SmartTrainGloves: Monitoring gym training using smart gloves

Yanxiang Huang (Eindhoven University of Technology)

Supervisor: Dr. Oliver Amft (ACTLab, Eindhoven University of Technology)

July 11, 2012

Abstract

Weight training is efficient in keeping body fit. However, the coach for supervising trainer's performance are not always available. In this report, acceleration BodyAnt sensors attached on gloves was utilized to acquire sufficient motion information. This allows athletes and amateurs to train independently. With the connection with smart-phone platform, trainers can get feedback on which exercise is performed, the number of exercise repetitions, the duration of the exercise training, and what can be improved based on the motion of the athletes.

Contents

1 Introduction **2**

1.1 Background 2

1.2 Smartglove gym monitor 2

2 Main functions **4**

2.1 Interested exercise 4

2.2 Functions 4

2.2.1 Counting 5

2.2.2 Exercise recognition 5

2.2.3 Bad-motion recognition 6

3 Data processing **7**

3.1 Data recording 7

3.2 Counting of repetitions 8

3.3 Exercise recognition 9

3.4 Bad-motion recognition 10

4 Result analysis **11**

4.1 Data recording 11

4.2 Counting of repetitions 13

4.3 Exercise recognition 13

4.4 Bad-motion recognition 15

5 Conclusion and Outlook **16**

Chapter 1

Introduction

1.1 Background

Strength training contributes to cardiovascular fitness and overall well-being, and is thus frequently pursued by people attending fitness studies [1]. In order to track training progress and monitor correct exercise execution, expert coaches are usually needed to provide advise. However, coaches are not always available and thus there is a substantial risk for training injuries.

Inertial sensors attached or embedded in gloves could acquire sufficient motion information to allow athletes and amateurs to train independently. When continuously worn during exercise, the gloves could provide progress and performance guidance through personal displays or other forms of feedback.

In order to successfully use sensor-embedded training gloves, such gloves need to forward their motion measurement for processing and feedback, e.g. visualisation of performance and skills information. As smart phones have become a commonly available personal computing platform, phones could be an ideal complementary device for the smart gloves [2].

1.2 Smartglove gym monitor

This project deployed inertial motion sensors with 3-D accelerometer (the BodyANT devices [3]) at gloves for both hands and interface these units with a Android smart-phone. An example exercise using the sensors and gloves is shown in Figure 1.1 and Figure 1.2.

The algorithm proposed in this report are tested for data recording in a gym with different training devices. Among the analysis goals, several training parameters can be detected:
1. Which exercise is performed in a set of 5-10 exercises.

Figure 1.1: *Set-up of BodyANT sensor equipment, and first exercise Front Raise*

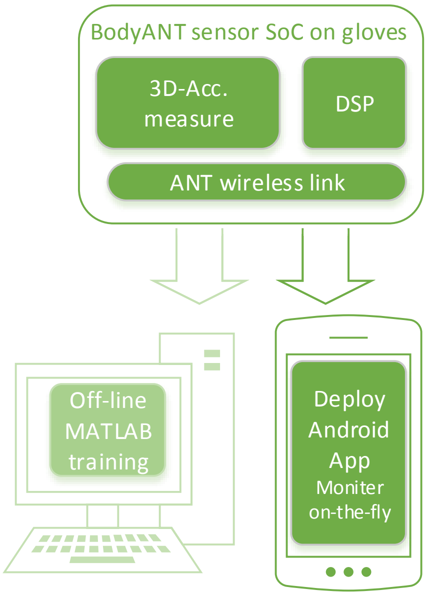

Figure 1.2: *System processing flow. The BodyANT devices [3] measures body movement data, transmit them to a PC for training, and transmit them to smartphones for application.*

2. The number of exercise repetitions and the duration of the exercise training (the time of muscle under load).
3. How is the exercise performed (good performance vs. errors).

Chapter 2

Main functions

This chapter describes the main functions SmartTrainGloves can achieve as a guidance to gym exercises, including what are the interested exercise.

2.1 Interested exercise

Since the sensor that collecting the 3-D acceleration data is attached on the gloves, naturally, the most suitable exercise to measure should be those with significant hand movement, which corresponds to upper body exercises like chest and shoulder exercises. Firstly, all the weighted upper body exercise from reference [4] and [5] were listed. Then, with the advices of coach El Allouche, A. from TU/e sports center, 10 most popular exercises were chosen. The detailed information of how to perform these activities can be found in reference [6].

These exercises are: (1) *Front Raise (FR)*; (2) *Lateral Bicep Curls (LBC)*; (3) *Shoulder Abduction (SA)*; (4) *Forearm Curl (FC)*; (5) *Chest Press (CP)*; (6) *Chest Fly (CF)*; (7) *Reverse Forearm Curl (RFC)*; (8) *Preacher Curls (PC)*; (9) *Triceps Kick Backs (TKB)*; (10) *High Cable (HCL).*

With these exercises, one can exercise almost all muscles in upper body, which makes this application very useful as a virtual gym exercise coach. For example, *Front Raise* can exercise the shoulder, *Lateral Bicep Curls* can exercise Bicep mussels, *Triceps Kick Backs* can exercise Triceps. In addition, the hand motion of the 10 exercises are quite different from each other, which makes it possible to recognize them.

2.2 Functions

Firstly, a trainer has to wear the gloves with the sensor. Then, the trainer can start the smartphone application. From then on, the application will record the

Figure 2.1: *Exercises from No. 2 to No. 10. Refer to Interested exercises list*

motion data for the trainer. After finishing the exercise and clicking the *finish* button, the application will process the data that achieve the following functions: (1) counting, (2) exercise classification, and (3) bad-motion recognition.

2.2.1 Counting

The application will count the number of exercise repetitions. In addition, it provides the total training repetitions as well as the duration for each time. This is very useful because in some exercise it can help the trainer to exercise in a moderate speed.

2.2.2 Exercise recognition

It will also tell the trainer which exercise (from these 10 listed before) is performed. This will help the system automatically record the training log for each exercise.

2.2.3 Bad-motion recognition

For some exercises, it is important to keep the motion right. Otherwise, the muscles won't be well trained as desired, or even get injured. In these circumstances, the guidance of a coach is necessary. The SmartTrainGloves can relief the workload of the coach by giving some simple advices based on the athlete's error.

Chest Fly is chosen to demonstrate our capability in dealing with these problems and developed corresponding function. There are 3 rules to follow when doing *Chest Fly*: (1) wrist angle should always be kept at 180 degrees; (2) elbow angle should always be kept at 135 degrees; (3) the maximum shoulder angle should be 180 degrees.

This application can detect athletes' error and give feedback on it.

Chapter 3

Data processing

This Chapter discusses the data processing procedure of this application. In addition, the theory of each procedure is introduced.

The flowchart in Figure 3.1 describes the data processing flow for this application. The program starts when user start the recording application. From then on, the program will keep recording the 3-D acceleration data from the sensor. Once the exercise is finished, it will start the processing procedure. Firstly, counting is conducted by detecting peaks of the data, with one peak representing one repetition. Data is divided into clips with acceleration signal peaks acting as the borders for each clip. Secondly, each clip of data will be recognized to one of the 10 categories mentioned in Chapter 2. In this way, the exercise can be detected. Finally, this application will detect the bad motion in the exercise by comparing it with each item from the error model library.

3.1 Data recording

In this project, we use two wireless sensor nodes called BodyANT to collect the 3-D acceleration data [3]. Each sensor can measure the acceleration data at a sampling rate of 16 Hz. At the mean time, it will send the data at a package rate of 32 Hz by local communication protocol called ANT. In each package, the data

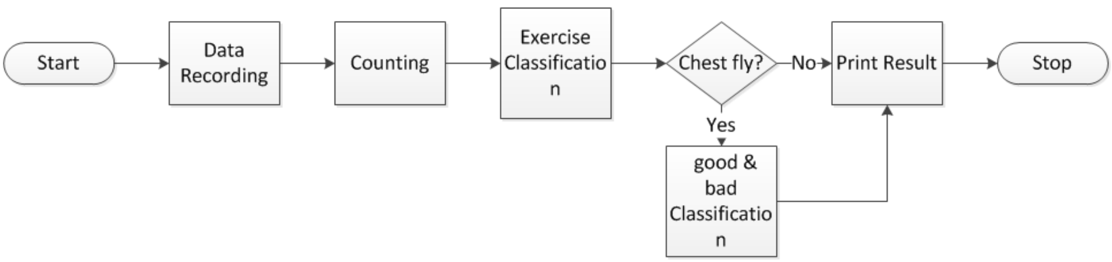

Figure 3.1: *Flowchart of data processing*

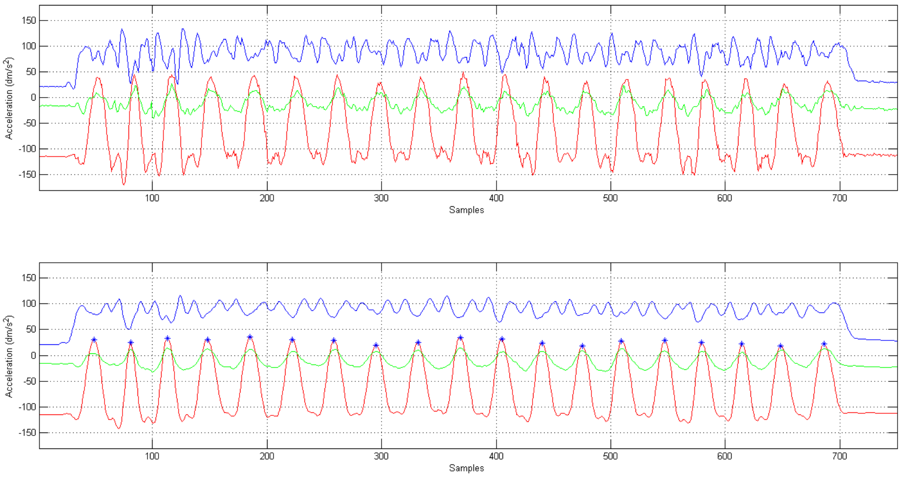

Figure 3.2: *Data of Front Raise for left hand. The upper figure shows raw data of Front Raise; the lower figure shows data of Front Raise after taking the mean of sliding window; and with counting results. Red lines represents data for x-axis; green lines for y-axis; Blue lines for z-axis; red dots for peaks.*

include: (1) acceleration in x axis, (2) acceleration in y axis, (3) acceleration in z axis, (4) temperature, (5) battery voltage, (6) a correction counter data that will be added for each sample. The reason why each data package are sent twice is to ensure every package can be received in case of data loss due to wireless channel problems.

When the smart-phone receives data from the sensors by pre-allocated channels, it will check the counter for each package. If the correction counter data is the same compared with the one from last package, which means the package is already received before, then the package will be omitted. Otherwise, the package is a new package and will be recorded.

When the user starting the recording application, every data will be recorded on the temporary memory for following-up processing. In addition, the packages will also be stored in a file named with the starting time of recording for further analysis, if necessary.

3.2 Counting of repetitions

It should be noted that data from one axis is enough for counting. Thus, the program will calculate the variance of the data for each axis, then only the axis with the largest variance will be selected as the interested axis for counting, because the range of data is larger, which increases the accuracy. It can be summarized from the recorded data of *Front Raise* that, counting is nothing but finding out all the local peaks (or equivalently, valleys). Because there will always be one peak/valley in each repetition, which corresponding to the largest/lowest acceleration.

However, it is difficult to pick out all the peaks/valleys wisely since there are some local maxima/minima produced by noises. There are two general solutions for this, of which both are implemented in this project. Firstly, the average for each axis was calculated in order to smooth graphical representation. To do this, raw data was divided into sliding windows with the window size of 8 samples and step size of 1 sample, after that the mean in each window was calculated. With this step, local maxima/minima are significantly reduced. Secondly, based on the smoothed data, the hill climbing algorithm [7] was deployed to make sure local maxima/minima is avoided. The variance of the data from the interested axis is be calculated, then multiplied with 0.8 as the parameters of positive and negative magnitude threshold value for search in the hill climbing algorithm. The multiplying factor 0.8 is get by testing.

In addition, once the peaks are located, the duration of each repetition can also be calculated easily. After that, data are divided into clips with peaks acting as borders of each clip (repetition) to study the performance of each repetition.

3.3 Exercise recognition

In [8], different features and window lengths are studied to recognize the activities. They showed that the best performance is achieved when different window lengths and features are chosen separately for each activity. However, it might be too costly to automotive adjust window lengths. In this work, we use the window size of 8 sample and step size of 1 sample, just as it was done for counting.

As the data from the sensors on left and right hands are treated independently, both hands can be used to determine the exercise category. The flowing charts are based on the data from right hand's sensor. For each clip produced by counting stage, the mean and variance of acceleration can be calculated for each axis. Thus, 6 features (X_m (mean of x), X_v (variance of x), Y_m, Y_v, Z_m, Z_v) can be used to perform Naive Bayes classification. Although the assumption of using Naive Bayes classification that each features are independent may not be satisfied in this project, it can still be deployed since Naive Bayes sometimes produces some meaningful results and solve the problem in the real world. For each test clip, $P(C_i|test)$ represents the posterior of a test data to be recognised as a category (C_i),

$$P(C_i|test) = \frac{P(C_i)P(X_m|C_i)P(X_v|C_i)P(Y_m|C_i)P(Y_v|C_i)P(Z_m|C_i)P(Z_v|C_i)}{evidence},$$

$$evidence = \sum_{i=1}^{n} P(C_i)P(X_m|C_i)P(X_v|C_i)P(Y_m|C_i)P(Y_v|C_i)P(Z_m|C_i)P(Z_v|C_i)$$

Since evidence is all the same for every category, to classify each test clips, only the numerator of $P(C_i|test)$ is of interest. Besides, an assumption is made that $P(C_i)$ for each category are the same, so it is set to 1/n (n=10), which means that the trainer will perform each of the ten exercise at the same possibility. Another assumption is that each features in certain category are Gaussian distributed. For example, the mean acceleration in x axis are normal distributed for different clips of *Front Raise*. So $P(X_m)$ is the PDF of the Gaussian distribution. In order to get the PDF, a lot of data are recorded as training data to get the mean (μ) and variance (σ^2) for each feature (e.g. X_m). Thus

$$P(X_m) \sim \mathcal{N}(\mu, \sigma^2)$$

The others like $P(X_v)$ can be acquired by the same approach. For each testing clip, the posterior of all category are examined, and the category with the largest posterior is the recognition result. The final result of the recognition is made by integrating all the result of each clip.

3.4 Bad-motion recognition

The approach we use in exercise recognition is also applied in bad-motion recognition. When the classification result shows the performed exercise is *Chest Fly*, each testing clip will be recognised to one of the following seven exercise model categories:

- (1) The performance is good, wrist angle is 180 degrees; elbow angle is 135 degrees; maximum shoulder angle is 180 degrees. (later call it *Correct* for short)
- (2) wrist angle is too small (*WS*), i.e. 135 degrees;
- (3) wrist angle is too large (*WL*), i.e. 225 degrees;
- (4) elbow angle is too small (*ES*), i.e. 90 degrees;
- (5) elbow angle is too large (*EL*), i.e. 180 degrees;
- (6) maximum shoulder angle is too small (*SS*), i.e. 135 degrees;
- (7) maximum shoulder angle is too large (*SL*), i.e. 225 degrees;

If the performance is recognised to the first category, then the performance is good. Otherwise, the performance is bad due to the certain the reasons corresponding to the recognised category.

Chapter 4

Result analysis

In this chapter, the processing result will be discussed. Firstly, data loss rate is measured to discuss how reliable the raw data is. Then, the accuracy of counting of repetition is examined. After that, confusion matrices are used to verify recognition result on both what exercise is performed and what the error is.

In order to get the data for processing, we tested with a young male athletes. The athletes was asked to perform each exercise for 20 repetitions. With the guidance of a professional coach, he performed for every motion models of *Chest Fly*, each for 20 repetitions. to analysis the recognition result, the data was cut to 2 sets, with 10 repetitions as training data, and the other 10 repetitions as testing data.

4.1 Data recording

All the recording data has been stored to further analysis, from which the data loss rate can be calculated. According to the data loss rate graph (Figure 4.1 to 4.10), the average loss rate for all exercise is no more than 0.5 %, indicating

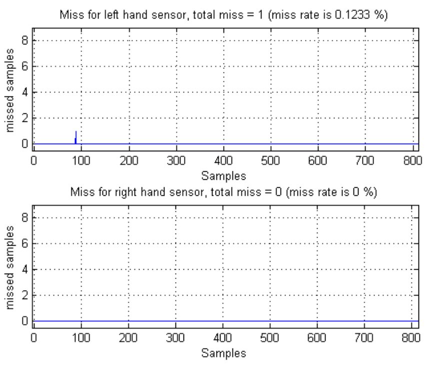

Figure 4.1: *Data loss rate of (1) Front Raise*

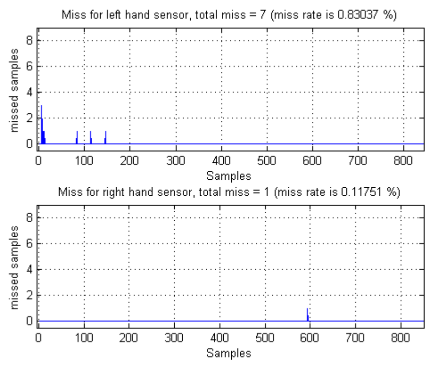

Figure 4.2: *Data loss rate of (2) Lateral Bicep Curls*

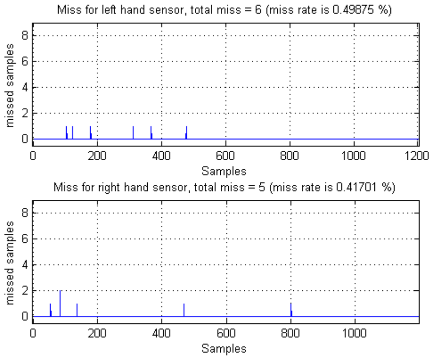

Figure 4.3: *Data loss rate of (3) Shoulder Abduction*

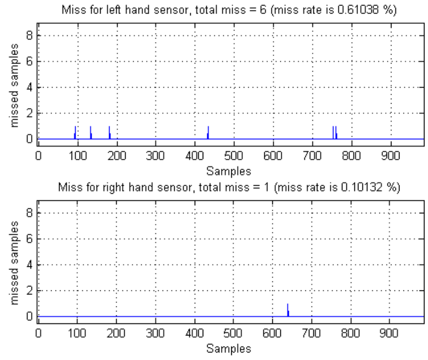

Figure 4.4: *Data loss rate of (4) Forearm Curl*

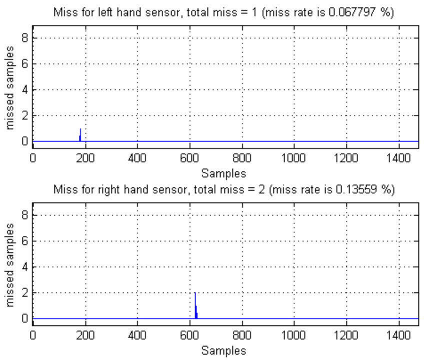

Figure 4.5: *Data loss rate of (5) Chest Press*

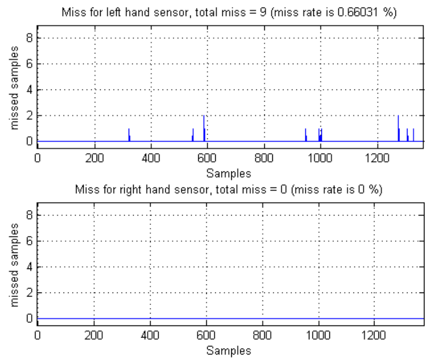

Figure 4.6: *Data loss rate of (6) Chest Fly*

that it is a reliable way for data recording.

However, the data loss rate for right hand of (8) *Preacher Curls* is as high as 4.5 %, which is mostly because the training equipment shields the wireless signal from the recording smart-phone, which was located at the athlete's left hand side.

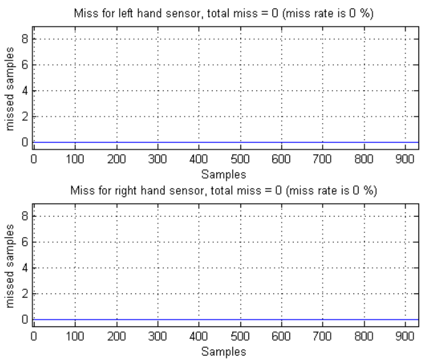

Figure 4.7: *Data loss rate of (7) Reverse Forearm Curl*

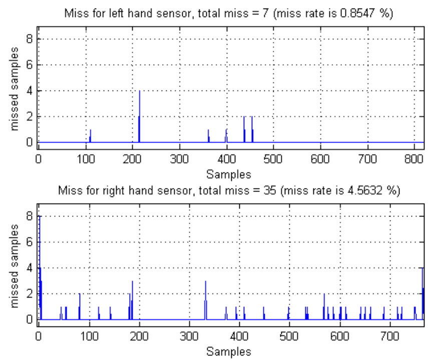

Figure 4.8: *Data loss rate of (8) Preacher Curls*

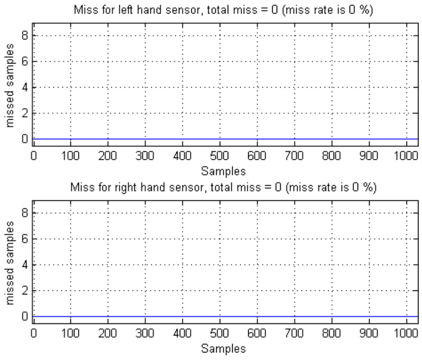

Figure 4.9: *Data loss rate of (9) Triceps Kick Backs*

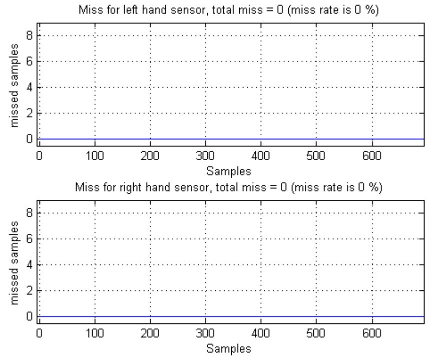

Figure 4.10: *Data loss rate of (10) High Cable Low*

In addition, the antenna of the sensors is pointed to the body rather than the air so that the battery can be changed easily. However, the battery then acts as a shield to the wireless signal, making the signal strength low. Thus, future design can be improve the data loss rate, at the cost of more difficulties when changing the battery, by turning the sensor around so that the antenna points to the open air.

4.2 Counting of repetitions

Based on the comparison of counting result of all the 10 exercises with recording log, the accuracy of counting result is 100 %. Since a high accuracy is reasonable because the hill climbing algorithm is reliable once the threshold parameters are set properly.

However, the trainer is required to stop recording as soon as he finishes exercise motion. If not, he might move their hands arbitrarily for relaxing after finishing the exercise, like most trainers will do, then some extra data which has nothing to do with the exercise will be added to the end of the data, making the counting result inaccurate.

4.3 Exercise recognition

Data was recorded when a trainer performs 10 exercise, each with 20 repetitions. We used the first 10 repetitions as training data, and the last 10 repetitions as testing data to test the accuracy of recognition.

Figure 4.11 shows the posterior matrix of 15th repetition. To make it more readable, the posterior are transformed to -log(posterior) in the figure. So the smaller the values are, the larger the posteriors are. The recognition result of this sample, which is done by picking up the category with the largest posterior, shows an

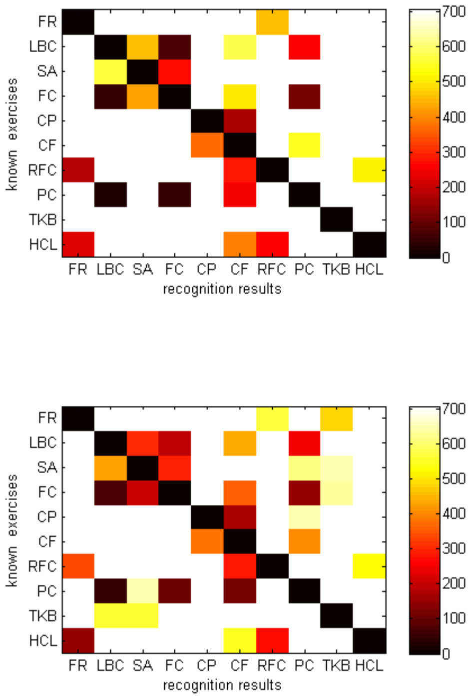

Figure 4.11: *Posterior matrix of exercise recognition for one sample (clip). Upper figure is for left hand sensor data, lower figure is for right hand sensor. Y axis represents categories data belongs to, X axis represents recognition results.*

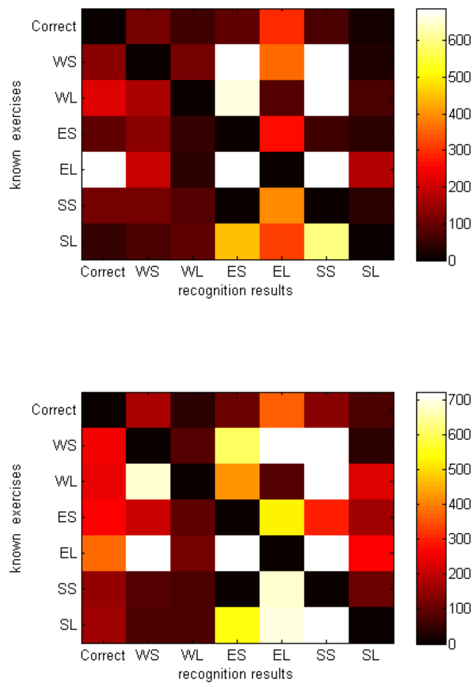

Figure 4.12: *Posterior matrix of typical error recognition of Chest Fly for one sample. Upper figure is for left hand sensor data, lower figure is for right hand sensor. Y axis represents categories data belongs to, X axis represents recognition results.*

accuracy of 20 out of 20 (10 exercise times two hands). However, posteriors of exercise *Lateral Bicep Curls (LBC)*, *Shoulder Abduction (SA)*, *Forearm Curl (FC)*, and *Preacher Curls (PC)* are close to each other. That is because their physical motions are quite similar. Especially for *Lateral Bicep Curls (LBC)* and *Preacher Curls (PC)*, the posteriors are even more similar since the motions have no difference except for the motion range of *Forearm Curl (FC)* is larger.

Then, we used the last 10 repetitions as training data, and the first 10 repetitions as testing data to test the accuracy. The total accuracy of these two groups of recognition result is 351 out of 360 (97.5 %), which is shown in Figure 4.13.

Finally an overall recognition was done by adding the recognition result for each repetition together to build a frequency distribution, and by choosing the category with highest frequency as the final recognition result. In this way, the accuracy of recognition result is 40 out of 40 (100 %), which is constituted by 10 exercises times 2 hands times 2 sets of training data.

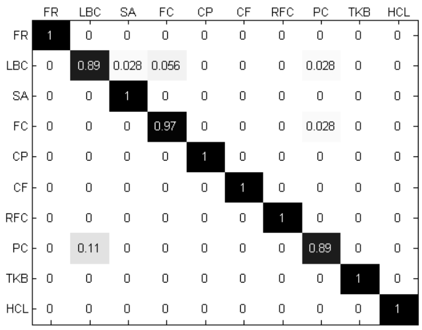

Figure 4.13: *Confusion matrix of exercise recognition result. Y axis represents categories data belongs to, X axis represents recognition results.*

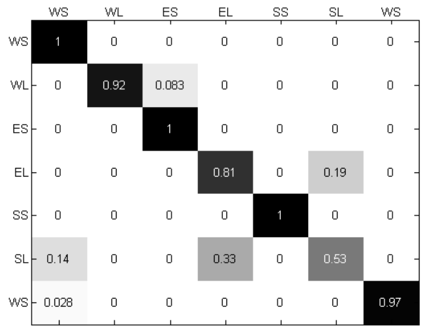

Figure 4.14: *Confusion matrix of Bad-motion recognition result. Y axis represents categories data belongs to, X axis represents recognition results.*

4.4 Bad-motion recognition

A similar test approach can be applied to bad exercise recognition. We recorded the data for the 7 models in Chapter 3.4 with 20 repetitions. Then the data of the first 10 repetitions was used as training data, and the data of the last 10 repetitions was served as testing data for recognition. After that, the training and testing data sets was swapped.

Figure 4.12 indicates the posterior matrix in motion models recognition of *Chest Fly* for the data of 13th repetition, when the data of the first 10 repetitions was used as training data. The category with largest posterior is the most likely bad-motion model, and the second largest posterior gives the second likely bad-motion model. For example, according to the posterior matrix for the left hand sensor in Figure 4.12, the biggest motion problem of the known data (SS) is *the maximum shoulder angle is too small (SS)*, and the second problem is *the elbow angle is too small (ES)*.

The overall accuracy in the recognition of the most likely model is 224 out of 252 (88.9 %), according to Figure 4.14. The result is not so satisfying comparing with the exercise recognition, because the motions are more alike in this case. The lowest the bad-motion recognition accuracy lies in distinguishing *elbow angle is too small (ES)* and *maximum shoulder angle is too small (SS)*, since the wrist motions of these two categories are almost identical. To solve this problem, additional sensors are needed, e.g. attached to the arms.

It should be noted that the data of mean and variance of bad-motion models are very user-dependent, to apply it to other users more accurately, user-specific training is needed [9]. While in this project, only one athlete's data was recorded and tested for bad-motion recognition.

Chapter 5

Conclusion and Outlook

This system works well in terms of recording data (with the average data loss rate less than 0.5 %), counting the repetitions, and classifying exercises (with an ideally 100 % accuracy). It is useful as an assistance in tracking gym exercises and giving feedback to motion performance.

In addition, it can still be improved in several parts:

Firstly, because orientation sensors provide more information on the type of exercise, and are less influenced by the motion speed. We could change the accelerometers by orientation sensors, although orientation sensors are larger and more expensive [10].

Secondly, in this program, athletes has to click the smart-phone to start and stop recording, which is not accurate for counting and recognition. It is better to have a detecting algorithm for annotation, so that the program can get the exact start and end time, and make it more accurate.

Thirdly, currently, the data processing will not start until the exercise is finished. It could be better if it can offer feedback in real-time instead of after the exercise finishes.

Finally, to improve exercise (or bad-motion) recognition accuracy, additional sensors are needed, e.g. attached to the arms, to collect data from other parts of the body.

Reference

[1] Eric Dishman. Inventing wellness systems for aging in place. *Computer*, 37(5):34–41, May 2004.

[2] Oliver Amft and Paul Lukowicz. From backpacks to smartphones: Past, present and future of wearable computers. *IEEE Perv Comput*, 8(3):8–13, July–September 2009. Wearable Computing Department.

[3] Martin Kusserow, Oliver Amft, and Gerhard Tröster. Bodyant: Miniature wireless sensors for naturalistic monitoring of daily activity. In *Bodynets 2009: Proceedings of the 4th International Conference on Body Area Networks*, page 1–8. ICST, ICST, 2009. ISBN:978-963-9799-41-7.

[4] Wikipedia. List of weight training exercises.

[5] F. Delavier. *Strength Training Anatomy.* Sports Anatomy. Human Kinetics, 2010.

[6] D. Johnson-Cane, J. Glickman, and J. Cane. *The Complete Idiot's Guide to Weight Training.* Complete Idiot's Guide to. Alpha Books, 2002.

[7] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach.* Prentice Hall, 2 edition, 2003.

[8] Tâm Huynh and Bernt Schiele. Analyzing features for activity recognition. In *Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies*, sOc-EUSAI '05, pages 159–163, New York, NY, USA, 2005. ACM.

[9] Stephen S. Intille, Ling Bao, Emmanuel Munguia Tapia, and John Rondoni. Acquiring in situ training data for context-aware ubiquitous computing applications. In *Proceedings of CHI 2004 Connect: Conference on Human Factors in Computing Systems*, pages 1–8. ACM Press, 2004.

[10] A.Y. Benbasat. *An Inertial Measurement Unit for User Interfaces.* Massachusetts Institute of Technology, Program in Media Arts & Sciences, 2000.