

RPiとEPICSを使った 静磁場可視化システム

2017.12.15

Noboru Yamamoto

J-PARC Center/KEK

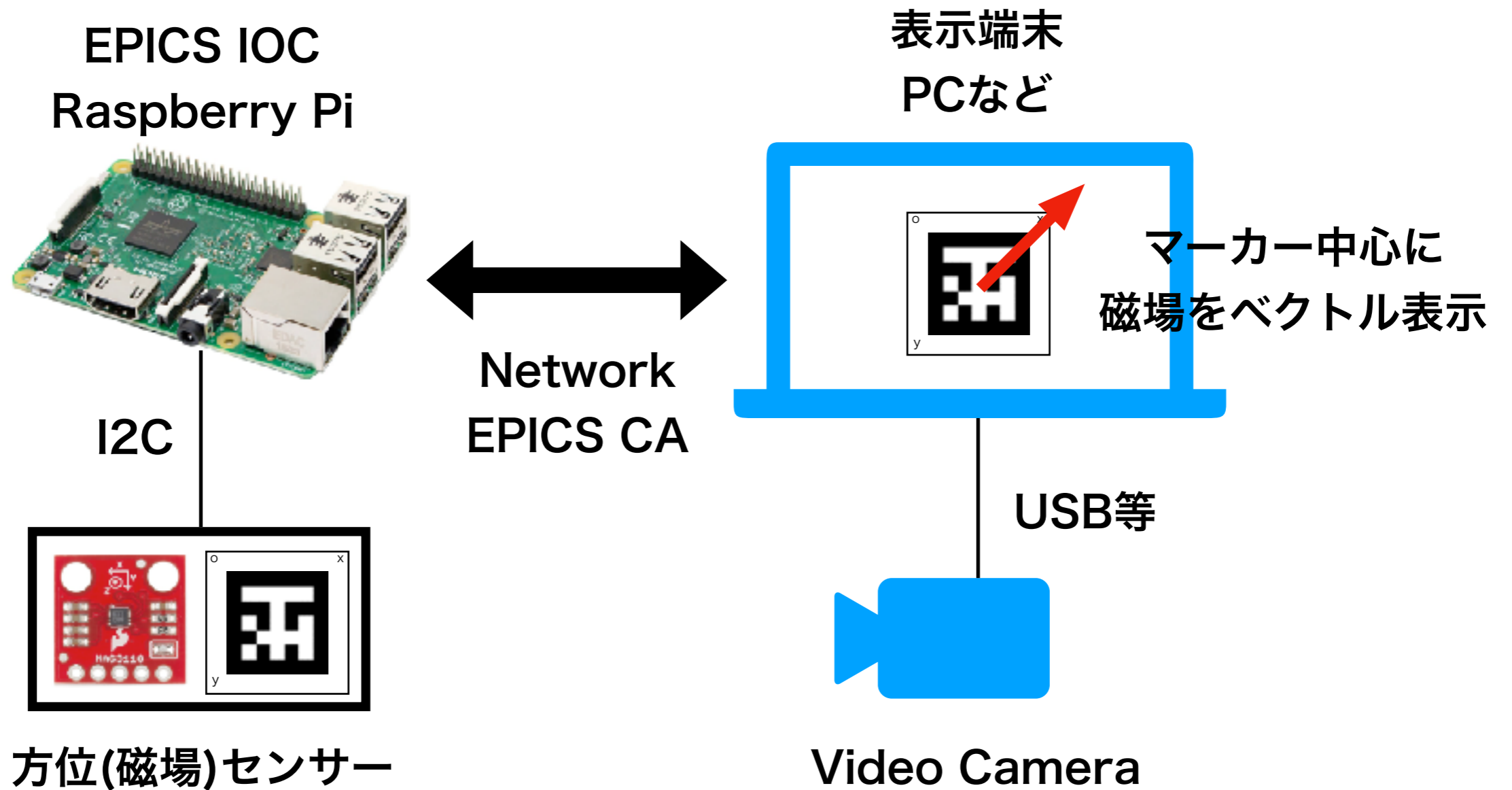
Accelerator Control Group

概要

- 小型計算機Raspberry Piと汎用ホール素子を使った、安価でポータブルな静磁場可視化システムを作成した。
- USBカメラの画像に磁場センサの場所での磁場強度をベクトルとして表示する。(数値としても画面表示している)
- ベクトルの記録機能により、付近の三次元的な磁場分布を表示可能
- 画像の記録機能もある。



システムの概要

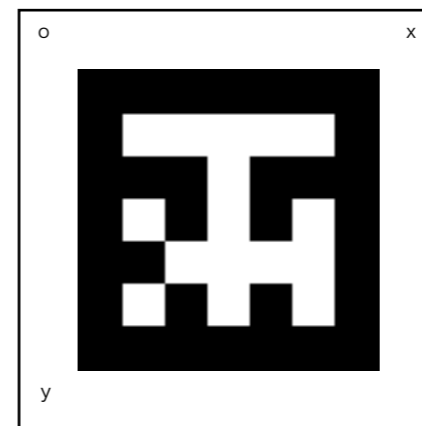


構成

- データ収集にはI2C接続の3軸ホール素子とEPICS+devAsynI2Cを使った
- 磁場の可視化の為に、USB接続カメラ、ARToolKitに依るマーカーおよびProcessingを利用する。
- ProcessingにEPICS JAVA CAライブラリを導入することで、測定部と表示部を分離することが可能になった。



Raspberry Pi



ARマーカー
(NyID)



検出素子
MAG3110

ソフトウェア構成

- 検出器側
 - OS: Raspbian : Debian based Linux for Raspberry Pi
 - EPICS: Stream Device + devAsynI2C+AsynDriver
- 表示側：
 - Processing : 電子アートとビジュアルデザインのためのプログラミング言語であり、統合開発環境。Javaをベースにしている。
 - NyARToolkit: Processing 向けARマーカライブラリ

Raspberry Pi

- 英国ラズベリーパイ財団によって教育用を意図に設計された小型ボードコンピュータ。複数のメーカーによって実装され、販売されている。低価格：～数千円
- Pi-1A/Pi-1A+/Pi-B/Pi-1B+/Pi-2B/Pi-3B/CM/CM3/Pi-Zero/Pi-Zero W といったバリエーションがある。
- GP-IO端子をもち、I2C/SPIなどの素子を接続可能。
- Linux baseの 開発環境が提供されている。公式のOSは Raspbian/NOOBS

磁気センサ

- 方位検出用の3軸ホール素子が販売されている。
 - HMC 5 8 8 3 L (ハネウェル社)
 - MAG3110(Free Scale)
 - HMC6343(Honeywell)
- 方位測定用のため、動作レンジが狭い(~数ガウス)なのが弱点
 - 地球磁場は 数十 μ T(数100 mガウス)程度
- いずれもI2C/SPIをサポートしている。
 - デジタルデータとして磁場の強さを読み出せる。

I2C/SPI

- デジタル素子をマイクロコントローラに接続するための業界標準の規格
 - I2C(I²C) : Inter-Integrated Circuit
 - SPI : Serial Peripheral Interface
- 数本(2~4)の信号線で複数のデバイスと通信できる (バス形式)
- 素子毎のレジスタに書き込み/読み出しを行うことで制御
- 素子の指定は、I2Cではデバイスのアドレス(7bits)、SPIではCS信号に依る。

EPICS

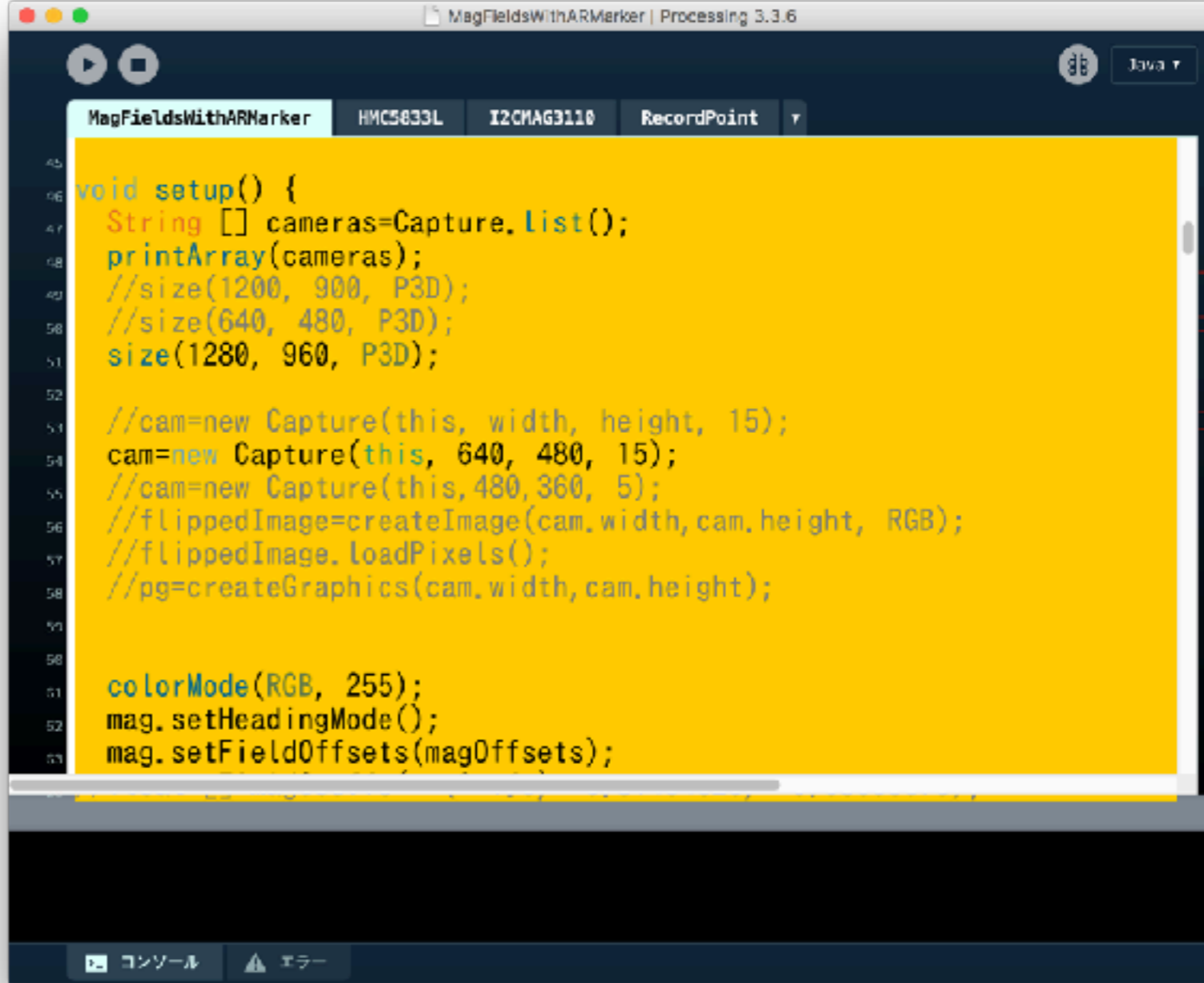
- EPICS: Experimental Physics and Industrial Control System
 - Software framework for construction of network based distributed computer control system.
 - Distributed as free software.
 - Used in the various accelerator facilities and experimental physics facilities around world

Asyn/Stream/devAsynI2C

- Asyn/Stream
 - generic frame work to support class of devices using serial-like communication.
- devAsynI2C
 - library to support I2C devices on Raspberry Pi platform running EPICS/asyn/stream.

Processing

- 電子アートとビジュアルデザインのためのプログラミング言語であり、統合開発環境でもある。
- Javaをベースにしている
 - Javaなどにより拡張可能。このプロジェクトではEPICS-CAをJCAライブラリを使って拡張している。
- 音響／画像などの処理が簡単にかける。



```
MagFieldsWithARMarker | Processing 3.3.6
MagFieldsWithARMarker HMC5833L I2CMAG3110 RecordPoint
void setup() {
  String [] cameras=Capture.list();
  printArray(cameras);
  //size(1200, 900, P3D);
  //size(640, 480, P3D);
  size(1280, 960, P3D);

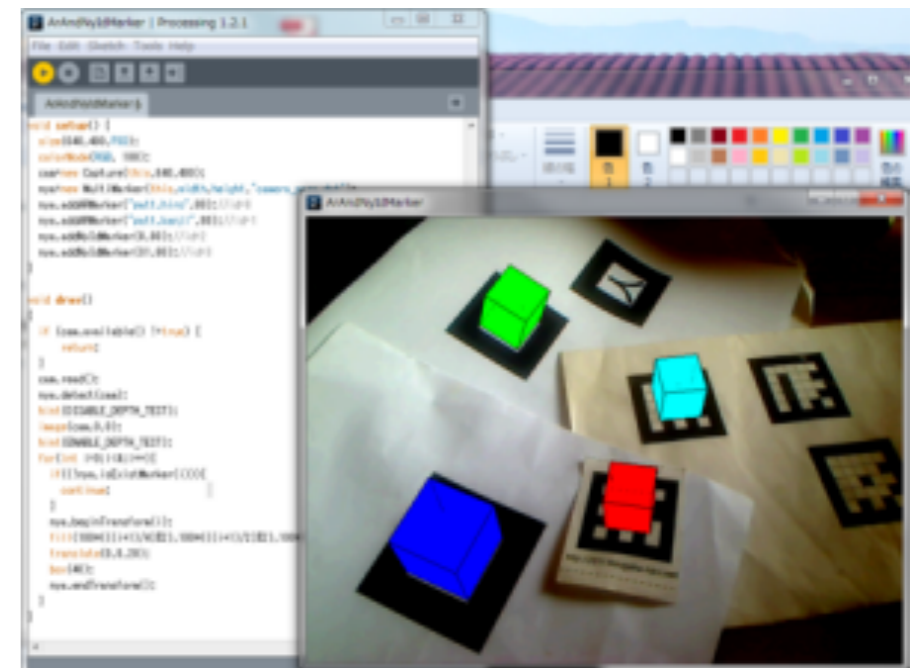
  //cam=new Capture(this, width, height, 15);
  cam=new Capture(this, 640, 480, 15);
  //cam=new Capture(this, 480, 360, 5);
  //flippedImage=createImage(cam.width, cam.height, RGB);
  //flippedImage.loadPixels();
  //pg=createGraphics(cam.width, cam.height);

  colorMode(RGB, 255);
  mag.setHeadingMode();
  mag.setFieldOffsets(magOffsets);
}
```

Processing 開発環境

NyARToolkit

- ARToolkit: an open-source computer tracking library for AR(Augmented Reality).
 - originally developed by Hirokazu Kato of Nara Institute of Science and Technology in 1999
- NyARToolkit: ARToolkit互換なビジョンベースARライブラリ
 - Java/Processing/C#/Unityを対象とする。
 - <https://nyatla.jp/nyartoolkit/wp/>
- このプロジェクトでは、USBカメラで取得した画像内のAR Markerの位置および向きをこのToolkitを使って取得している。それに合わせて、磁場のベクトルを表示している。



NyARToolkit ホームページから：ARマーカのトラッキング



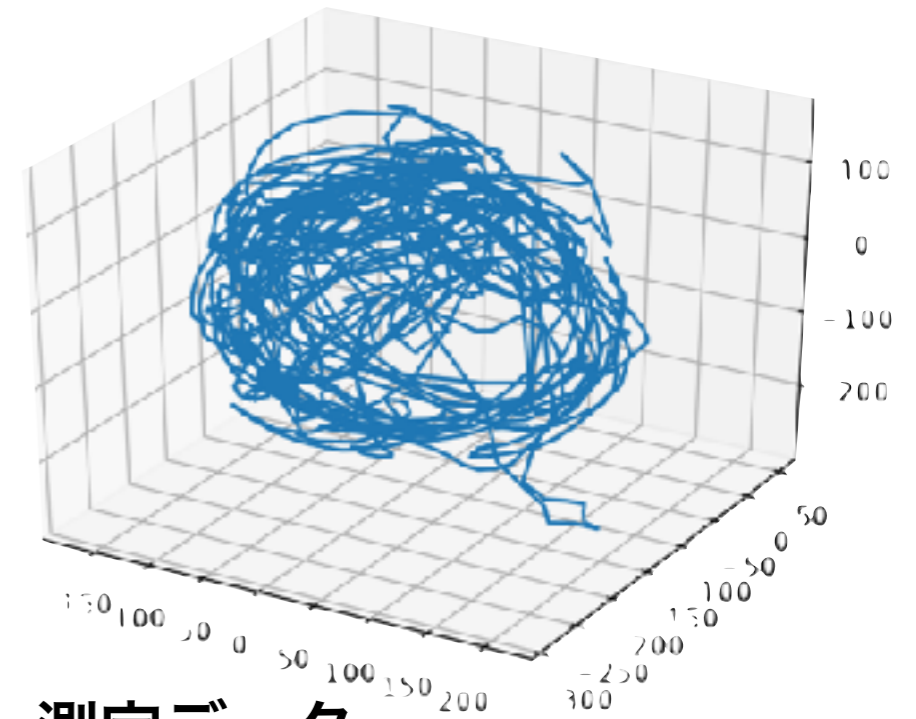
NFT Trackingの例

EPICS/caj on Processing

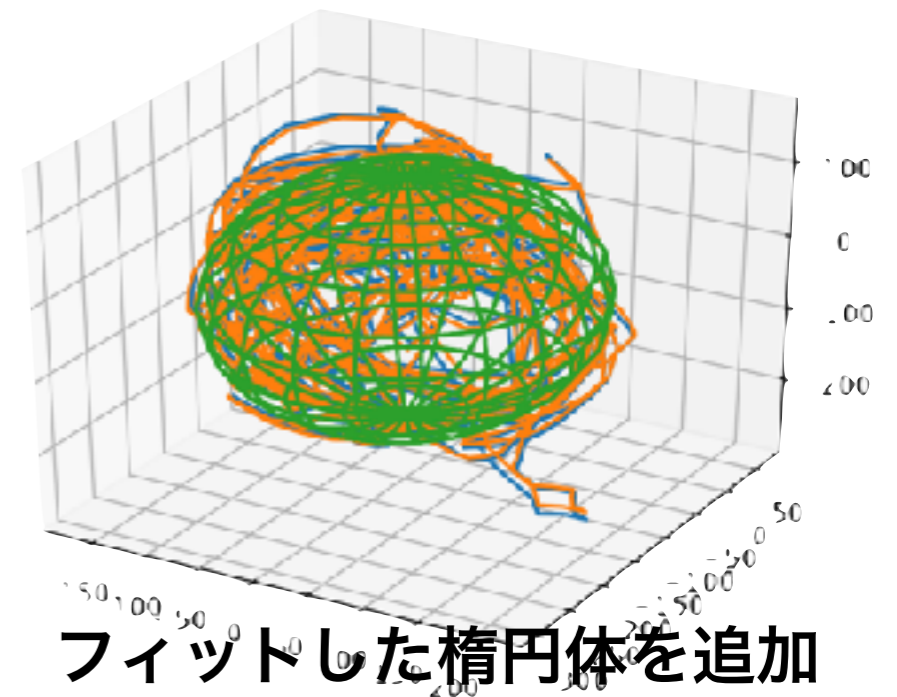
- 磁気センサからデータを収集するCPUとUSBカメラの画像を処理するCPUはネットワークで繋がっている。センサデータはEPICS のCA(Channel Access)プロトコルで交換される。
- EPICS のJavaによるCAの実装である cajをprocessingのライブラリに登録する。
- caj の設定ファイルJCALibrary.propertiesで addressListなどを環境に合わせて調整すること。

方位計の簡易校正

- 方位計素子はその場所での磁場に比例した出力を出力するがそのデータにはオフセットや軸毎の感度の違いが含まれている。
- 地磁気は空間上の狭い領域ではほぼ一定とみなせるので、この領域での様々な向きで測定した方位計のデータを3次元空間上にプロットすると、地磁気の高さを半径とする球面上に分布するはずである。このことを使って、実測データから、軸ごとの感度の違い及びオフセットを求めることができる。



測定データ



フィットした楕円体を追加

Appendix

RPi/I2C 設定

- i2c-toolsのインストール
 - `sudo apt-get install i2c-tools wiringpi`
- raspi-config
- i2c-tools
 - `i2cdetect` : i2c-busに接続されているモジュールの検出
 - `i2cdump`: i2c モジュールのレジスタからデータを取り出して表示する
 - `i2cset/i2cget`: レジスタへの書き込み/読み出し

Python/I2C on RPi

- python moduleのインストール
 - `sudo apt-get install python-smbus wiringpi`
- python プログラム例
 - `import smbus`
`bus=smbus.SMBus(1)`
`data=bus.read_byte_data(mod_addr, reg_addr)`