# METRICS FOR ANALYTICS AND VISUALIZATION OF BIG DATA WITH APPLICATIONS TO ACTIVITY RECOGNITION

BY

ROHAN R. ARORA

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2016

Urbana, Illinois

Adviser:

Associate Professor Prashant G. Mehta

# ABSTRACT

Activity recognition systems detect the hidden actions of an agent from sensor measurements made on the agents' actions and the environmental conditions. For such systems, metrics are important for both performance evaluation and visualization purposes. In this thesis, such metrics are developed and illustrated.

For human activity recognition datasets, a reporting structure is described to visualize the metrics in a systematic manner. The other contribution of this thesis is to describe a visualization tool for estimating the orientation (attitude) of a rigid body from streaming motion sensor (accelerometer and gyroscope) data. A feedback particle filter (FPF) is implemented algorithmically to solve the estimation problem.

*To my parents, friends and professors who have helped me reach this point.*

# ACKNOWLEDGMENTS

Above all I would like to thank my adviser, Professor Prashant G. Mehta, for his infinite support, encouragement and guidance through my master's program at the University of Illinois at Urbana-Champaign.

I would like to thank my lab-mates Adam Tilton, Dr. Tao Yang, Shane Ghiotto, Chi Zhang, Amirhossein Taghvaei, Sahand Hariri, Shiya Liu and Mayank Baranwal for the bond we share as co-workers and friends. A special shout-out to Adam for being an awesome mentor and making me a part of his research and entrepreneurial journey.

I am highly indebted to my parents for their unconditional love and support.

I wish to thank Angie M. Ellis for her help in dealing with my administrative duties, Laurie Fisher and Jeniffer M. Carlson for their patience in answering my questions related to the graduate program.

I would like to thank Jamie Hutchinson for the extensive and swift proofreading.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# CHAPTER 1

# INTRODUCTION

The goal of activity recognition (AR) is to identify the physical activity from a set of sensor measurements. In an offline system, the activities are identified from stored time-indexed values of sensor data. In an online system, the activities are identified from streaming sensor measurements.

An automatic AR system has many benefits. In the 21st century our work and leisure activities tend to be less physically demanding. Activity recognition systems provide us with means to study the impact of this sedentary lifestyle and to an extent establish links between physical activity and say hypertension, diabetes, cancer and depression. Healthy People 2010 [1], a program of nationwide health-promotion and disease-prevention goals set by the United States Department of Health and Human Services has tagged physical activity as a leading health indicator. Furthermore, tracking physical activities and providing useful metrics to the user can help motivate him/her to lead an active lifestyle.

With advancements in the fields of medicine and enhancement in the quality of life, life-expectancy has increased dramatically leading to an increase in the elderly population. One of the major hazards for the elderly population is falling. The fall can be attributed to muscle weakness, variations in blood pressure, balancing issues, etc. An AR system can help detect a fall and alert emergency workers. This can help reduce the emergency response time and help lower the rate of fall-related deaths among the elderly. In a best case scenario, it could be used to track situations or events that may lead to a fall and alert the elderly user, preventing the fall as well as expense of emergency services.

Sources of measurements for current AR systems are based are heart-rate monitors, video cameras and micro-electro-mechanical-systems (MEMS) such as accelerometers and gyroscopes. Heart-rate monitors involve strapping the sensor around the chest and are considered obtrusive and uncomfortable.

Processing video content is known to be computationally exhaustive and it is difficult to obtain accurate real-time results.

MEMS that are widely used in AR applications include the accelerometers and gyroscopes measuring 3D accelerations and angular velocities respectively. The first publication reporting the use of accelerometers for activity recognition dates back to 1983 [2]. Advancements in MEMS technology over the past two decades have led to their miniaturization, lowered the power consumption and reduced manufacturing costs. As a result accelerometers, gyroscopes and magnetometers are widely embedded in smart phones, smart watches, etc., and can be leveraged to design AR systems.

The contributions of this thesis are two-fold:

1. We develop metrics and associated methodology to evaluate different AR machine learning algorithms. The results are presented in Chapter 3.

2. We develop a visualization framework to re-create the in-place 3D rotational motion using gyroscope and magnetometer measurements with the objective of understanding the physics underlying the motion of the motion sensors. This framework is presented in Chapter 4.

# CHAPTER 2

# DATASET

This chapter provides a brief overview of the University of Illinois at Urbana-Champaign (UIUC) dataset and the University of Southern California Human Activity Recognition dataset (USC-HAD). These datasets are primarily composed of measurements logged at a frequency of 50 Hz to 100 Hz from MEMS sensors such as accelerometers, gyroscopes and magenetometers. The framework discussed in Chapter 3 is applied and illustrated for these datasets.

## 2.1 MEMS Sensors

### 2.1.1 Accelerometers

Accelerometer sensors measure acceleration. The unit of measurement is meters per second squared ($m/s^2$) or G-forces (g). For activity recognition a 3-axis accelerometer is used. Such an accelerometer measures acceleration along the $x$, $y$ and $z$ directions in the device's coordinate system.

### 2.1.2 Gyroscopes

Gyroscope sensors measure angular velocity. The unit of measurements is degrees per second (°/s) or radian per second (rad/s).

### 2.1.3 Magnetometers

Magnetometer sensors measure the strength and direction of the magnetic fields. The unit of measurement is microtesla (µT).

## 2.2 Datasets

### 2.2.1 UIUC Dataset

This dataset was collected at the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign (UIUC). An Android application was developed to log motion sensor (accelerometer, gyroscope and magnetometer) data from Android Wear smart watches and Android smart phones. The application was set to sample data at 50 Hz. The Android Wear smart watch was mounted on either the left or right wrist of the subject. The subject performed an average of 10 repetitions of the 7 different strength training exercises/activities listed in Table 2.1 for 10 days.

Table 2.1: Overview of the UIUC Dataset

| **Dataset** | **No. of Subjects** | **Activities** | **Sensors** |
|---|---|---|---|
| UIUC Dataset | 15<br>10 male<br>5 female | Dumbbell Hammer Curl<br>Dumbbell Front Raise<br>Dumbbell Flyes<br>Dumbbell Lunges<br>Dumbbell Rear Lunge<br>Dumbbell Shoulder Press<br>Dumbbell Side Raise<br>Barbell Curl<br>Barbell Squat<br>Kettlebell One-Arm Row<br>Machine - Walking | 3-axis accelerometer ($\pm 2$ g)<br><br>3-axis gyroscope ($\pm$ 500 °/s) |

A session is defined as set of measurements for the subject performing 10 (or more) repetitions of a particular activity/exercise. In each session for each incoming measurement the following data was logged:

1. Timestamp
2. Acceleration along the X-axis
3. Acceleration along the Y-axis

4. Acceleration along the Z-axis
5. Angular velocity around the X-axis
6. Angular velocity around the Y-axis
7. Angular velocity around the Z-axis

The coordinate system relative to the device is show in Figure 2.1. Each session is labeled with the corresponding activity/exercise name, wrist location and the number of repetitions performed by the user during a session.

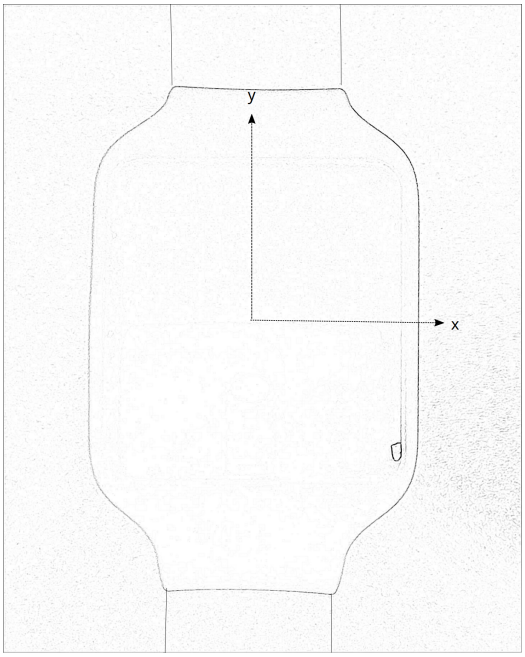


Figure 2.1: Coordinate System Relative to the Device (Sony Smartwatch 3)

### 2.2.2 University of Southern California Human Activity Dataset (USC-HAD)

USC-HAD is an open-source dataset [3]. Activity types and the sensors used are listed in Table 2.2. Each subject performs each activity for 5 trials (a trial is defined as an individual performing an activity repeatedly for approximately 20 seconds), and the data for each single trial is stored separately. The inertial measurement unit (IMU) comprising a 3D accelerometer and gyroscope is affixed to the front right hip of the subject, with the $x$-axis

pointing in the gravity direction and perpendicular to the $y - z$ plane. The sampling rate of the IMU is 100 Hz. Figure 2.2 illustrates the data of the 3D accelerations and angular velocities for walking forward and jumping. It can be seen from the figure that jumping incurs more intensive accelerations but relatively milder angular rotations compared with walking forward.

Table 2.2: Overview of USC-HAD

| **Dataset** | **No. of Subjects** | **Activities** | **Sensors** |
|---|---|---|---|
| USC-HAD | 14<br>7 male<br>7 female | Walk forward<br>Walk left<br>Walk right<br>Walk upstairs<br>Walk downstairs<br>Run forward<br>Jump<br>Sit on a chair<br>Stand<br>Sleep<br>Elevator up<br>Elevator down | 3-axis<br>accelerometer<br>(±6 g)<br><br>3-axis<br>gyroscope<br>(±500 °/s) |

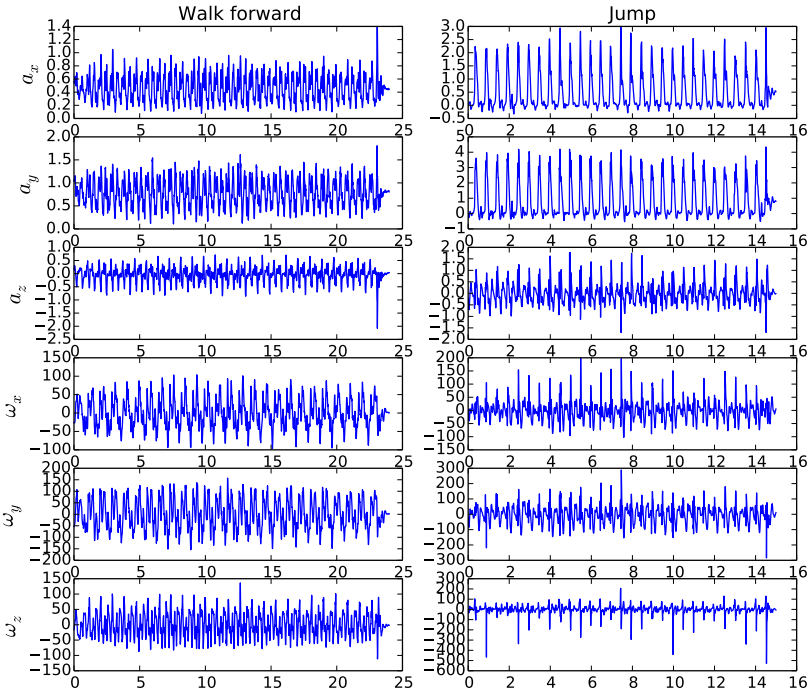


Figure 2.2: Comparison of Sensor Measurements for Walking Forward and Jumping Activities

# CHAPTER 3

# FRAMEWORK

The aim of this chapter is to propose a framework for reporting and visualization for the purposes of performance evaluation and comparison of machine learning algorithms for activity recognition.

## 3.1 Definitions

The following terminology will be used throughout this chapter:

- **Configuration**: A user wearing a smart watch or smart phone at a particular body location
- **Set**: A collection of repetitive motion(s) of the same activity
- **Workout**: A collection of sets of one or more activities
- **Ground Truth**: The number of repetitive motions for a set as reported by the user
- **Estimated Count**: The number of repetitive motions estimated by the algorithm for each set
- **Noisiness**: The assimilation of measurements from a workout for which no repetitive activity is reported by the user
- **Representative Model**: Features extracted from a single set of an activity for a configuration

## 3.2 Set-wise Performance Evaluation

The objective is to evaluate the performance of a feature-set on the sensor data from which it was developed. The following metrics are developed for

each set.

### 3.2.1 Ground Truth minus Estimated Count (GT-EC)

Subtract the number of repetitions detected by the algorithm from the number of repetitions reported by the user. In an ideal case this metric should be zero; i.e., the algorithm tracks the number of repetitions with 100 percent accuracy.

### 3.2.2 False Positive - Estimated Count (FP-EC)

The number of repetitions detected by the feature-set on the data points tagged as noisiness. In an ideal case this metric should be zero.

### 3.2.3 Normed Maximum Acceleration (NMA)

For each of the measurements across a set, compute using the following:

$$\text{Normed Acceleration} := \sqrt{{a_x}^2 + {a_y}^2 + {a_z}^2} \tag{3.1}$$

where:

$a_x$ is the acceleration along the X-axis extracted from the measurement

$a_y$ is the acceleration along the Y-axis extracted from the measurement

$a_z$ is the acceleration along the Z-axis extracted the measurement

Consider the maximum value for Equation (3.1) across measurements for the set.

### 3.2.4 Normed Maximum Gyroscopic Rotation (NMG)

For each of the measurements across a set, compute the following:

$$\text{Normed Gyroscopic Rotation} := \sqrt{{g_x}^2 + {g_y}^2 + {g_z}^2} \tag{3.2}$$

where:

$g_x$ is the angular velocity around X-axis extracted from the measurement

$g_y$ is the angular velocity around Y-axis extracted from measurement

$g_z$ is the angular velocity around Z-axis extracted from the measurement
Consider the maximum value for Equation (3.2) across measurements for the set.

### 3.2.5 Logging Time

The logging time is computed as end time minus start time of the set. This information can be used to determine if the set is valid or not. If the logging time is zero or exceptionally large (exceptions are walking, running, etc.) the set can be classified as an invalid set. An invalid set is not considered during the performance evaluation.

### 3.2.6 Overview

Table 3.1 is constructed based on the metrics developed above for each of the activities.

Table 3.1: Performance across Sets for an Activity

| 1 - **Activity Name** | | | | | | |
|---|---|---|---|---|---|---|
| No. | Identifier | GT-EC | FP-EC | NMA | NMG | Logging Time |

In the table:

No. is the set number

Identifier is the unique identifier used to capture additional information (user, device, device location, etc.) of the set

Thus for each set of an activity we have an overview of the performance of the algorithm. Table 3.2 presents a snapshot for UIUC's dataset for the activity Barbell Curl.

Table 3.2: Snapshot of Performance across Sets for Barbell Curl

| 1 - **Barbell Curl** | | | | | | |
|---|---|---|---|---|---|---|
| No. | Identifier | GT-EC | FP-EC | NMA | NMG | Logging Time |
| .. | .. | .. - .. | .. | .. | .. | .. |
| 9 | 86f1 58b3 | 10.0 - 10.0 | 0.0 | 5.922 | 2.165 | 19.26 |
| 10 | fe6b 1ba2 | 10.0 - 9.0 | 0.0 | 4.75 | 2.165 | 21.20 |
| 11 | 007d fd0d | 10.0 - 10.0 | 0.0 | 5.251 | 2.139 | 19.22 |
| 12 | 8a44 6d9d | 10.0 - 10.0 | 0.0 | 5.105 | 2.373 | 19.32 |
| .. | .. | .. - .. | .. | .. | .. | .. |

## 3.3 Activity-wise Performance Evaluation

In order to garner an overview of the algorithm's performance for a single activity, three metrics are developed which are described in the following subsections.

### 3.3.1 Average Accuracy (Avg. Acc.)

Based on number of repetitions detected by the algorithm, the accuracy metric is defined as:

$$(\text{Ground Truth} - \text{ Estimated Count}) \tag{3.3}$$

This accuracy metric is compartmentalized into:

- Accuracy $\pm 0$: Herein the absolute value of (3.3) is 0
- Accuracy $\pm 1$: Herein the absolute value of (3.3) is 1
- Accuracy $\pm 2$: Herein the absolute value of (3.3) is 2

An average accuracy metric is computed for each of the above listed compartments by summing up the accuracy values corresponding to a compartment for each valid set for an activity and then dividing it by the total number of valid sets for that activity.

### 3.3.2 Average Normed Acceleration (ANMA)

ANMA is derived by adding the NMA values for all of the valid sets for an activity and then dividing by the total number of valid sets for that activity.

### 3.3.3 Average Normed Gyroscopic Rotation (ANMG)

ANMG is derived by adding the NMG values for all of the valid sets for an activity and then dividing by the total number of valid sets for that activity.

### 3.3.4 Overview

Table 3.3 is constructed based on the metrics developed above for each of the activities.

Table 3.3: Overview of Set-wise Performance across Activities

| Activity | # S | Acc. 0 | Acc. 1 | Acc. 2 | ANMA | ANMG |
|---|---|---|---|---|---|---|

In the table:

# S corresponds to the total number of valid sets for an activity

Acc. 0 is the average accuracy metric for the first compartment

Acc. 1 is the average accuracy metric for the second compartment

Acc. 2 is the average accuracy metric for the third compartment

Table 3.4 presents a snapshot of the performance on UIUC's dataset across activities.

## 3.4 Insights

### 3.4.1 GT-EC Histogram

A histogram of ground truth values along an estimated count for each activity helps us to draw a conclusion about the performance of the algorithm.

Table 3.4: Overview of Set-wise Performance across Activities for the UIUC dataset

| Activity | # S | Acc. 0 | Acc. 1 | Acc. 2 | ANMA | ANMG |
|---|---|---|---|---|---|---|
| Dumbbell Front Raise | 66 | 0.3 | 0.89 | 0.97 | 5.03 | 1.80 |
| Dumbbell Hammer Curl | 60 | 0.2 | 0.9 | 0.93 | 6.64 | 2.39 |
| Kettlebell One-Arm Row | 54 | 0.24 | 0.85 | 0.98 | 4.66 | 0.83 |
| Dumbbell Flyes | 64 | 0.19 | 0.81 | 0.94 | 4.19 | 0.77 |
| Dumbbell Side Raise | 14 | 0.07 | 0.57 | 0.71 | 5.24 | 2.12 |
| Dumbbell Lunges | 44 | 0.20 | 0.34 | 0.52 | 1.56 | 0.31 |
| Barbell Curl | 75 | 0.32 | 0.92 | 0.97 | 5.29 | 2.21 |
| Dumbbell Rear Lunge | 33 | 0.16 | 0.35 | 0.39 | 1.714 | 0.31 |
| Dumbbell Shoulder Press | 57 | 0.14 | 0.41 | 0.63 | 1.66 | 0.35 |
| Barbell Squat | 48 | 0.42 | 0.79 | 0.89 | 1.36 | 0.17 |

### 3.4.2 NMA vs. NMG Scatter Plot

NMA vs. NMG plot provides a quantitative description with regard to the quality of the feature-set developed.

## 3.5 Workout-wise Performance Evaluation

The objective here is to evaluate the performance of a feature-set(s) selected for an activity on workouts. This workout-wise evaluation facilitates the following:

- Quick understanding of the flow of the workout
- Re-validation of the ground truth

A typical tabular visualization for a workout session is described in the following three subsections.

### 3.5.1 Ground Truth minus Estimated Count (GT-EC)

Subtract the number of repetitions detected by the algorithm from the number of repetitions reported by the user.

### 3.5.2 Confusions

This accounts for the cases where the algorithm's prediction contradicts the ground truth, i.e., a user may be performing barbell curl but the algorithm predicts that the user is performing hammer curl. One of the effective ways to represent this information is by listing the activity and the number of counts predicted by the algorithm against the set.

### 3.5.3 False Positives

This metric is the number of repetitions detected by the algorithm during periods of a workout session tagged as noisiness (no repetitive activity periods). In an ideal case this metric should be zero.

### 3.5.4 Overview

Table 3.5 is constructed for each workout session. In addition to the tabular information, the false positives estimated during the workout session are also computed. Below the table the activity names and the false positive counts

estimated by the algorithm for each of the representative models during the workout are listed.

Table 3.5: Workout Overview

| **User** | Workout Session Identifier | |
|---|---|---|
| **Activity - Set** | True Positive $GT - EC$ | Confusions |

A workout-wise performance evaluation for one of the workout sessions from UIUC's dataset is represented in Table 3.6.

Table 3.6: Overview of a Workout from the UIUC Dataset

| **User - Tim** | 88317ac9-5f3d-4a4d-beea-99f19b41aa33 | |
|---|---|---|
| **Activity-Set** | True Positive $GT - EC$ | Confusions |
| Barbell Curl | **10.0, 6.0** | None |
| Barbell Curl | **10.0, 6.0** | None |
| Barbell Curl | **10.0, 3.0** | None |
| Dumbbell Row | 10.0, 11.0 | None |
| Dumbbell Row | 10.0, 8.0 | None |
| Dumbbell Row | 10.0, 8.0 | None |

**False Positives**
Machine - Walking: 3.0

### 3.5.5 Workout-wise Activity-wise Performance Evaluation

The objective is to evaluate the performance of the representative model. The metrics garnered in the workout-wise performance evaluation (previous section) are viewed differently. The sets of the same activity are viewed collectively. The sets tagged as invalid based on the discussion in Section 3.2.5 are not taken into consideration. An asterisk(*) is used to indicate the representative model. Table 3.7 showcases the outline for the workout-based activity-wise tabular visualization.

Table 3.7: Workout-wise Activity-wise Performance

| **Activity Name** | | | |
|---|---|---|---|
| **No.** | **Identifier** | **GT-EC** | **Conf.** |

The workout-wise performance evaluation for sets across an activity are presented in Table 3.8.

Table 3.8: Workout-wise Activity-wise Performance for Barbell Curl

| **Barbell Curl** | | | |
|---|---|---|---|
| **No.** | **Identifier** | **GT-EC** | **Conf.** |
| 1 | e4c 4c5f * | 10 - 10.0 | None |
| 2 | e4c 7386 | 10 - 9.0 | None |
| 3 | e4c ba18 | 10 - 10.0 | None |
| 4 | dbe 4c86 | 10 - 9.0 | None |

### 3.5.6 Summary

Based on this collective drawn above and drawing from our definitions of average accuracy defined in Section 3.3.1, Table 3.9 is constructed. In the table:

# W is the total number of workouts having the particular physical activity

# S is the total number of valid sets for the particular physical activity

For definitions of Acc. 0, Acc 1 and Acc. 2 see Section 3.3.1

Table 3.9: Outline for the Performance Summary Table

| Activity | # W | # S | Acc. 0 | Acc. 1 | Acc. 2 |
|---|---|---|---|---|---|

Performance summary for the UIUC dataset is shown in Table 3.10

Table 3.10: Performance Summary Table (UIUC Dataset) - Sorted as per Accuracy 2: ±2

| Activity | # W | # S | Acc. 0 | Acc. 1 | Acc. 2 |
|---|---|---|---|---|---|
| Dumbbell Hammer Curl | 19 | 57 | 0.18 | 0.95 | 0.98 |
| Dumbbell Front Raise | 22 | 64 | 0.13 | 0.8 | 0.89 |
| Kettlebell One-Arm Row | 18 | 54 | 0.13 | 0.83 | 0.89 |
| Dumbbell Side Raise | 5 | 13 | 0.08 | 0.62 | 0.85 |
| Dumbbell Flyes | 21 | 63 | 0.17 | 0.68 | 0.78 |
| Barbell Squat | 14 | 39 | 0.23 | 0.49 | 0.64 |
| Barbell Curl | 24 | 72 | 0.1 | 0.51 | 0.61 |
| Dumbbell Shoulder Press | 18 | 53 | 0.09 | 0.38 | 0.45 |
| Dumbbell Lunges | 12 | 37 | 0.08 | 0.22 | 0.38 |
| Dumbbell Rear Lunge | 11 | 33 | 0.09 | 0.21 | 0.36 |

## 3.6 An Overview of False Positives

### 3.6.1 Maximum Count (Max. Count)

Herein the maximum value of the false positive estimated count across workout sessions for an activity is listed.

### 3.6.2 Median

Herein the median value of the false positive estimated count across workout sessions for an activity is listed.

### 3.6.3 Occurrences

The unique values of the false positive estimated counts and their associated number of occurrences across workout sessions for an activity are listed.

To study of false positives in Table 3.11 is constructed, where # W is the total number of workouts having the particular physical activity.

Table 3.11: Overview of False Positives across Workouts

| Activity | # W | Max. Count | Median | Occurrence |
|---|---|---|---|---|

# CHAPTER 4

# REAL TIME VISUALIZATION

The objective of this work is to set up a framework to visualize the motion of a motion sensing device in real time, primarily an implementation of attitude estimation. An attitude is defined as the orientation of a rigid body sensor device with respect to an inertial reference frame.

## 4.1 Framework

The device we selected for developing our framework is TI's SensorTag CC2650. The framework, as shown in Figure 4.1, encompasses the following:

1. **SensorTag CC2650[4]**: A Bluetooth low-energy device with an accelerometer, gyroscope and magnetometer. The device streams out sensor measurements at 10 Hz. SensorTag CC2650 measures the acceleration in G-forces in each of the three axes with a range of ± 8G, the angular velocity around each of the three axes with a range of 250 degrees/sec and the magnetic field in microtesla.
2. **C library**: Low-level byte manipulations on incoming sensor measurements
3. **PyQt application**: Performs mathematical computations for control and OpenGL based 3D visualization as explained in sections 4.2 and 4.3.

### 4.1.1 Sensor Calibration

Magnetometers are essential for gyroscope drift correction. One of the problems in using a magnetometer is that the response surfaces are not ideally centered at the origin in 3D space. This artifact is largely due to the mismatch between the operating condition and the testing conditions during

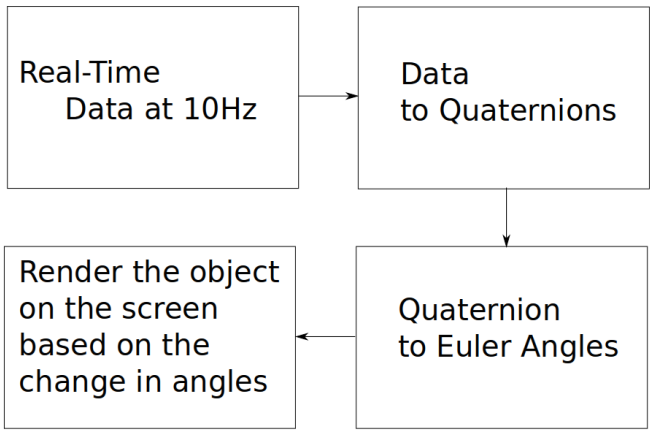


Figure 4.1: Overview of the Framework

manufacturing. As a result, some form of calibration is required prior to the use of magnetometers.

There are two types of corrections involved in the calibration:

1. Hard iron correction

2. Soft iron correction

Hard iron correction is basically to recenter the response to the origin via appropriate shifting. More specifically, the amount of shift is estimated to be the mean of the largest and the smallest calibration data. These offset values (also known as bias) are then subtracted from the real measurements for each axis. Mathematically,

$$(M_i)_{bias} = \frac{(M_i)_{max} + (M_i)_{min}}{2}, \quad (M_i)' = M_i - (M_i)_{bias}, \quad i = x, y, z$$

where $M_i$ and $(M_i)'$ are respectively magnetometer response before and after hard iron correction.

Soft iron correction is dedicated to make the response more spherical through proper scaling in each direction. The scaling factor of each direction is estimated according to Equation (4.1).

$$(M_i)_{scale} = \frac{\sum_{j=x,y,z}[(M_j)_{max} - (M_j)_{min}]}{3[(M_i)_{max} - (M_i)_{min}]}, \quad i = x, y, z \tag{4.1}$$

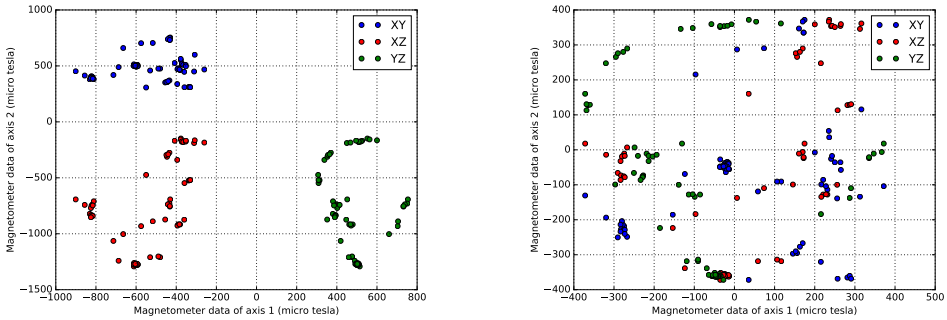


Figure 4.2: Magnetometer Responses in XY, XZ and YZ Planes Before and After Applying both Hard Iron and Soft Iron Correction

The real measurements in each direction are then multiplied by their respective scaling factors. Figure 4.2 shows the magnetometer responses before and after hard and soft iron correction.

## 4.2 Algorithm

The mathematics underlying the 3D visualization are based on the theory developed for attitude estimation using the feedback particle filter (FPF)[5]. FPF is a Monte Carlo estimation algorithm comprising of a particle system. FPF provides for a generalization of the Kalman filter to a general class of nonlinear non-Gaussian problems. It inherits the innovation error-based feedback structure and robustness properties from the widely accepted Kalman filter which has been widely applicable over the past five decades. The procedure for initialization of the particles:

1. Sample $\boldsymbol{X}_i$ from Gaussian $\mathrm{N}(\boldsymbol{0}, \boldsymbol{I}_4)$
2. Normalize: $\boldsymbol{q}_i = \boldsymbol{X}^i / |\boldsymbol{X}^i|$

In Algorithm 1, $N$ is total number of particles for FPF. $\boldsymbol{U}_t$ and $\boldsymbol{Z}_t$ are the gyroscope and the magnetometer measurements in the three axes. $\Delta t$ is the time interval between two consecutive measurements. $\sigma_u$ is the sensor-specific gyroscope noise parameter. $\boldsymbol{q}_t$ is the quaternion vector at time $t$. $\boldsymbol{K}_x$, $\boldsymbol{K}_y$ and $\boldsymbol{K}_z$ is Galerkin gain vector for each of the axis. $\Delta\boldsymbol{\omega}_t$ is the instantaneous angular velocity change at time $t$. $\otimes$ is the quaternion product operator. $R$ is the rotation matrix associated with the quaternion. $\boldsymbol{v}_{ref}$ is the reference model for the magnetometer.

**Algorithm 1** Feedback particle filter using quaternion

**for** each measurement at time $t$ **do**
  **for** $i = 1, 2...N$ **do**
    // Prediction
    $(\Delta\boldsymbol{\omega}_t^i)_1 = \boldsymbol{U}_t\Delta t - \sqrt{\Delta t}\Delta\boldsymbol{U}_t^i, \quad \Delta\boldsymbol{U}_t^i \sim \mathrm{N}(\mathbf{0}, \sigma_u\boldsymbol{I}_3)$
    // Update
    $\hat{\boldsymbol{h}} \approx \left(\sum_{i=1}^N \boldsymbol{h}(\boldsymbol{q}_t^i)\right)/N, \quad \boldsymbol{h}(\boldsymbol{q}_t^i) = R(\boldsymbol{q}_t^i)^T\boldsymbol{v}_{ref}$
    $\Delta\boldsymbol{I}_t^i = \boldsymbol{Z}_t\Delta t - \left(\boldsymbol{h}(\boldsymbol{q}_t^i) + \hat{\boldsymbol{h}}\right)\Delta t/2$
    $(\Delta\boldsymbol{\omega}_t^i)_2 = \boldsymbol{K}_x(\boldsymbol{q}_t^i)\Delta\boldsymbol{I}_{t,x}^i + \boldsymbol{K}_y(\boldsymbol{q}_t^i)\Delta\boldsymbol{I}_{t,y}^i + \boldsymbol{K}_z(\boldsymbol{q}_t^i)\Delta\boldsymbol{I}_{t,z}^i$
    $\Delta\boldsymbol{\omega}_t^i = (\Delta\boldsymbol{\omega}_t^i)_1 + (\Delta\boldsymbol{\omega}_t^i)_2$
    $\boldsymbol{q}_{t+\Delta t}^i = \boldsymbol{q}_t^i \otimes \begin{pmatrix} cos\left(\frac{|\Delta\boldsymbol{\omega}_t^i|}{2}\right) \\ \frac{\Delta\boldsymbol{\omega}_t^i}{|\Delta\boldsymbol{\omega}_t^i|} sin\left(\frac{|\Delta\boldsymbol{\omega}_t^i|}{2}\right) \end{pmatrix}$
  **end for**
**end for**

For detailed mathematical derivation(s), refer to [6].

## 4.3 3D Visualization

After the quaternion is obtained from the controls step, the quaternion is mapped back to the rotation space. The conversion from quaternion to a rotational vector is given by Equation (4.2).

$$\phi = 2arctan(|\boldsymbol{q}_v|, q_w), \quad \boldsymbol{u} = \boldsymbol{q}_v/|\boldsymbol{q}_v| \tag{4.2}$$

where $\boldsymbol{q}_v$ and $q_w$ are respectively the vector and the scalar parts of quaternion $\boldsymbol{q}$. $\phi$ is the rotation angle and $\boldsymbol{u}$ is the axis the object rotates about. It is worth pointing out that the rotation is defined with respect to the sensor frame rather than the global inertial frame.

The 3D object is rendered using OpenGL library written in C++. For this framework, we make use of Python wrappers, namely PyOpenGL and PyQt, to access the functions in the OpenGL library. The code rendering the 3D object is derived from the PyQt4 OpenGL example [7]. Plotting of the gyroscope measurements is purely done via PyQt. Figure 4.3 displays our framework in action.

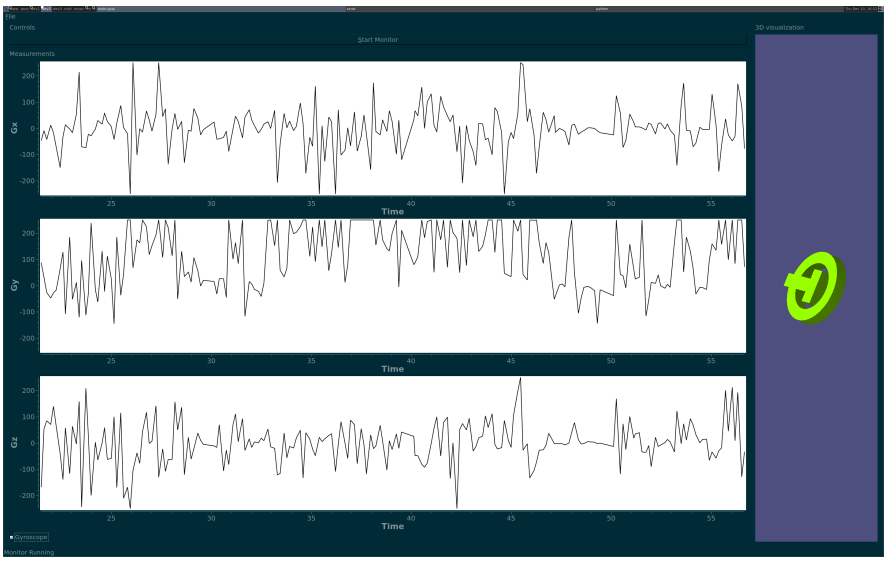


Figure 4.3: Current Framework in Action - Gyroscope Measurements against Rotational Motion of a 3D Object

# CHAPTER 5

# CONCLUSIONS AND FUTURE WORK

Metrics for comparing different algorithms for activity recognition have been developed, and a visualization framework to benchmark the performance of attitude estimation algorithms has been set up.

Future work will focus on extending the proposed metrics and framework for non-repetitive activities such as drawing alphabets and numbers. Another work would be to extend the visualization environment to replicate the translational and rotational motion of the motion sensing device in free space.

# REFERENCES


[1] Centers for Disease Control and Prevention. "Healthy People 2010", http://www.cdc.gov/nchs/healthy_people/hp2010/hp2010_indicators.htm, 2011 Aug., Accessed: 2015-10-23.

[2] H. Montoye, R. Washburn, S. Servais, A. Ertl, J. Webster, and F. Nagle, "Estimation of energy expenditure by a portable accelerometer," *Medicine and Science in Sports and Exercise*, vol. 15, no. 5, p. 403407, 1983. [Online]. Available: http://europepmc.org/abstract/MED/6645869

[3] M. Zhang and A. A. Sawchuk, "USC-HAD: A daily activity dataset for ubiquitous activity recognition using wearable sensors," in *ACM International Conference on Ubiquitous Computing (Ubicomp) Workshop on Situation, Activity and Goal Awareness (SAGAware)*, Pittsburgh, Pennsylvania, USA, September 2012.

[4] "Texas Instruments CC2650 User's Guide," http://processors.wiki.ti.com/index.php/CC2650_SensorTag_User's_Guide, [Online] Accessed: 2015-12-07.

[5] T. Yang, P. Mehta, and S. Meyn, "Feedback particle filter," *Automatic Control, IEEE Transactions on*, vol. 58, no. 10, pp. 2465–2480, Oct. 2013.

[6] C. Zhang, A. Taghvaei, and P. G. Mehta, "Feedback particle filter on matrix lie groups," *arXiv preprint arXiv:1510.01259*, 2015.

[7] OpenGL Example, https://github.com/Werkov/PyQt4/tree/master/examples/opengl, [Online] Accessed: 2015-12-07.