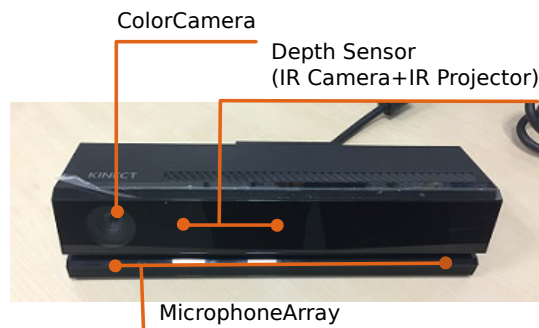


Fig. 2.1 Kinect の外観 (参考文献 [?] より自作)

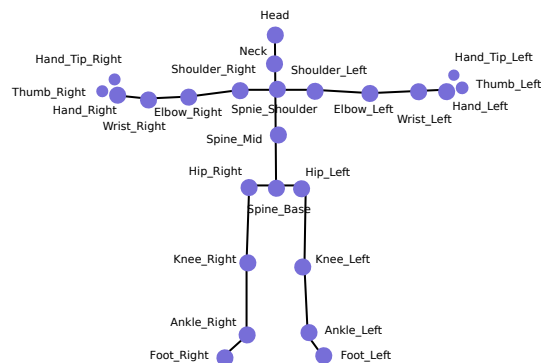


Fig. 2.2 骨格検出位置 (自作)

位置座標の値は Kinect から見たときの X, Y, Z 座標それぞれの値を表している. この座標のイメージを Fig. 2.3 に示す. X 軸は左右方向の揺れ, Y 軸は上下方向の揺れ, Z 軸は前後方向の揺れを表している. 実験で用いる Kinect v2 の動作要件を Table. 2.1 に表す. Kinect を NIRS 装置と同期させる場合は Presentation で同期が可能である.

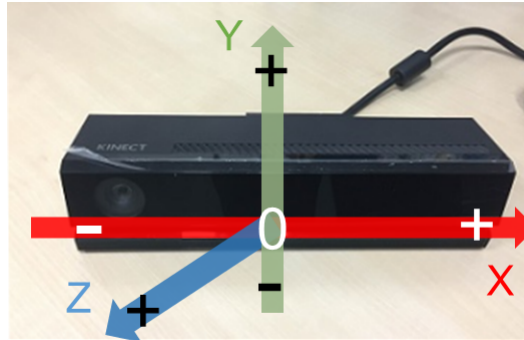


Fig. 2.3 3次元座標取得イメージ (自作)

Table. 2.1 Kinect v2 の動作要件

OS	Windows 8
コンパイラ	Visual Studio 2012 以降
接続端子	USB3.0
CPU	Dual-Core 2.66[GHz]
GPU	DirectX 11.0 対応の GPU
RAM	2.0[GBytes]

## 2.2 補間方法

Kinect でデータを取得し csv ファイル等に保存するプログラムは、Visual Studio や Presentation で作成することが可能である。しかしデータの書き込みなどにより等間隔に取れない場合がある。そのため、周波数解析を行うために 3 次スプライン補間を行うなどの対策が必要である。Fig. 2.4 に python での 3 次スプライン補間のコード例を示した。

```
D1=data.ix[0:1000,0]
D2=data.ix[0:1000,5]
convert=intp.interpld(D1,D2,kind='cubic')
x=np.arange(D1[0],D1[1000],33)
y=convert(x)
```

Fig. 2.4 3 次スプライン補間プログラム例 (自作)

---

## 第 3 章 Kinect データの活用

---

### 3.1 高速フーリエ変換

高速フーリエ変換 (Fast Fourier Transform: FFT) とは、離散フーリエ変換 (Discrete Fourier Transform: DFT) を高速に処理するためのアルゴリズムである [?]. このフーリエ変換とは、ある連続信号を様々な周波数の正弦波と余弦波の重ね合わせとして表現することである。これにより信号の周波数成分を知ることができる。しかしコンピューターでは無限個の連続信号を扱うことはできないため、1 秒間に適当な個数のデータを抽出する。この 1 秒間に取り込むデータ数をサンプリング周波数という。そして離散した点の集合を有限の時間で打ち切り、計算を行う。これが DFT である。

DFT の計算は大量の繰り返し乗算を必要とし、データ数を増やすと指向的 ( $N^2$ ) に乗算回数が増える。しかしこの DFT の計算には対称性がある。この対称性を利用する手法である FFT ではその乗算回数を  $N \cdot \log_2 N / 2$  回に減らすことができる。実際に FFT を行い出力されるスペクトルの例を Fig. 3.1 に示す。

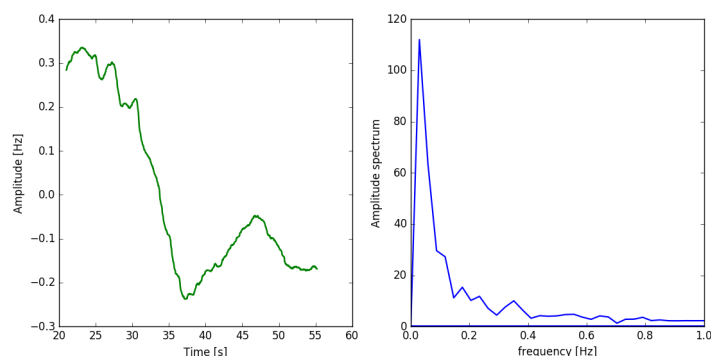


Fig. 3.1 FFT 実行例 (左:時系列ローデータ, 右:スペクトル図)

### 3.2 短時間フーリエ変換

短時間フーリエ変換 (Short time Fourier Transform) とは信号を一定の大きさの区間で分割し、区間ごとにフーリエ変換を行い、信号スペクトルの時間変化を解析する手法である [?]. 分割する区間を窓といい、窓関数を用いて切り出す。窓ごとに FFT が行われ、横軸時間、縦軸周波数で表される Fig. 3.2 のようなスペクトログラムを見ることができる。窓幅を短くすると時間分解能が高くなり、周波数分解能が低くなる。一方窓幅を大きくすると周波数分解能が高くなり、時間分解能が低くなる。時間分解能と周波数分解能を同時に上げることはできない。



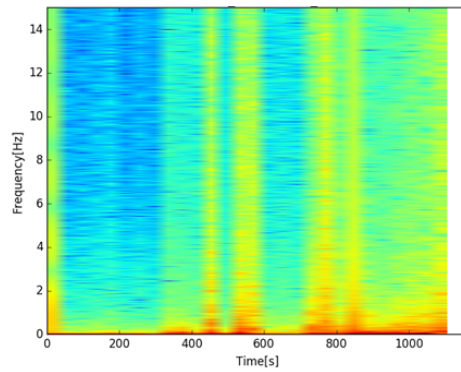


Fig. 3.2 スペクトログラム

### 3.2.1 窓関数

高速フーリエ変換のように、ある区間で波形を切り出すと端が不連続となる。これを防ぐために窓関数を掛けることで、切り出す端のずれの影響を少なくする。窓関数の種類を Fig. 3.3 に示した。

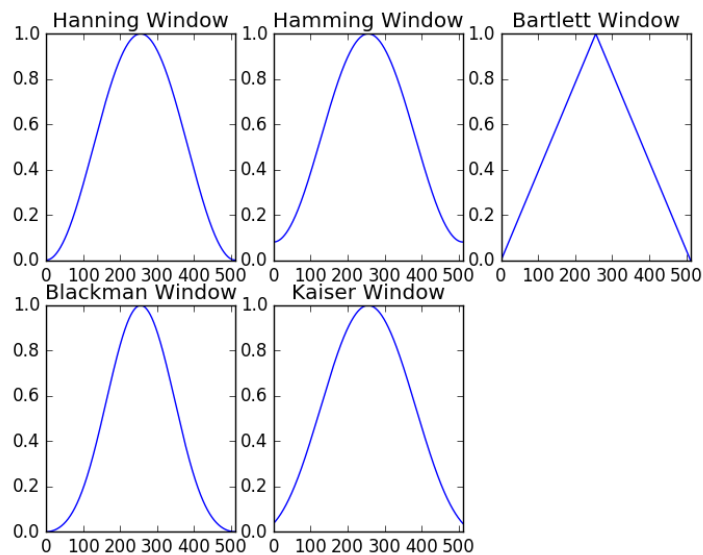


Fig. 3.3 窓関数例 (参考文献 [?] より自作)

一般的に用いられることが多いのはハニング窓とハミング窓である。ハニング窓は区間の端の向かってなだらかに小さくなり、両端は0である。サイドローブが比較的小さいため、小さい電力のスペクトルの検出に向いている [?]. しかしハニング窓は両端が0であることにより信号成分が全く反映されないという欠点がある。この欠点を考慮したものがハミング窓である。ハニング窓に比べて不連続性が残る分、周波数分解能は高まる。