Measuring movement of golfers with an accelerometer

CHANGSU JUNG

Master's Thesis at ICT
Supervisor: Stina Nylander
Examiner: Professor Markus Hidell

Master of Science Thesis
Stockholm, Sweden 2012

Abstract

The purpose of this thesis is to analyze a golfer's movement and provide the feedback related to the golfer's skill with the simple and novel ways. A golfer can easily measure golf swings and get feedback based on his performance using an Android smart phone without expensive or complicated devices.

In this thesis, we designed and implemented an Android application using an accelerometer sensor to analyze swing data for identifying critical points and to give various kinds of feedback based on the data. The feedback helps golfers to understand their swing patterns, timing and speed so it makes them improve their skills.

Keywords

Golf swing analysis, feedback, accelerometer, Android application, acceleration

Sammanfattning

Syftet med detta arbete är att analysera en golfspelares rörelse och ge feedback relaterad till golfspelarens färdigheter med enkla och nya sätt. En golfspelare kan lätt mäta golf svängningar och få feedback baserad på hans prestanda när den använder en Android smartphone utan dyra eller komplicerade enheter.

I detta arbete har vi utformat och implementerat en Android applikation med en accelerometer sensor för att analysera svängnings data och för att identifiera kritiska punkter och ge olika typer av feedback baserat på data. Detta feedback hjälper golfspelaren att förstå sitt svängnings mönster, timing och hastighet så det gör dem att förbättra sina färdigheter.

Contents

List of Figures

List of Tables

1. Introduction

1.1 Motivation

Nowadays, many studies have been carried out in the field of sports to measure athletes' movements and provide real-time feedback to improve their skills using various equipments. Through these approaches, athletes can realize their motions better and get information of their movements which they do not notice while playing and practicing [1]. This information makes them enhance their performance and correct their problems immediately.

In golf, there are already many kinds of equipments and measurement systems for analyzing a golfer's swing form and movement [2, 3, 4, 5]. However, these approaches demand many sensors to be attached to a golfer's body or additional expensive hardware such as high-speed cameras, 3-D gyro sensors and motion sensors [6]. Therefore, these systems have been used for elite players to level up their skills and for scientific research. However, as the popularity of golf is increasing for leisure among individuals, the desire of improving their activities such as swing form and their score using low-cost devices and more convenient ways have become strong [6]. Fortunately, various sensors such as an accelerometer, a gyro-sensor, and a camera and are embedded in a smart phone so it can satisfy people's needs with low-cost device which is handy and convenient.

Moreover, we decided to use an accelerometer for measuring a golfer's movement because one of the main interests for the majority of golfers is how to increase distance when they hit balls [5]. For longer distances, a golfer should obtain maximum club-head speed at impact timing. If a golfer can generate greater acceleration on his club at impact, the force is delivered to a ball and it affects the distance [7].

$$\text{Force} = \text{Mass} \times \text{Acceleration} \tag{1}$$

In this thesis, an Android application was designed and implemented to analyze a golfer's acceleration data and provide feedback. Moreover, this method provides a very simple and convenient way for individuals to comprehend their movement and improve their performance using an Android smart phone.

1.2 Goals

This project consists of developing a sensor-based prototype for golfers using an embedded accelerometer in an Android smart phone and providing different kinds of feedback with respect to participants' performance.

The main goals of this thesis are below:

- Design and implementation of software functionalities for extracting core information from accelerometer data
- Design and implementation of software modules for identifying critical points of the swing movement.
- Implementation of different types of feedback for the identified critical points.
- Evaluation of software performance and accuracy of identifying critical points.

1.3 Delimitation

This work was designed and implemented using only an accelerometer inside of a smart phone. Therefore, there are some limitations in analyzing the exact golf swing motions and calculating the exact swing duration without additional equipments.

1.4 Thesis structure

This thesis begins with the brief statement of problems of previous researches and introduction of this thesis in Chapter 1.

Chapter 2 consists of the related works about measurement system in golf.

Chapter 3 is composed of the analysis of golf swings using the previous work in SICS project.

Chapter 4 describes the design of an Android application for measuring a golfer’s movement.

Chapter 5 provides the implementation of the software and explains algorithms for detecting critical points.

Chapter 6 consists of evaluation and the analysis of the result.

Chapter 7 includes the conclusion of this thesis and future works.

2. Background and Related work

2.1 Background

2.1.1 Acceleration

The definition of acceleration is "*the time rate of change of velocity with respect to magnitude or direction*" [9].

$$\text{Acceleration}(\text{m/s}^2) = \frac{\Delta\,\text{Velocity}(\frac{\text{m}}{\text{s}})}{\text{Time}(\text{s})} \tag{2}$$

With the definition, acceleration is changed if there are variations in the velocity of an object such as increase or decrease and also if a moving entity changes its direction to a positive or negative. In addition, if acceleration is negative, it does not simply mean that the velocity of an object is decreasing. The object's velocity is either increasing to a negative direction or decreasing to a positive direction [10].

2.1.2 Accelerometer and the coordinate system of Android

The accelerometer measures acceleration which is generated by the movement and the change of velocity [11]. The measured data consist of three floating point numbers, which are the x-axis value for horizontal movement, the y-axis value for vertical movement and the z-axis value for the movement of the direction towards the outside of the screen [12]. Figure 1 illustrates two coordinate systems, which are the sensor coordinate system and the screen coordinate system.

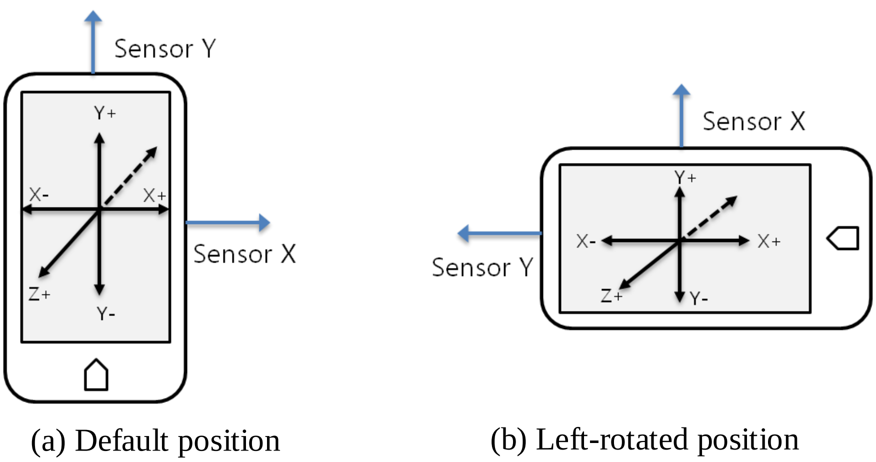


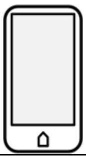
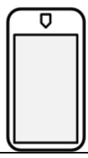
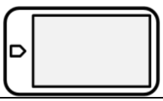
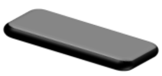
Figure 1. The two coordinate systems of the sensor and screen [12, 13]

When a smart phone is in the default position like Figure 1(a), the two coordinate systems are the same. However if the smart phone is rotated to the left side, the two systems indicate different directions as shown in Figure 1(b). The screen coordinate system which is provided by the Android sensor APIs is changed whenever the smart phone is rotated. The upper side of the screen is allocated for the positive y-axis and the right side of it is assigned to the x-axis.

On the contrary, the sensor coordinate system is fixed regardless of rotation. The vertical direction in the default position is set to the y-axis and the horizontal direction is appointed to the x-axis like Figure 1(a). When the smart phone is rotated like Figure 1(b), the two axes point to the same directions which are allocated in the default position. Therefore, the position of the smart phone should be considered and the two coordinate systems should be remapped appropriately related to the position while collecting acceleration data [13].

Moreover, an accelerometer has different acceleration values along the three axes according to the smart phone's positions because of the gravity applied to the smart phone. When the smart phone is in Up position, the common position, the acceleration value of the y-axis is 9.81 m/s^2 and when the position is Left, the value of the x-axis is 9.81 m/s^2 as illustrated in Table 1 [14]. Therefore, the acceleration values should also be adjusted during data collection procedure considering its position.

Table 1. Acceleration values for different positions [14]

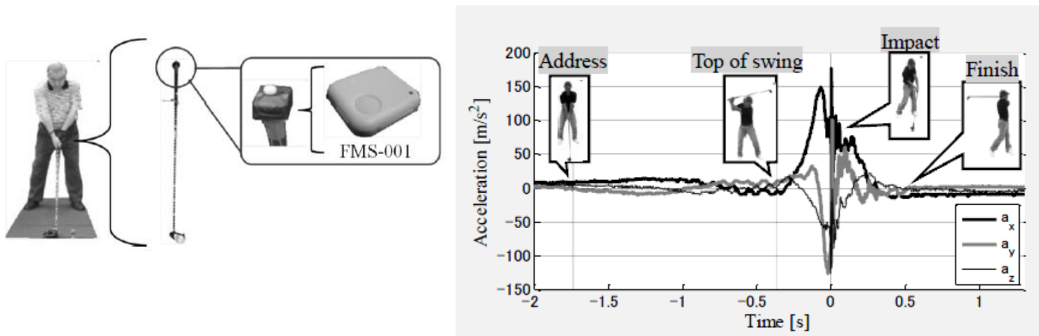
Position		X	Y	Z
Up		0	9.81 m/s^2	0
Left		9.81 m/s^2	0	0
Down		0	-9.81 m/s^2	0
Right		-9.81 m/s^2	0	0
Front up		0	0	9.81 m/s^2
Back up		0	0	-9.81 m/s^2

2.2 Related work

Many scientific approaches were already carried out to measure and analyze golf swings using various measurement systems. In general, kinematical measurements of golf sport are divided into two types as contact and non-contact types [4]. Non-contact type uses images and videos captured by high-speed cameras for analyzing the motion of a golfer's body and the movement of a golf club. On the contrary, the contact type adopts devices like accelerometers and gyroscope sensors by means of data collection. The devices are attached to the player's body and calculate his kinematic movements [4]. This section illustrates some examples of the measurement systems.

2.2.1 Motion sensor

There is one example of using a motion sensor embedded with a three-dimensional accelerometer and a gyroscope sensor to measure golf swing movement at the grip [6]. The motion sensor was placed on top of a club header as shown in Figure 2(a) and collected acceleration and angular velocity. The collected data was delivered to a PC through a Bluetooth connection to be analyzed. The data was compared with the motion capture system to evaluate the performance. The result showed that this motion sensor could identify golf swing motions as the address, top of swing, impact and finish as described in Figure 2(b).



(a) The motion sensor (b) The graph of acceleration data

Figure 2. The measurement system using a motion sensor [6]

2.2.2 Three-dimensional gyro sensor

Gyro sensors are used to calculate angular velocities of an object. In golf sport, this sensor is used for analyzing a golfer's rotational movement and the sensors are attached to a few parts of the body.

An earlier paper measured and analyzed the golf swing movement using gyro sensors [4]. In the paper, they attached seven gyro sensors to the body and placed a microphone and a high-speed camera as illustrated in Figure 3. They measured angles and maximum displacement of the body, namely, the waist, shoulder, and head.

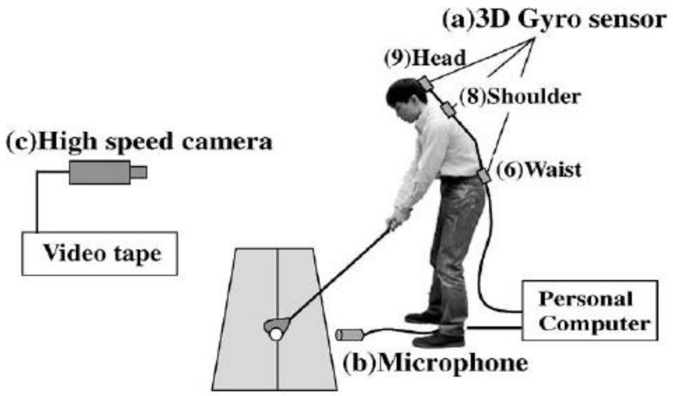
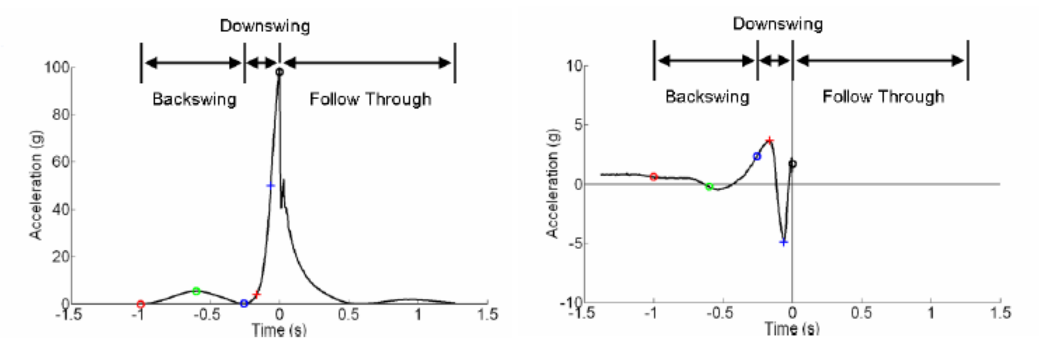


Figure 3. The measurement system using three-dimensional gyro sensor [4]

2.2.3 Accelerometers in the shaft or club head

An earlier paper [8] investigated golf swings using two accelerometers, which are attached to the golf club head and the grip. An accelerometer was located on the club head and measured the acceleration of the club head. This data was used for analyzing the patterns of the golf swing and identifying points in the swing, which are the start point, the changing point from the backswing to the downswing and impact point as illustrated in Figure 4(a). The other one was employed for interpreting the movement of the hands like Figure 4(b). Also, the paper identified and calculated swing duration such as the backswing, downswing, and follow-through using accelerometers. This research is based on Sonic Golf System-1 [15].



(a) The acceleration of the club head (b) The acceleration of the hands

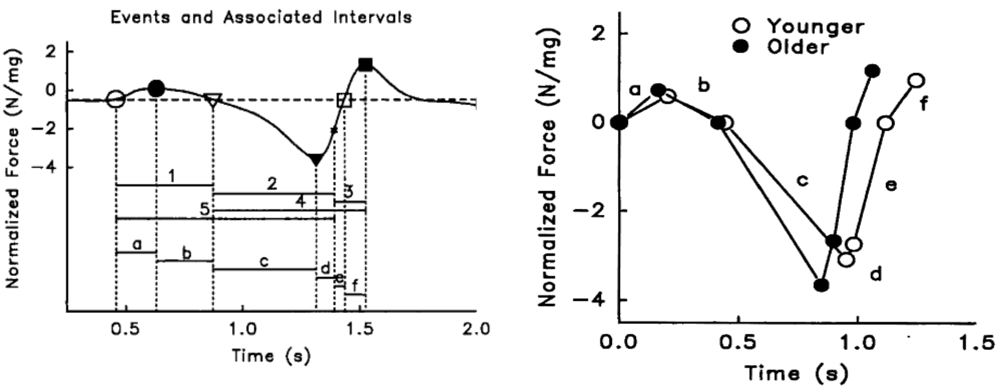
Figure 4. The acceleration graphs measured by two accelerometers [8]

The Sonic Golf System-1 provides the seamless real-time sound feedback using music tones according to the swing speed. For example, if the golfer's swing is fast, he can hear a high-pitched tone. The system helps the golfer understand his swing tempo and rhythm through the sound feedback [19].

2.2.4 Measurement of tempo and rhythm using an accelerometer

A scientific research had been made to find the difference between older and younger golfers in terms of tempo and rhythm using an accelerometer [16]. An accelerometer was placed on the back side of the club head to check the applied force and also, the study calculated the six time durations (***a***-***f***) of the swing as shown in Figure 5 (a). The durations are below.

- ***a***: the duration from the beginning of the swing to the maximum value during the backswing
- ***b***: the interval from the maximum value of the backswing to the transition point to the downswing
- ***c***: the period from the beginning of the downswing to the minimum value of the downswing
- ***d***: the duration from the minimum value of the downswing to the impact point
- ***e***: the interval from the impact to the transition point from negative value to positive
- ***f***: the period from the transition point to the maximum value during follow-through



(a) The six intervals (b) Average force-time graph

Figure 5. The measurement of force and time using an accelerometer [16]

There were some characteristics in the swing durations between the two groups as described in Figure 5(b). The older group's swing was slower only in the interval

d but the younger golfers took longer time in the intervals ***e*** and ***f***. This research illustrated that the intervals ***d***, ***e*** and ***f*** are related according to the age groups [16].

2.2.5 Measurement of rotation angles and weight transfer patterns

A research was completed to measure the different patterns regarding a golfer's trunk rotation and weight transfer. In this study, the moving sequences of the body and weight shift were emphasized as the most important elements in the golf swing to get the optimal club head speed. The rotation angles of trunk and pelvic were gauged by high-speed cameras and video analysis system and the two-place force platform system was used for measuring the weight shift patterns [2].

The subjects for this study were divided into two groups; the skilled golfers (handicap less than 5) and low skilled golfers (handicap between 20 and 36).

The research identified big differences between the two groups during the backswing, downswing and follow-through motions. The skilled golfers had bigger horizontal rotation angles of the upper body during the backswing. On the contrary, the low skilled group showed greater horizontal angles of the upper trunk in the follow-through motion. In the pelvic horizontal rotation angle's case, the skilled group had bigger rotation angles during the backswing and downswing period.

In terms of the weight transfer patterns, the skilled group showed earlier weight shift to the right foot than the low skilled group during the backswing. In addition, the skilled golfers had quicker weight transfer to the left foot than the low skilled golfers during the downswing.

2.2.6 Measurement of hip and should rotations

A scientific study had been made to determine the rotational patterns of a golfer's hip and shoulder using video cameras. The study also calculated the timing of the golf swings with eight sub-10 handicap golfers [5].

The eight swings of every golfer were analyzed and the timing was calculated as shown in Table 2. The result illustrated that the backswing period was longer than the downswing in all subjects. Furthermore, the mean time of the backswing was 0.95±0.12 seconds compared to 0.26±0.05 seconds of the downswing [5].

Table 2. Timing of golf swings from participants [5]

Subject	Backswing (second)	Downswing (second)	Total swing (second)
A	0.76	0.24	1.00
B	0.96	0.34	1.30

C	0.94	0.30	1.24
D	1.04	0.26	1.30
E	0.94	0.20	1.14
F	1.06	0.28	1.34
G	0.82	0.22	1.04
H	1.10	0.24	1.34
Mean	0.95	0.26	1.21
Standard deviation	0.12	0.05	0.14

The study also showed that each golfer's shoulder rotation was faster than the hip rotation and the slope of the shoulder curve was sharper than the hip curve during the backswing as illustrated in Figure 6. In addition, six golfers completed their hip rotation earlier than the shoulder rotation. It meant that six subjects exhibited the same sequential rotation patterns. During their downswing, the shoulder rotations were faster than the hip rotations and both motions in the downswing were quicker than those of the backswing. This analysis suggested that golfers should rotate their hips and shoulders successively to obtain a high club head speed [5].

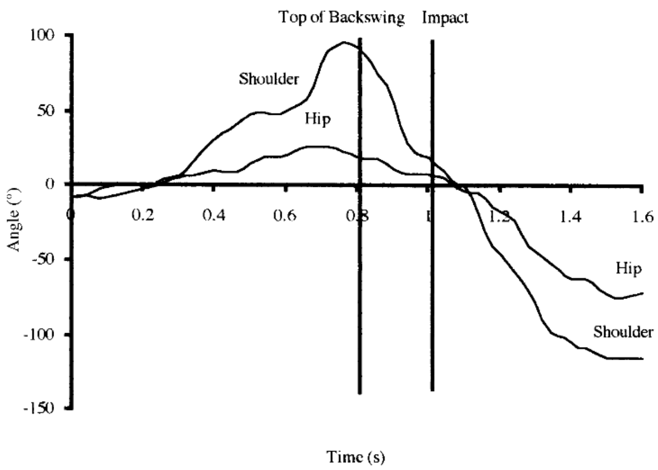


Figure 6. Hip and shoulder angles of a participant [5]

2.2.7 Measurement of kinematic sequence

A scientific research evaluated the motion sequence and timing between amateur and professional golfers using three-dimensional motion analysis system and twelve sensors [17]. The research analyzed the rotational speed and sequence of pelvis, thorax, upper arm and the golf club in the downswing as shown in Figure

7. According to the graph, the proximal segment of the body had a lower rotational speed than the distal segment.

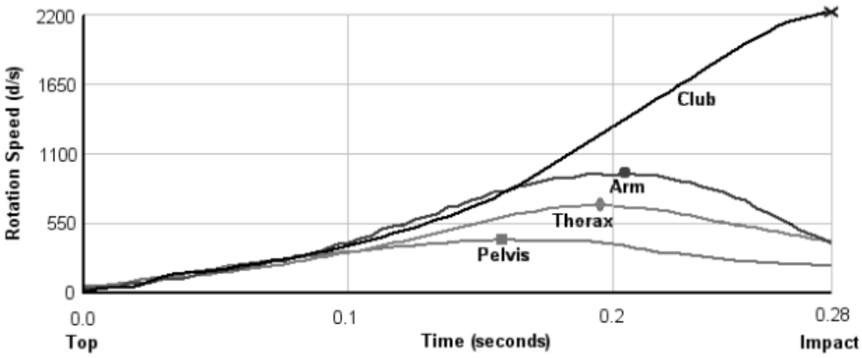


Figure 7. Rotational speed in the downswing of a professional golf [17]

Moreover, the research illustrated a big difference in the sequence of reaching the maximum rotational speed of the body segment during the downswing between the two groups. The body segment order of the professional golfers' was pelvis, thorax, arm and the golf club, whereas the amateurs showed a different sequence such as pelvis, arm, thorax and the golf club as described in Figure 8.

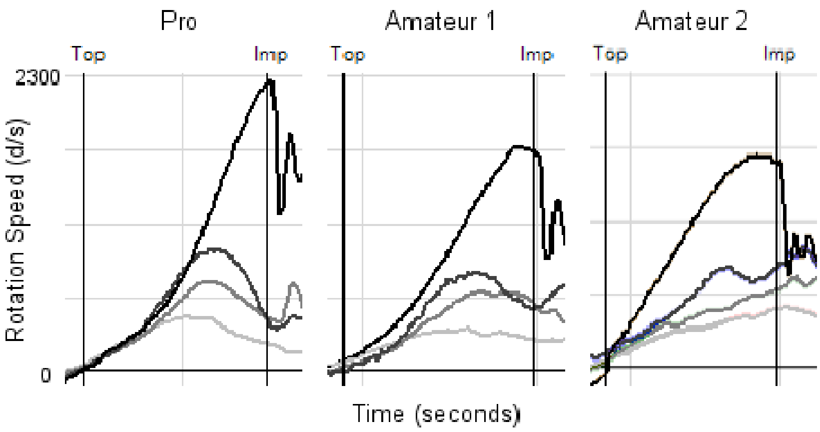
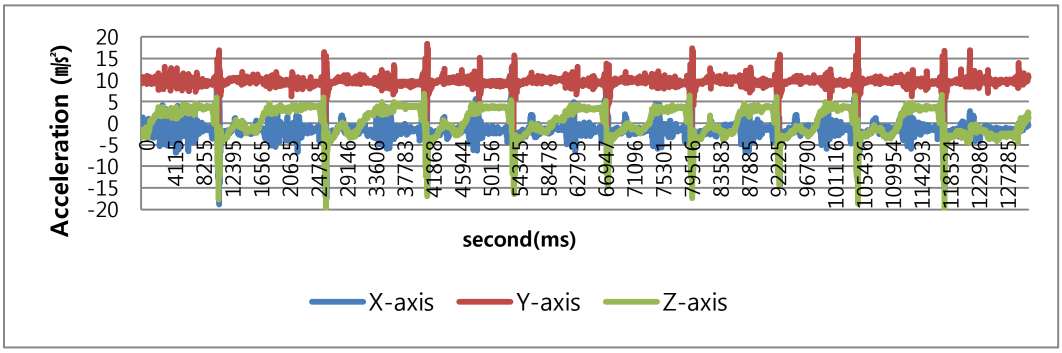


Figure 8. The comparison of the peaking sequence of the body [17]

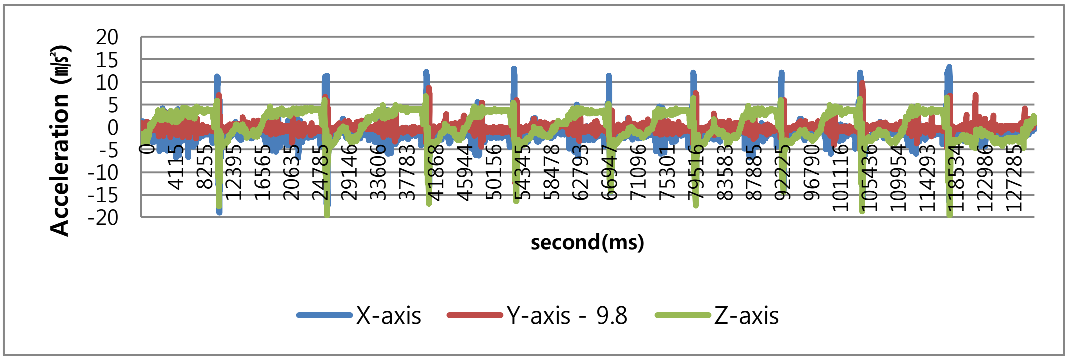
3. Analysis of golf swing

3.1 The graphs of acceleration data

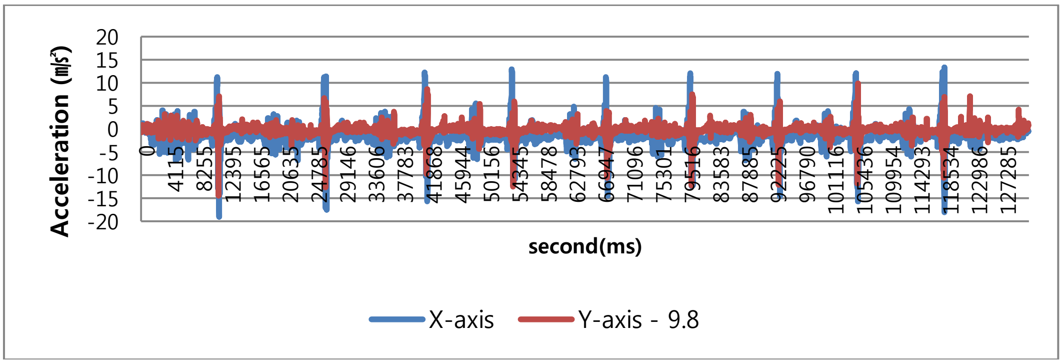
At the first step of this project, a SICS researcher collected the acceleration data of golf swings from eight golfers using the previous work of SICS [18]. Eight golfers swung nine or ten times and these data were saved in each file. Before the design and implementation for this thesis, the collected acceleration data were analyzed first.



(a) The original acceleration graph of golf swings



(b) The acceleration graph subtracted gravity from the y-axis (Y-9.8)



(c) The acceleration graph of the x and y-axis (Y-9.8)

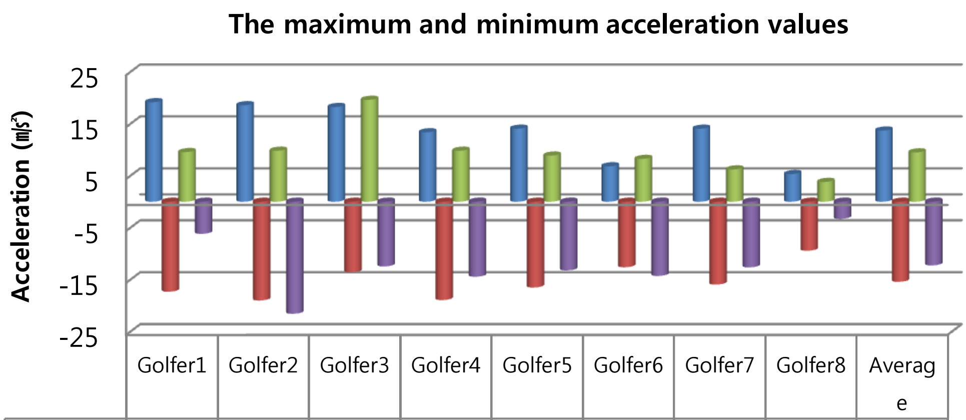
Figure 9. The acceleration graphs of golf swings

As mentioned in 2.1.2, acceleration data consist of three values which are the x, y and z-axis as shown in Figure 9(a). However, when the smart phone is in Up position as described in Table 1, the default acceleration value of the y-axis is 9.81 m/s^2 even though there is no movement. Therefore, the default value should be subtracted from the collected data to find a correlation among each axis' data. Figure 9(b) illustrates the subtraction graph of the y-axis data (Y-9.8) and then the acceleration data of the y-axis are collocated to other axis' data.

Moreover, the acceleration values of the z-axis do not show close correlation when the x-axis and y-axis data have the maximum and minimum values in Figure 9(a). Therefore, the data of the z-axis are not used for detecting critical points in this thesis. Only the x and y-axis data are utilized for this thesis like in Figure 9(c).

3.2 The maximum and minimum values

For the purpose of constituting threshold criteria while identifying golf swings, the maximum and minimum acceleration values were analyzed throughout their whole swing data. The difference of the maximum and minimum acceleration values is significant among golfers as illustrated in Figure 10. These values can be used to decide on some criteria for analyzing golf swing data.



	Golfer1	Golfer2	Golfer3	Golfer4	Golfer5	Golfer6	Golfer7	Golfer8	Average
Max of X	19.16016	18.58535	18.24047	13.37379	14.06356	6.82102	14.06356	5.32652	13.7043
Min of X	-17.3208	-19.0069	-13.5654	-18.9302	-16.5161	-12.6074	-15.9413	-9.42680	-15.4143
Max of Y	9.55176	9.82	19.62000	9.82	8.89031	8.24887	6.21789	3.80371	9.496568
Min of Y	-6.16957	-21.5643	-12.4541	-14.4368	-13.2205	-14.2835	-12.6357	-3.28555	-12.2563

Figure 10. The maximum and minimum acceleration values of the eight golfers

3.3 The analysis of acceleration data

One swing data is extracted from a swing file which contains nine or ten swings. The data consist of four elements such as the timestamp, the acceleration values of the x, y and z-axis but the z-axis data is removed as shown in Figure 11. The data is analyzed and compared with the related research described in 2.2. The acceleration graph has close similarities when it is compared with the two graphs in Figure 2(b) and Figure 5(a) even though the coordinate systems are different from the Android coordinate system. In this chapter, the characteristics of swing durations and the changes of the acceleration values are described.

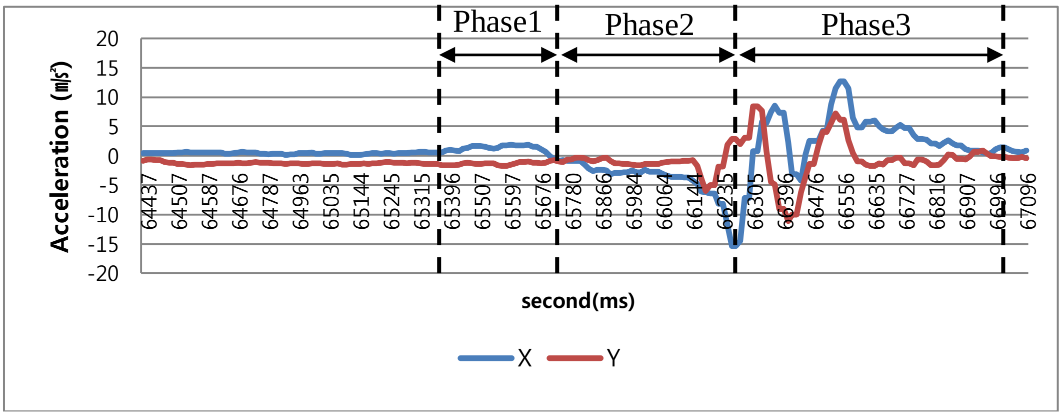
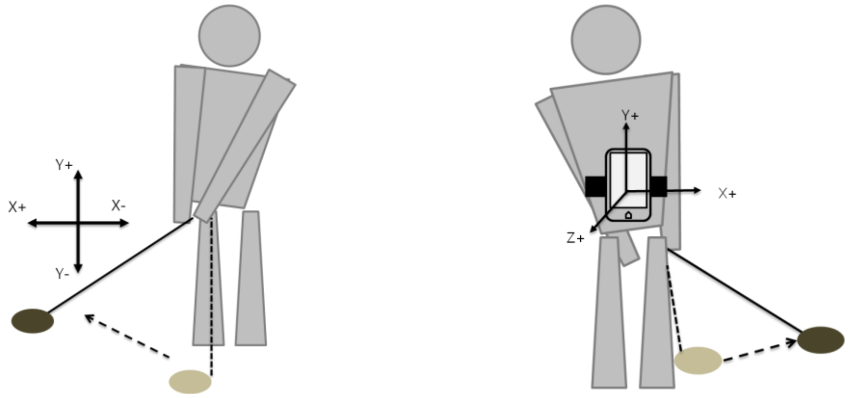


Figure 11. The acceleration graph of a golf swing data

3.3.1 The feature of acceleration during backswing: Phase 1

When the golfer moves the golf club backwards for the backswing after attaching a smart phone on his backside, the acceleration values of the x-axis become positive according to the Android coordinate system as shown in Figure 12.



(a) The front view (b) The backside view

Figure 12. The movement of a golfer during the backswing

The backswing movement is shown in the acceleration graph as phase 1 in Figure 11 and other research also identified this period using acceleration data as described in Figure 4(a) and 5(a). In addition, when the golfer reaches the top swing, he stops his movement for a very short time before starting the downswing. It represents that the acceleration values become zero for a moment. If the acceleration data of the x-axis are simplified by the round-down operation, the backswing duration is distinguishable and can be divided into two areas, which are zero area and non-zero area as shown in Figure 13. It can be a good evidence of identifying the backswing motion even though the acceleration values are small.

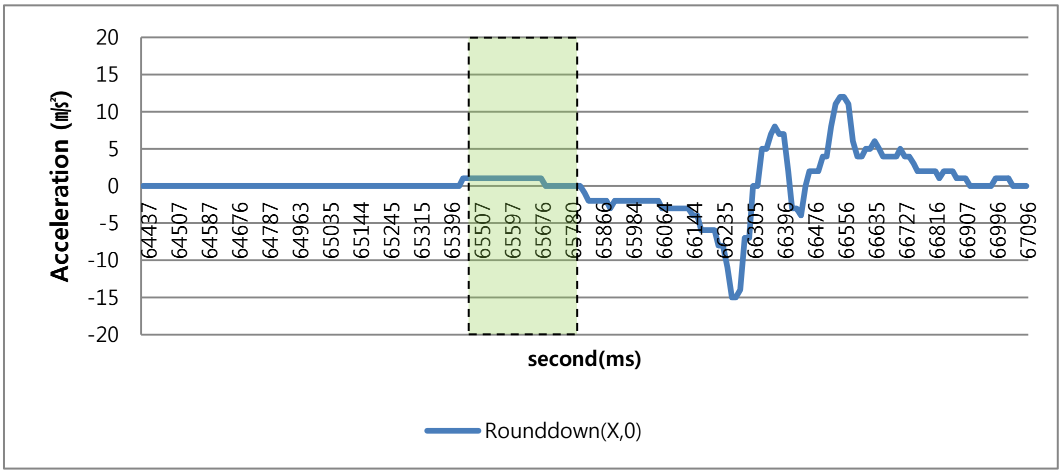


Figure 13. The backswing duration simplified by the round-down operation

Table 3 shows the acceleration values of the x-axis manipulated by the round-down operation and the original collected values. As shown in the table, there are small changes in the values before the negative acceleration value in timestamp 65816.

Table 3. The acceleration values of the backswing duration

Timestamp	X	Round-down(X)	Description
65586	1.762734	1	Non-zero area
65597	1.762734	1	
65606	1.916016	1	
65616	1.762734	1	
65637	1.801055	1	
65646	1.801055	1	
65657	1.839375	1	
65666	1.532813	1	
65676	1.532813	1	

65692	1.111289	1	
65696	0.651445	0	Zero area
65717	-0.07664	0	
65737	-0.53648	0	
65746	-0.80473	0	
65766	-0.72809	0	
65780	-0.88137	0	
65786	-0.88137	0	
65796	-0.84305	0	
65807	-0.84305	0	
65816	-1.49449	-1	The start point of the downswing

3.3.2 The feature of acceleration during downswing: Phase 2

During the downswing, the golfer turns his trunk from the right to the left then the acceleration values increase in the negative direction according to the Android coordinate system. This movement causes the change of acceleration values from positive to negative as shown in Figure 13. Moreover, when the golfer hits a ball, the acceleration value of the x-axis becomes the biggest in the negative direction as described in Figure 2(b) and 4(b).

However, the acceleration values of the y-axis increases when the acceleration data of the x-axis reaches the negative peak point. It means the golfer’s trunk starts to move upwards during the downswing.

3.3.3 The feature of acceleration during follow-through: Phase 3

After the impact, the follow-through motion follows and the acceleration graph has fluctuated a few times. However, two positive maximum peak points of the x and y-axis are shown in every golfer. As mentioned in chapter 2, it means the acceleration is decreasing to the negative direction and the golfer moves towards the positive direction of the coordinate system.

3.3.4 The feature of acceleration in finish

When the player finishes his golf swing, his movement is stopped and the acceleration values can be almost zero. To verify this hypothesis, all swing data from the eight golfers were analyzed after manipulating the acceleration data by round-down operation to make them simple. Figure 14 illustrates the simplified acceleration graph and shows the areas selected for examination. The areas were chosen because these are considered as the end of each swing.

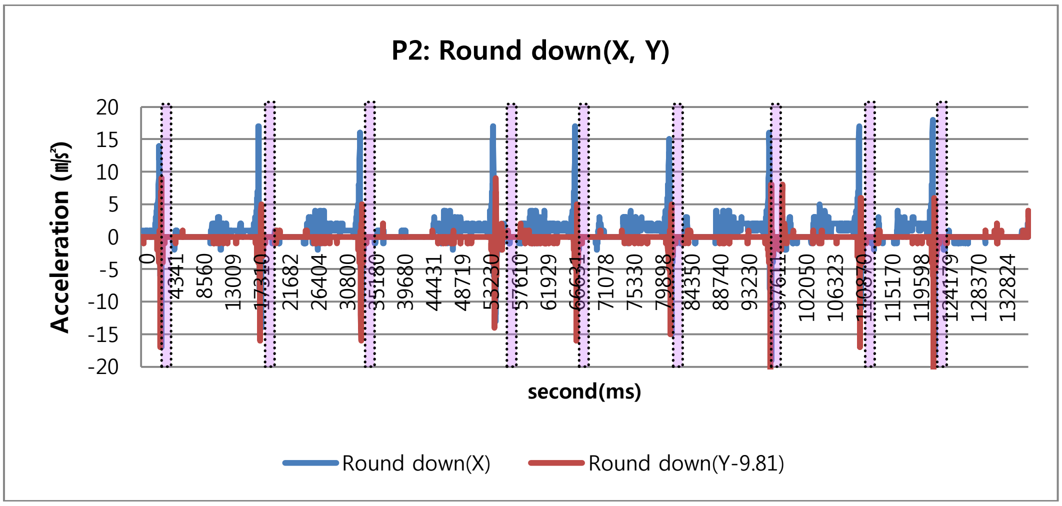


Figure 14. The common pattern of the finish motion

Every selected area has the common patterns which have zero consecutively more than five times in the x and y axes. Also the same patterns exist in all swing data collected from the eight golfers. Table 4 shows three examples of the patterns.

Table 4. The common patterns of finish

No.	Timestamp	X	Y	Timestamp	X	Y	Timestamp	X	Y
1	3638	0	0	18291	0	0	34299	0	0
2	3659	0	0	18411	0	0	34319	0	0
3	3669	0	0	18420	0	0	34329	0	0
4	3679	0	0	18431	0	0	34349	0	0
5	3689	0	0	18440	0	0	34359	0	0
6	3699	0	0	18451	0	0	34368	0	0

3.4 The definition of critical points of a golf swing

In this thesis, three phases are defined for the analysis and feedback of a golf swing. The phase 1 is the first movement of the swing. It contains the swing period from the backswing to the start of the downswing. The phase 2 is the second period of the swing from the downswing to the impact point. During this phase, the acceleration values of x-axis are negative. The last period is the phase 3, where most acceleration data of the two axes are positive. This phase includes the last movement of the swing from the impact point to finish.

In addition, seven critical points are also specified to analyze a golf swing data. Those are the backswing point, downswing point, two maximum peak points of the x and y axes, two minimum peak points of the x and y axes, and the end point respectively. The definition is based on the acceleration values from the backswing to finish motion. These points are illustrated in Figure 15.

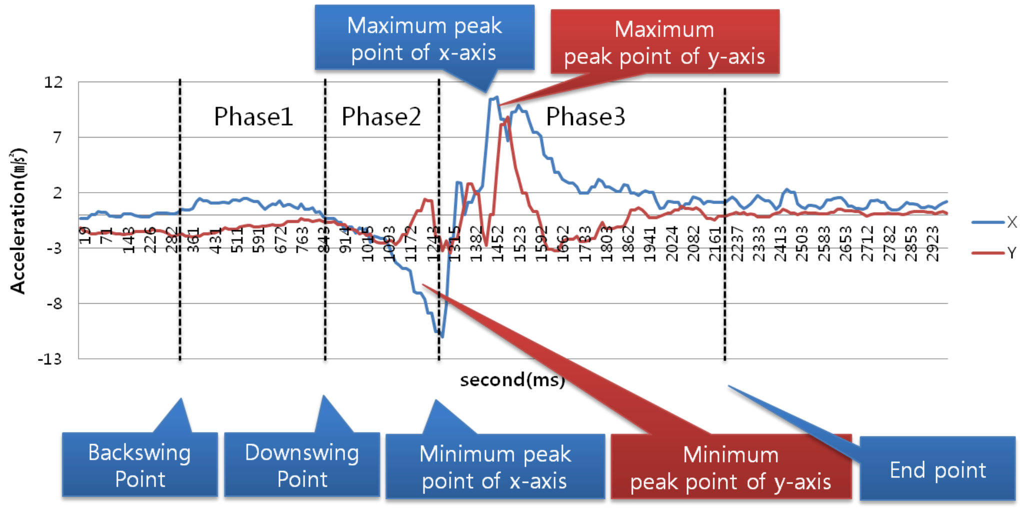


Figure 15. The definition of critical points in a golf swing

- Backswing point

  This point is the start point of a golf swing. The golfer moves his club backwards and then there are small movements and some changes in the acceleration data. Phase 1 contains two areas which are the backswing and the top swing. In this paper, phase1 is considered as the backswing period. When the feedback is provided to the user, it starts from this point.

- Downswing point

  This point is where the acceleration values become negative from positive. It is considered as the start of the down swing. The duration of phase 2 is computed from this point to the minimum peak point of the x-axis.

- Minimum peak point of the x-axis

  The point represents the change of direction of a swing and also it is considered as the impact point because the acceleration value is the biggest to the negative direction. After this point, the acceleration values turn to increase until the maximum peak point.

- Maximum peak point of the x-axis

  The maximum peak point of the x-axis is related to the movement from the impact point to the follow-through motion of the golfer. However, the point does not mean the maximum value of a whole swing but a positive peak after the minimum peak point.

- Minimum peak point of the y-axis

  This point has the smallest acceleration value of the y-axis during a swing motion and follows the minimum peak point of the x-axis.

- Maximum peak point of the y-axis

  This is the point where the acceleration data is the biggest in the y-axis. It is placed after the maximum peak point of the x-axis.

- End point

  It is defined as the end of the swing motion and is the point where acceleration values are around zero in both axes. When a swing is analyzed, the process is finished at this point. Also, this point is used for calculating the duration of phase 3.

4. Design

4.1 System overview

A golfer attaches an Android smart phone to his body and swings a golf club to collect his swing data. Whenever he swings a club, the acceleration data from an embedded accelerometer are collected and saved. After a swing is done, the analyzed data will be given to the golfer as a feedback as shown in Figure 16. This system can provide different types of feedback such as the acceleration data of the swing, the maximum and minimum values and the duration of the swing. The feedback is helpful for the golfer to improve his skills and identify some patterns during the swings.

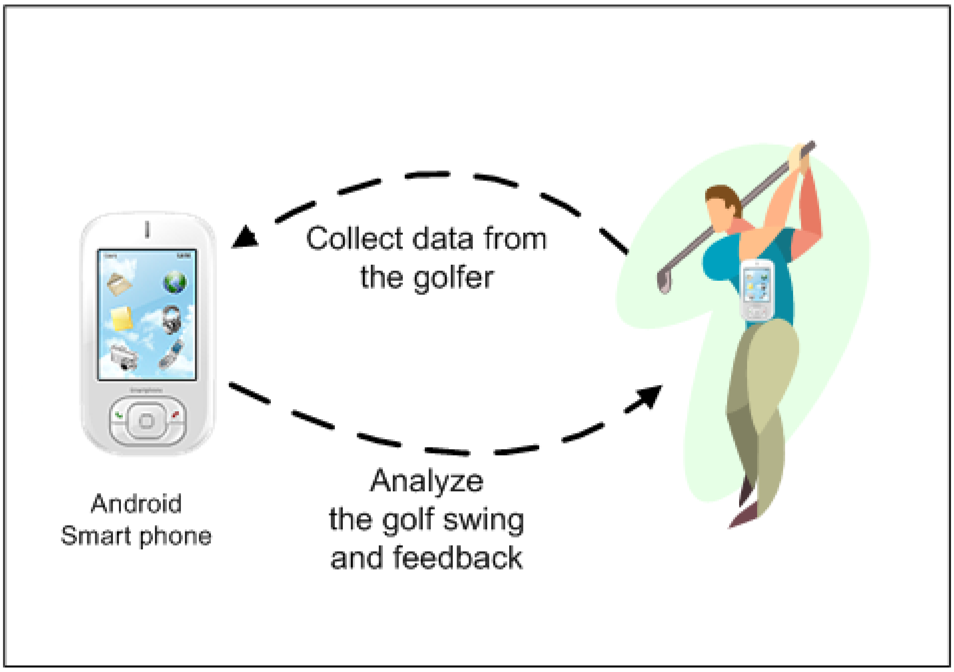


Figure 16. System overview

4.2 Software architecture

The software runs on an Android smart phone and collects acceleration data from the accelerometer of a smart phone. It analyzes the acceleration data to identify some critical points such as a start, an end point and two peak points of the x-axis and y-axis from the collected acceleration data. Moreover, it provides audio-visual feedback for a certain period related to the analyzed swing data and makes different sounds in accordance with the strength of the acceleration data. With this feedback, the golfer can recognize the patterns and strength of his swing. The architecture consisting of a few functional modules is described in Figure17.

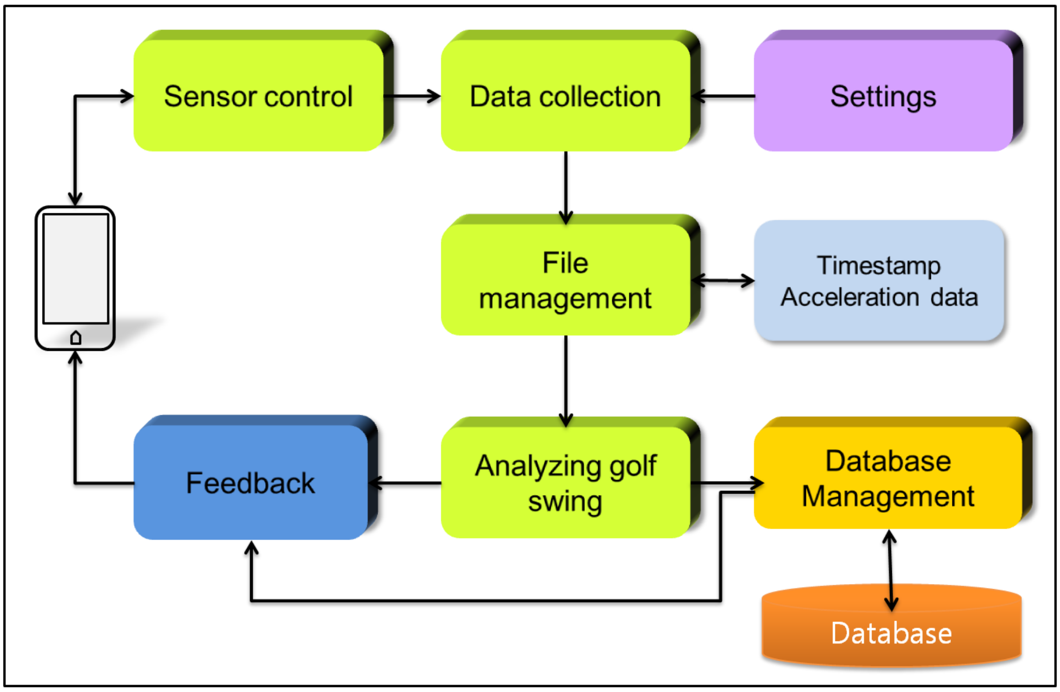


Figure 17. Software architecture

- Sensor control

  It initializes and enables an accelerometer before data collection. When the application does not need the sensor, it unregisters the sensor. This module also sets the sampling rate of the acceleration data.

- Data collection

  This module collects acceleration data while the golfer swings a club during the pre-defined collection time. The collection time and the number of swings are set in the settings module.

- Analyzing golf swing

  It analyzes the acceleration data and identifies the critical points defined in 3.4. In addition, it calculates the swing durations using the identified points. After the analysis, the data is saved in the database table.

- Feedback

  This module gives the player different kinds of feedback according to the analyzed swing data. It generates different beep sounds in accordance with the swing duration and the strength of acceleration data which are analyzed. Moreover, it provides a visual feedback using the collected acceleration values.

- File management

  It saves the collected acceleration data to a file for the purpose of using them in other modules and deletes files according to the data keeping time which is defined by the user.

- Database management

  This manages database system and displays important data when the user requests for swing information. In addition, it manages the golfer's previous swing data such as the maximum and minimum acceleration values and swing durations.

- Settings

  It provides a few functionalities about defining the collection time, the number of swing, the feedback types and threshold values. These defined values are saved in the internal memory and are used by other functional modules.

4.3 The design of feedback

In sports activities, audio feedback is adopted to inform players about their inappropriate movements or provide instructions. This feedback is useful for them to be aware of their actions [19]. In this thesis, we design audio feedback and other feedback functionalities to provide golfers with information related to golf swing.

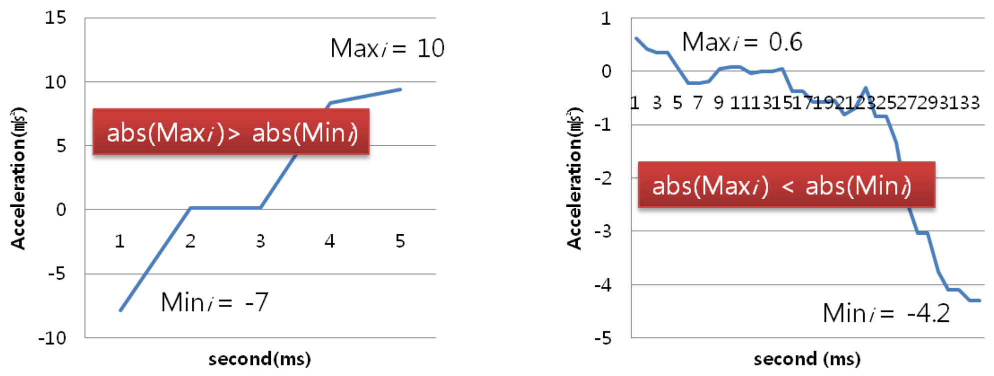
The software provides audio and visual feedback together according to the acceleration values and the identified critical points. These feedbacks can be chosen by the user with his preference and each type of feedback is given to the user for a specific time period related to the swing duration. Before providing a feedback, the feedback module calculates the swing duration and divides the period by 10 timeslots. Each timeslot is characterized by a minimum peak point, a maximum peak point or an absolute maximum value. Using this characteristic, each time slots displays an icon and a sound according to a defined value. This feedback makes the golfer figure out the strength and patterns of his swing.

4.3.1 Displaying acceleration values

Each timeslot contains many different acceleration values so a representative acceleration per timeslot is required to show the value due to the limitation of small size display. The biggest acceleration value regardless of the sign represents the maximum force and the fastest movement in a timeslot and this is one of the main interests of most golfers as mentioned in section 1.1. Therefore, the biggest value is chosen as a representative acceleration for the timeslot.

The feedback module finds the maximum and minimum acceleration values for each timeslot and chooses the biggest value between the maximum (Max_i) and minimum value (Min_i) regardless of the sign like equation (3). This value is displayed as a representative for each timeslot and is used for displaying icons and making feedback sounds. Figure 18 shows how to calculate the representative value for a timeslot.

$$X_i = \max(|Max_i|, |Min_i|) \quad (3)$$



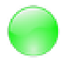
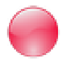
(a) abs(Max_i) > abs(Min_i) (b) abs(Max_i) < abs(Min_i)

Figure 18. Maximum and minimum values in a timeslot

4.3.2 Beep sound feedback with icons

The beep sound feedback uses four kinds of sounds and icons as illustrated in Table 5. The beep sounds consist of a normal pitch sound, a high pitch sound for the maximum value, a low pitch sound for the minimum value and a combined pitch sound when the minimum peak and the maximum peak exist in the same timeslot respectively. For example, if the timeslot contains a maximum peak value, it makes a high pitch sound and displays a red icon as a feedback. In addition, other timeslots which do not contain the maximum and minimum peak point generate normal pitch sounds and show green icons. This feedback can help the golfer to detect the two peak points well.

Table 5. Beep sound feedback

Type	Beep sound	Icon
Normal point	Normal pitch	
Maximum peak point	High pitch	
Minimum peak point	Low pitch	
Two peak points in a time slot	Combined pitch	

4.4 The design of database

The software manages a database table using SQLite [20] database system which is provided by Android SDK (Software Development Kit) [21]. Whenever the golfer swings his golf club, the software analyzes his swing data and saves the analyzed data into the database. The database provides the golfer's swing history such as the time durations of each swing period and the maximum and minimum acceleration values. The database is designed to provide the golfer with swing history and his progress. In addition, the golfer can be aware of his swing patterns and the changes of maximum and minimum acceleration values. The database table is described in Table 6.

Table 6. Database table

Index	Name	Type	Description
0	ID	Integer	Primary key
1	DATE	Text	The date of collection
2	TIME	Text	The time of collection
3	X_MAX	Text	The maximum acceleration value of x-axis
4	X_MIN	Text	The minimum acceleration value of x-axis
5	Y_MAX	Text	The maximum acceleration value of y-axis

6	Y_MIN	Text	The minimum acceleration value of y-axis
7	SWING_DURATION	Text	Total swing duration
8	SWING_PHASE1	Text	Backswing time
9	SWING_PHASE2	Text	Downswing time
10	SWING_PHASE3	Text	Follow-through time

5. Implementation

5.1 The implementation environment

This system was developed for Android SDK 2.3.3 or a later version and AChartEngine library version 1.0 [22] was used to provide visual feedbacks to the golfer. These environments are described in Table 7.

Table 7.The implementation environment

	Environment
Android SDK version	SDK 2.3.3
External library	AChartEngine version 1.0
LCD size	3 inch (Sony Ericsson Xperia Active [23])

5.2 Implementation details

The software is composed of java classes which support different functionalities as illustrated in Figure 19. When the golfer swings a golf club, the acceleration data are collected as the AccelerationData format in the DataCollection class. The data are delivered to the SwingDetectionThread for analyzing the swing data. The thread tries to seek the critical points described in 3.4. If the detection process is successful, the SwingDetectionThread sends messages to the SwingFeedback class to inform the timestamp of the detected points and the swing durations. Then the SwingFeedback class assigns icons and sounds to 10 timeslots according to the calculated acceleration values, the maximum and minimum peak points. After this procedure, the SwingFeedback class delivers the analyzed information to the SwingFeedbackView class to provide feedback to the golfer. This class takes responsibility of displaying icons and making sounds.

In addition, the software provides the functionality of analyzing the previous golf swing data in the PastSwingDataFeedback class. This class helps a golfer to review his previous swing patterns. For the feedback of the previous swing data, the SwingFeedbackView class is reused for providing the feedback to the golfer. The ExternalSwingAnalysis class is implemented to analyze external swing data which are collected from other devices using this software or other applications only if the data format is the same to this work.

The Statistics class is implemented to present two types of graph about the golfer’s swing history and patterns. This class uses the DatabaseHandler class to read the history information and uses the SwingGraph class to draw graphs.

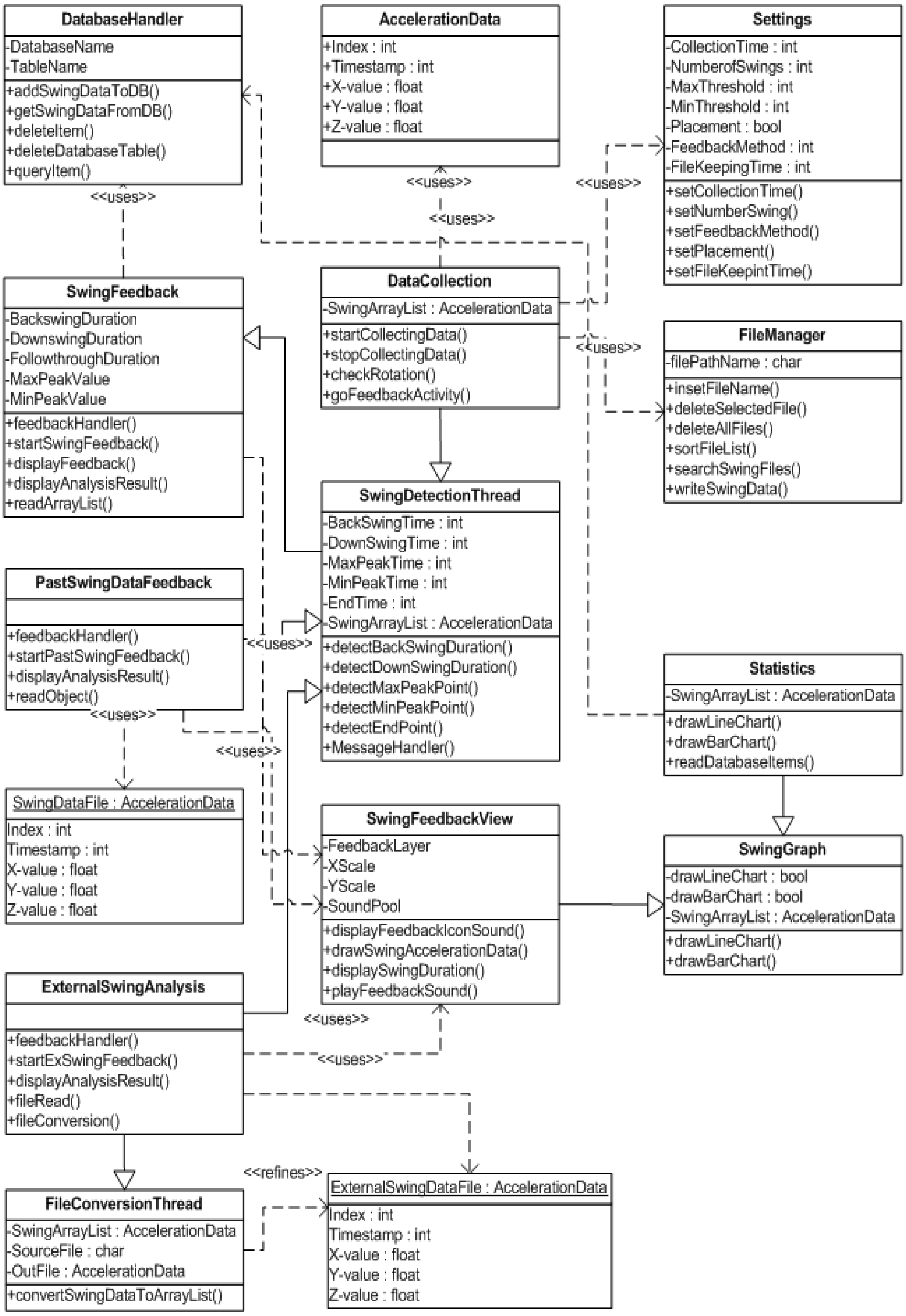


Figure 19. Class diagram

5.3 Data collection

The acceleration data are collected from an accelerometer in an Android smart phone whenever the movement of the golfer happens. The Android API supports four different sampling rates for the accelerometer according to the purpose. The fastest sampling rate has no delay (SENSOR_DELAY_FASTEST), the second rate has 20,000 microseconds delay (SENSOR_DELAY_GAME), and the third one has 60,000 microsecond delay for sampling (SENSOR_DELAY_UI). The default sampling rate (SENSOR_DELAY_NORMAL) collects acceleration data every 200,000 microseconds [12]. In this thesis, the fastest option is selected for data collection and the application can receive data from the sensor approximately every 10 msec.

Moreover, this software provides the function of automatic data collection for the user's convenience. If a golfer chooses a collection time and the number of swing in settings menu, the application can collect acceleration data without the user's intervention. For example, if the golfer sets 20 seconds for swing collection time and 5 times for the number of swings in the settings menu, then he can collect data five times for every 20 seconds without the user's intervention as shown in Figure 20.

In addition, the collected data is written into two kinds of file for drawing a graph and further feedback. One is just a plain text format which can be used in other software and the other is a converted format.

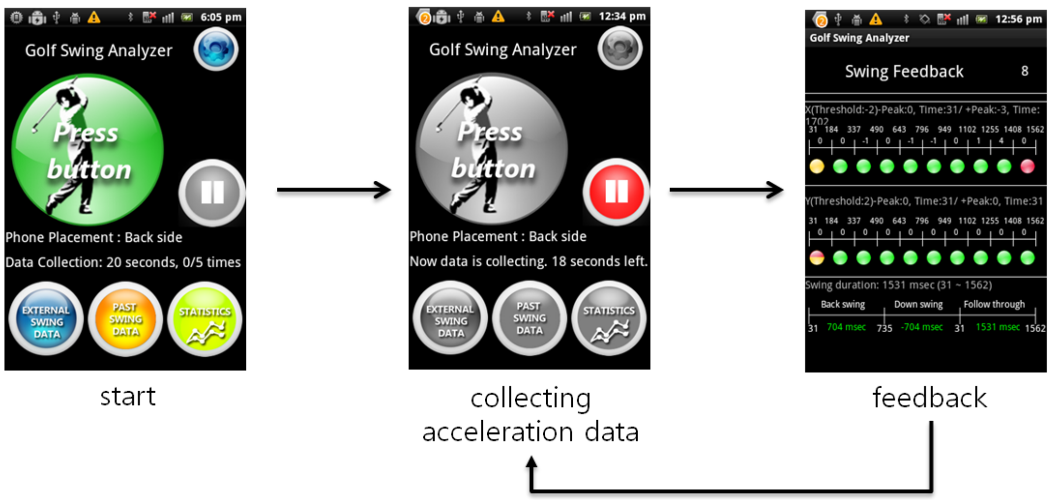


Figure 20.The scenario of collecting acceleration data

5.3.1 Data format

The software collects acceleration data, which are x, y, and z-axis data, from an accelerometer in an Android smart phone whenever the player makes a movement. However, these data are not enough for analyzing golf swings so the index and timestamp fields are added for a more precise analysis. The data format is shown in Figure 21. In addition, whenever the software receives acceleration data, the software saves the data to an array linked list before writing the data to a file. After collecting acceleration data during a specific period, the software uses the array linked list for analyzing a golf swing. The z-axis data are also saved even though those are not used for analysis.

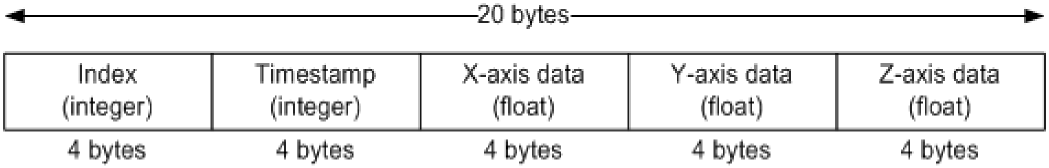


Figure 21. Data collection format

5.4 Message flows for detecting critical points

After collecting data for a given time period, the software starts to analyze the golf swing data as shown in Figure 22. The DataCollection class runs the SwingDetectionThread class as a thread with the collected data and it waits for messages from the thread which detects the critical points. Whenever the thread detects each point, it sends messages to the DataCollection class with the index and timestamp of the points as parameters. First, the thread tries to detect the points of the x-axis because the x-axis's data has the priority. After sending messages related to the x-axis, the thread tries to identify the minimum peak point and maximum peak point of the y-axis. For example, if the thread finds critical points of the x-axis, it sends messages such as MSG_DOWNSWING_POINT, MSG_BACKSWING_POINT, MSG_PEAK_X_MIN, MSG_PEAK_X_MAX, and MSG_END_POINT. After receiving a MSG_DETECTION_DONE message from the thread, the DataCollection class requests the SwingFeedback class to display feedback with the calculated time and values such as peak values, timestamps, the backswing time, the downswing time and the end time of a swing.

However, if the thread cannot detect any critical points, it also sends a MSG_DETECTION_DONE message with -1 as a parameter and the SwingFeedback class provides an error feedback.

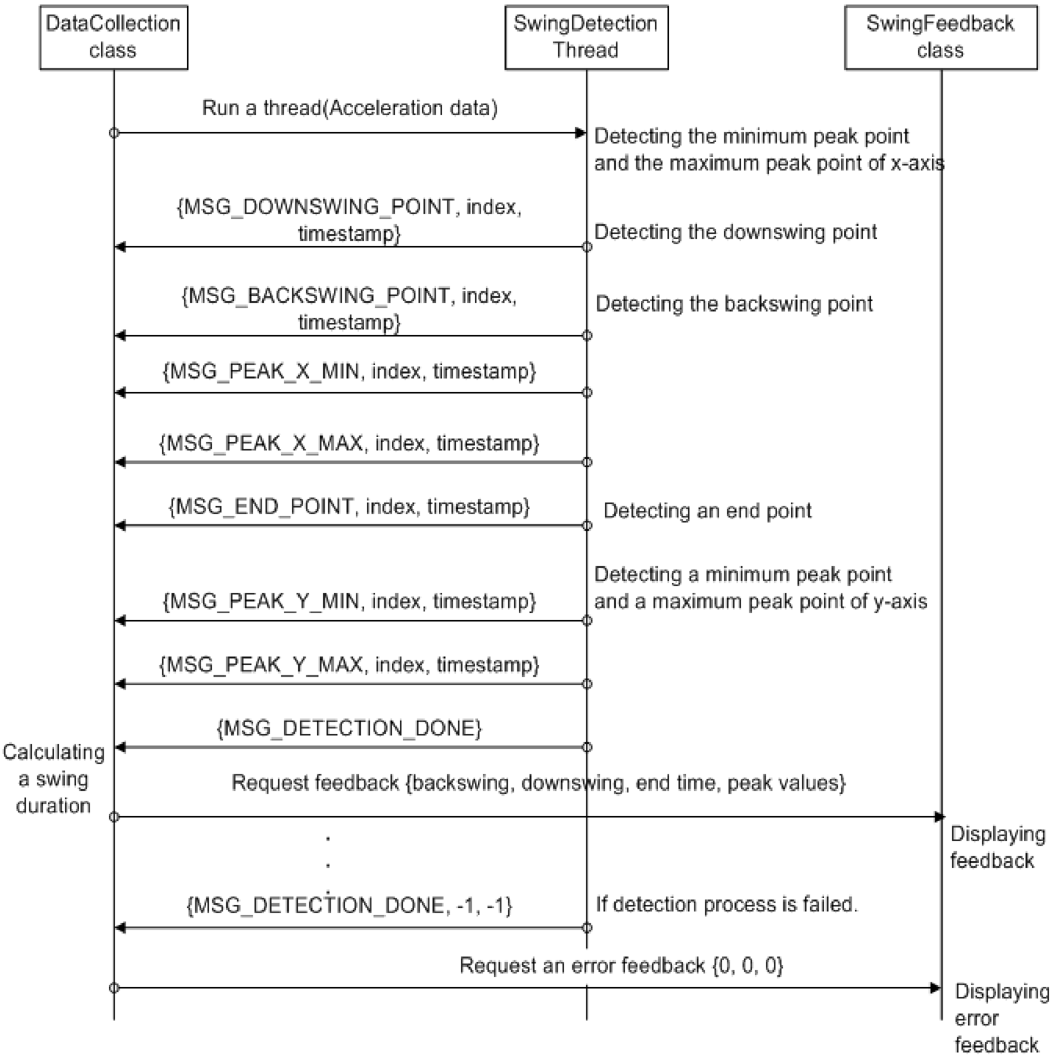


Figure 22. Message flows for the detection process

5.5 The detection process of critical points

5.5.1 The sequence of detection process

The detection process of critical points is illustrated in Figure 23. First, the software tries to find the minimum peak point of the x-axis because the point is used in the beginning of this process. If the minimum peak point is found, the software processes the further steps. However, if the point is not detected because the acceleration values are less than the threshold, the detection process stops. Second, after successful detection of the minimum peak point, the software analyzes the acceleration data forwards from the minimum peak point to find the maximum peak point of the x-axis. Third, the software traverses backwards from the minimum peak point of the x-axis to identify the downswing point. Fourth, the identifying the backswing is carried out from the downswing point because of exclusion of the golfer's preliminary motion in the earliest part of his swing. The golfer may have some motions to exercise or prepare his swing so this movement should not be included. Fifth, the detecting end point is performed from the minimum peak as well. The software moves forwards to find the end point. If the pattern of the end point is found, the software returns the index of the data and stops this process. Finally, the detection of the minimum and maximum peak points of the y-axis is processed.

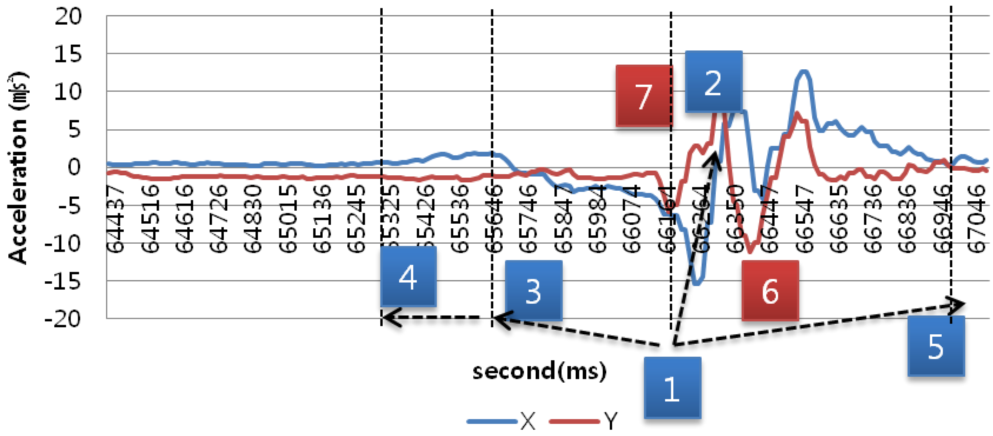


Figure 23. The sequence of detection process

5.5.2 Detecting the minimum peak point

The minimum peak point of the x-axis is considered as the most important point because the movement of the x-axis coordinate is much bigger and clearer than the data of the y-axis during a golf swing. Therefore, detecting this point is done prior to others and if it is not detected for various reasons, the software reports a detection error. In this thesis, two threshold values for the positive and negative direction are used for identifying each peak point. The default threshold values are set as 5 ㎨ and -5 ㎨ according to the eight golfers' maximum and minimum acceleration values described in Figure 10. Most golfers' acceleration values exceeded the default threshold values. However, the user can change these values in the setting menu according to his skill. In addition, if the acceleration values collected from the golfer are less than the threshold, the software does not proceed to find other points. It means that the acceleration values are too small so it is considered as a meaningless swing data. The algorithm for identifying this point is shown in Figure 24.

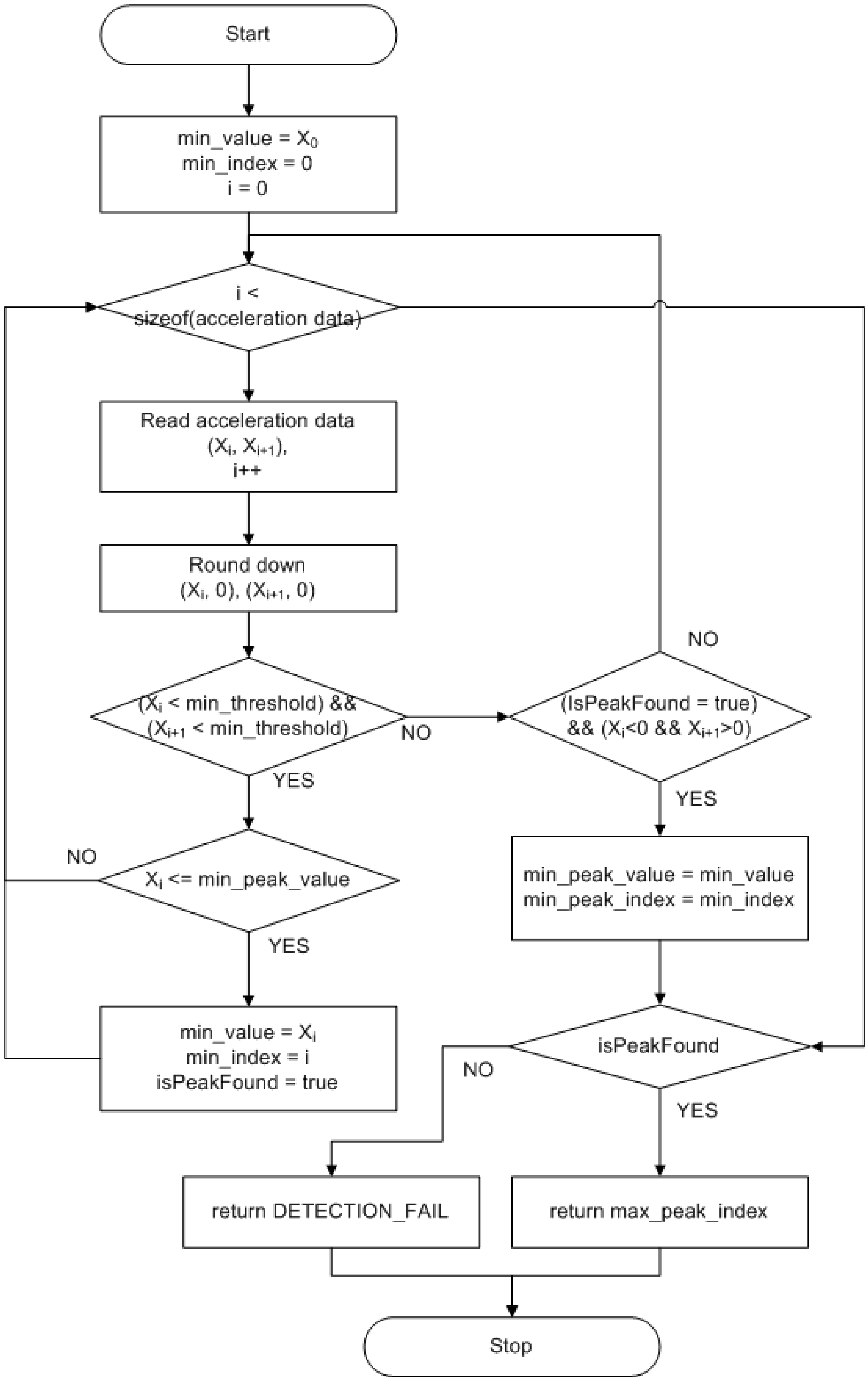


Figure 24. The flowchart of detecting a minimum peak point

The collected acceleration data are float-type numbers and contain many similar values and noises so the manipulating acceleration data is essential to find this point. Therefore, the round-down operation (see Appendix F) is used to make acceleration data simple.

This algorithm also finds the transition point where the data changes from a negative to a positive value for detecting the exact point. Finding the transition point is very important to identify the exact peak point because a few jagged patterns exist inside of the data as illustrated in Figure 25. This algorithm analyzes the data until finding the transition point and compares the data to the minimum value (see Appendix A). Without this procedure, the first pattern in Figure 25 can be considered as the peak point so it causes a wrong detection. When the transition point is found, the detected minimum value is chosen as the minimum peak point.

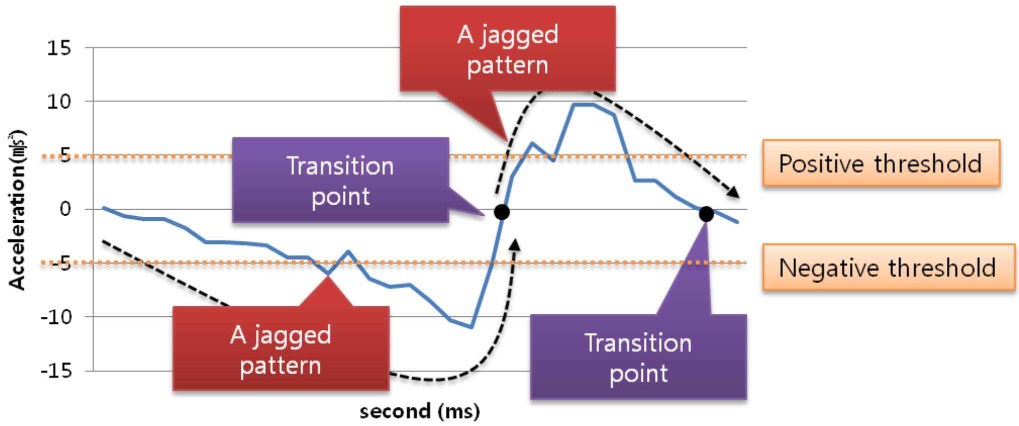


Figure 25. Transition points and thresholds

5.5.3 Detecting the maximum peak point

This procedure starts to analyze the acceleration values from the minimum peak point to a transition point where acceleration value changes from a positive to a negative value like the second transition point in Figure 25. This step also uses a threshold to skip small acceleration values considered as weak movements and manipulates the collected data using a round-down operation to eliminate noises and identify this point easily (see Appendix B). Figure 26 illustrates the algorithm of detecting the maximum peak point of the x-axis.

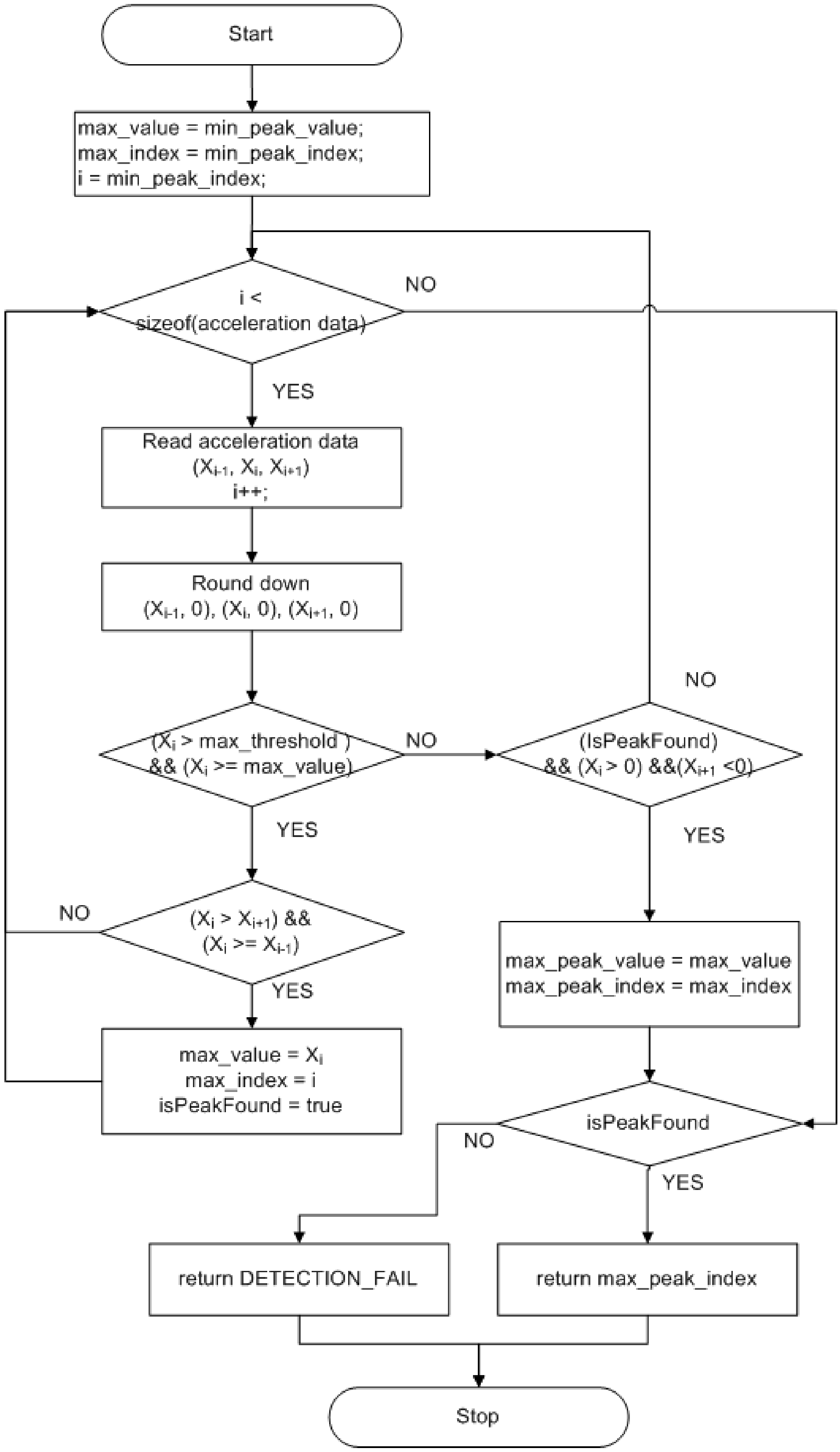


Figure 26. The flowchart of detecting the maximum peak point

5.5.4 Detecting the downswing point

The downswing point defines the first zero point from the minimum peak point of the x-axis in the reserve direction. In this thesis, only the acceleration data of the x-axis is used for detecting the downswing point because the movement along the x-axis is more distinct than that of the y-axis. However, the collected acceleration data do not have the exact zero values because the acceleration data are collected in a floating-point number format shown in Table 8. Therefore, data manipulation process is required for an effective detection of the point.

Table 8. Data manipulation for the downswing point

The original collected data	The manipulated data by round-down operation
-0.84305	0 (start point)
-1.34121	-1
-2.4525	-2
-3.02731	-3
-3.75539	-3
-4.10027	-4
-5.97797	-4
-7.12758	-5
-7.1659	-7
-8.04727	-7
-9.15856	-8
-9.54176	-9 (the minimum peak point)

In some cases, detecting the point is difficult because of some noises of the acceleration data or the player's abnormal movement. To solve this problem, if this point is not detectable, then the first index of the collected data is used as the starting point of the downswing for feedback (see Appendix C). The algorithm is described in Figure 27.

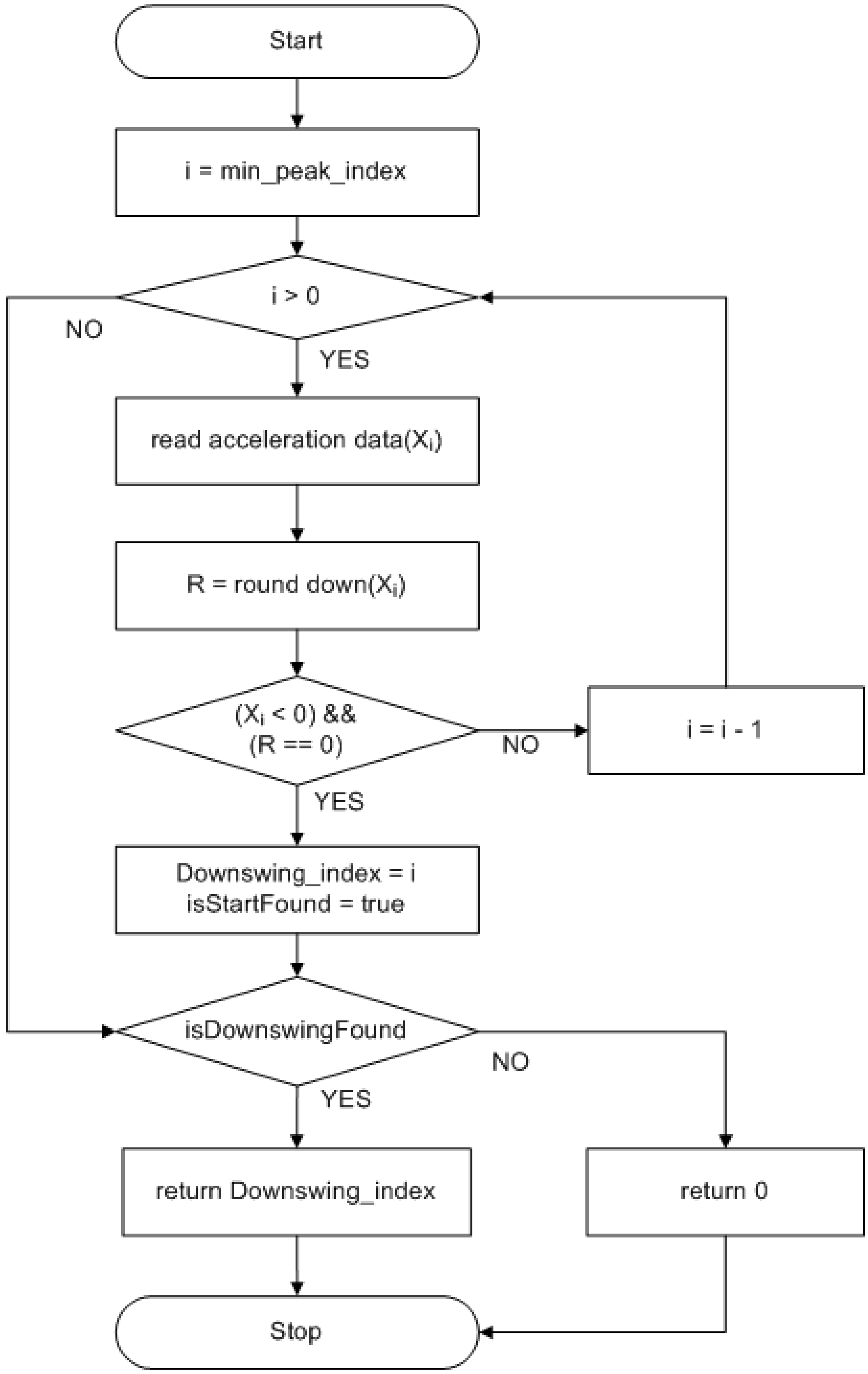


Figure 27. The flowchart of detecting the downswing point

5.5.5 Detecting the backswing

There are two areas in front of the downswing duration even though the acceleration values are very small. The areas consist of a non-zero area and a zero area as shown in Figure 28. When the golfer moves his club backwards for the backswing, acceleration data change along the x-axis and there is a motionless area because the player stops his swing at the top. The two areas are very good clues to identify the backswing duration.

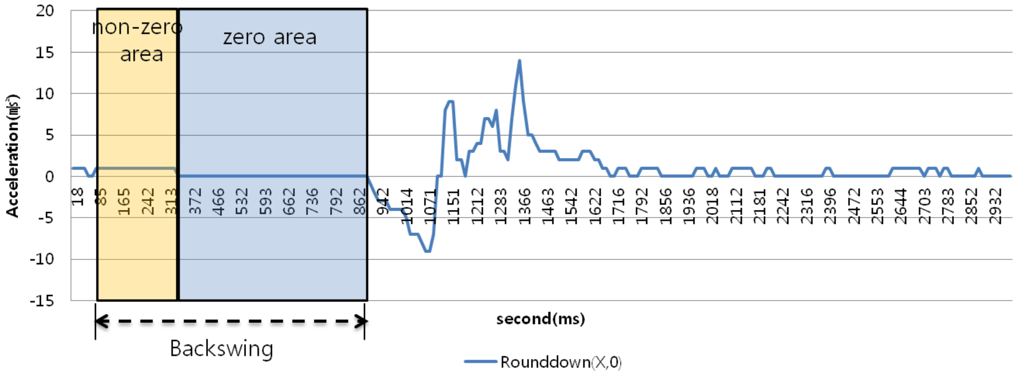


Figure 28. Backswing areas

To detect this point, the software reversely searches zero areas considered as the top swing area from the downswing point. If zero values exist more than four times (TOPSWING_CRITERION) continuously, the software considers the area as the zero area. After finding the zero area, this process proceeds to find non-zero areas before the zero area (top swing). If there are consecutive non-zero values more than four times (BACKSWING_CRITERION) in front of the zero area, the software continues to traverse until finding a zero value. If a zero value is found before the non-zero area, the software defines the area as the non-zero area and calculates the backswing duration from the start of the non-zero area to the end of zero area. If the software cannot discover these two consecutive areas, it returns zero for the backswing duration (see Appendix D).

In this thesis, two criteria values are defined for identifying the zero area and the non-zero area, which are TOPSWING_CRITERION and BACKSWING_CRITERION. The two criteria are defined as the number four according to the analysis of the eight golfers' swing. Figure 29 illustrates the algorithm of detecting the backswing.

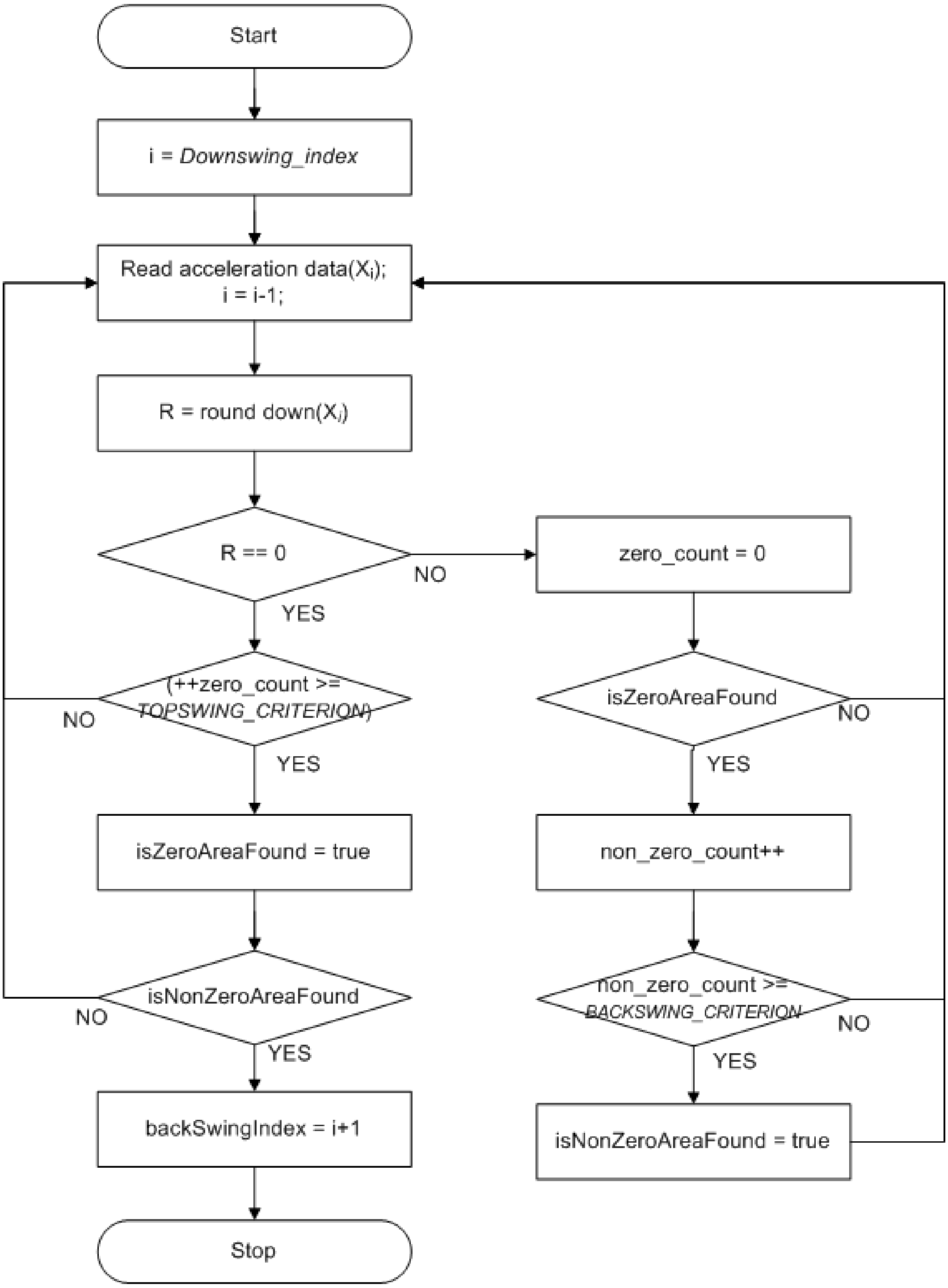


Figure 29. The flowchart of detecting the backswing

5.5.6 Detecting the end point

The end point is the motionless point after a golf swing. When the golfer finishes his swing, the movement must be very small and he seems to be inactive for a while. In that case, the acceleration values are around zero in the two axes. This idea is used to detect the end point of a swing and this process also includes a data manipulation like the other detection processes and uses both the x and y-axis data for a more precise identification.

This process starts from the minimum peak point and reads two values of the x-axis and y-axis from the saved array list. To simplify these data, the round-down operation is applied to detect the end point effectively as shown in Table 9. After this operation, some points which acceleration values are zero consecutively in the two axes are found in each swing.

Table 9. Data manipulation for an end point

The collected acceleration data		The manipulated acceleration data	
X-axis	Y-axis	X-axis	Y-axis
0.843047	0.498164	0	0
0.843047	0.498164	0	0
0.728086	0.306562	0	0
0.766406	0.153281	0	0
0.689766	0.076641	0	0
0.498164	0.153281	0	0

When these zero values are found after the minimum peak point of the x-axis, the number is counted. If the counted number is the same as the criterion (END_CRITERION = 5), the first index of the acceleration data is defined as the end point. However, if the consecutive zero values are not found because of some reasons such as abnormal golf swing or data collection timeout, then the last index of the acceleration data is used as the end point (see Appendix E). The algorithm of detecting the end point is shown in Figure 30. In this thesis, only the acceleration values of the x-axis are used for identifying the end point.

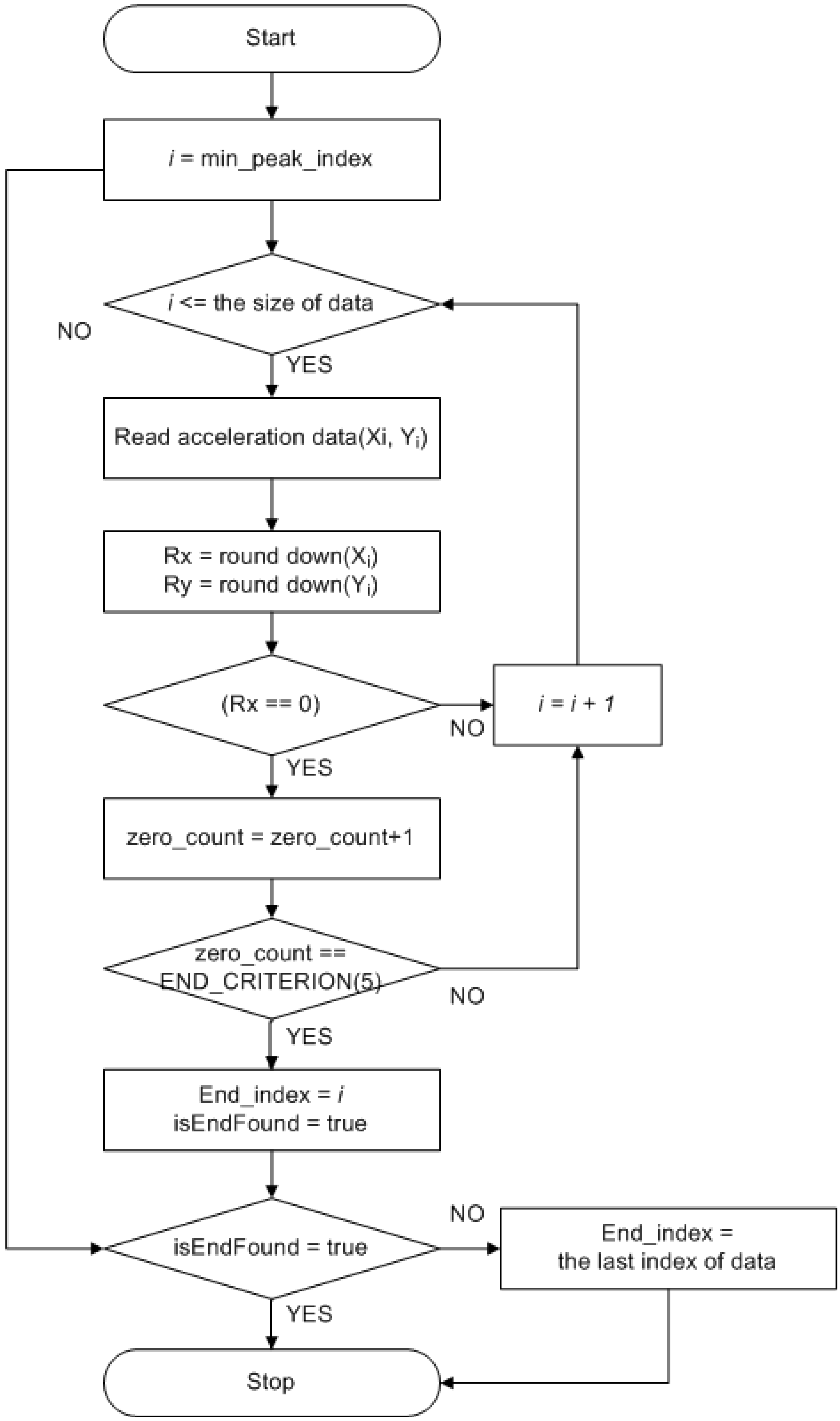


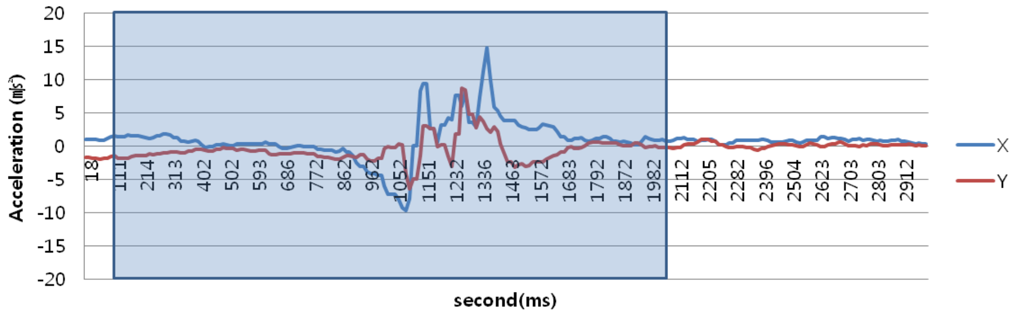
Figure 30. The flowchart of detecting an end point

5.6 Feedback

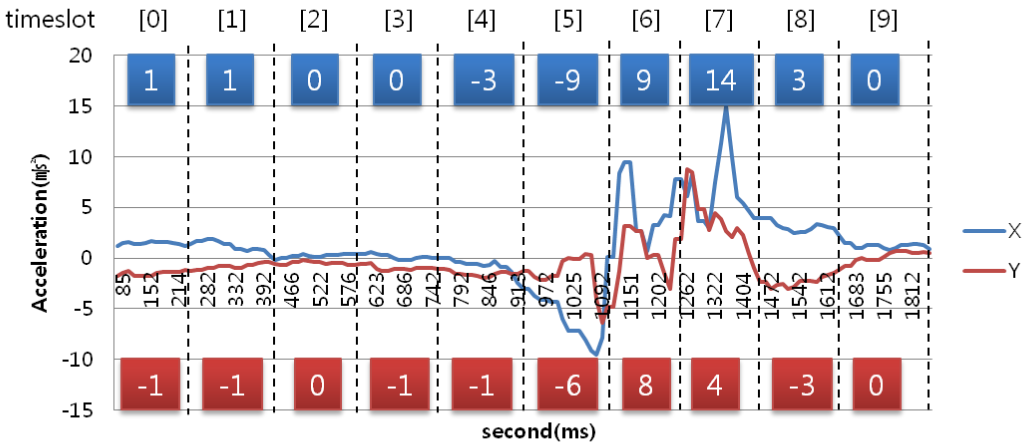
After collecting a swing data, the feedback is provided to the golfer automatically in the forms of sounds, icons and texts for the calculated swing duration from the backswing to the end point. The feedback consists of the x and y axes' acceleration values per timeslot, the total swing duration, each swing period such as the backswing, downswing and follow-through and timestamps of the maximum and minimum peak points.

5.6.1 Timeslots and acceleration values

If the critical points are detected and the swing duration is calculated, the swing time is subdivided into 10 timeslots and the representative acceleration values are calculated in each timeslot as described in 4.3.1. The reason of subdivision is to provide more detailed feedback to the golfer about his swing patterns.



(a) The original acceleration data



(b) The representative acceleration values in each timeslot

Figure 31. Timeslots and swing durations

The original acceleration data is shown in Figure 31(a) and the identified swing duration and the representative acceleration values in each timeslot is described in Figure 31(b). The representative values are the absolute maximum value in each timeslot. With these values, feedback is provided to the golfer with the corresponding icons and sounds.

5.6.2 Beep sound feedback

The beep sound feedback is concentrated on presenting the maximum and minimum acceleration values in a golf swing. This method generates different sounds and displays the pre-defined icons as mentioned in section 4.3.2. The green icon and the normal pitch sound are displayed in each timeslot except for the maximum and minimum peak points as described in Figure 32. This method can help the golfer to perceive the position of the two points.

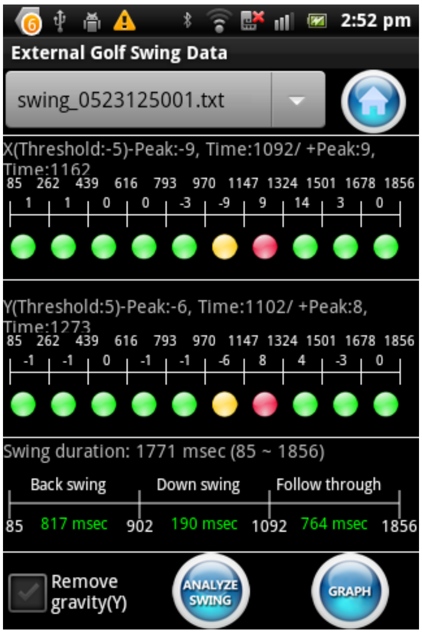
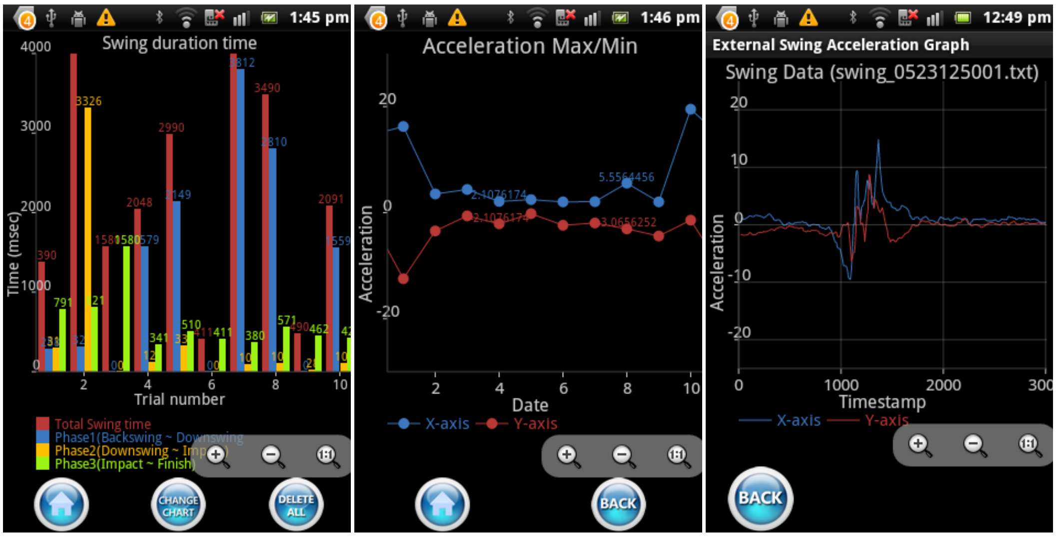


Figure 32. Beep sounds feedback

Each timeslot produces sounds during the calculated total swing duration and the related icons are displayed simultaneously. For instance, if the calculated swing duration of a golf swing is 1771 msec as shown in Figure 32, then each timeslot's feedback has 170 msec delays because the delay time is set after the swing duration is divided by 10. Consequently, the feedback time is entirely related to the swing duration. This mechanism helps the player to know his swing time without looking at the screen. In addition, the golfer can check the ratio of each swing period and the points of the maximum and minimum acceleration values.

5.6.3 Swing history with graphs

The database table described in section 4.4 is used for drawing two different types of graph as shown in Figure 33(a, b). The first graph displays the swing duration of each phase in milliseconds and the second line graph shows the changes of the maximum and minimum acceleration values. When there is a user's request, the software reads the related items from the database table and displays the graphs using them. This feedback provides the golfer with the record of his swing history to understand his swing patterns and the changes of improvement. In addition, the user can see the acceleration graph of each of his swing in the PastSwingData menu as shown in Figure 33(c).



(a) The swing duration time graph

(b) The acceleration graph

(c) An acceleration graph of a swing

Figure 33. Graphs for swing history

6. Evaluation and results

The purpose of this evaluation is to verify the software performance such as the probability of detecting critical points and the correctness of feedbacks.

6.1 The method of evaluation

There were two time field evaluations done by a SICS researcher on different days and had collected data from around 80 golf swings. To do this, he attached an Android smart phone to the backside of his waist as shown in Figure 34. This place was chosen because the golfer creates less noise here compared with the other parts of the body [24].

Figure 34. The placement of a smart phone

During the first test, the collection time was fixed at three seconds and feedbacks were given to the golfer after one second, but the player felt some inconveniences while testing. The collection time was too short because after his swing he wanted to see the trajectory of the ball and needed more time to prepare for the next swing. Therefore, a setting menu was added for the next evaluation and the golfer can choose the collection time according to his preference. In the second evaluation, the collection time was set to 10 seconds or 20 seconds and there was a 10 seconds break time after feedback to provide the golfer enough time to prepare for the next swing. The threshold values were set to -5 m/s^2 for the negative threshold and 5 m/s^2 for the positive threshold value during the two evaluations. Therefore, if the collected acceleration values were weaker than the threshold, the software ignores the swing data and displays an error dialog.

6.2 The result of evaluation

Functional verification and performance evaluation were carried out to accomplish the evaluation of the software. Table 10 shows the result of the two evaluations and illustrates the probability of detecting critical points out of 82 collected swing data. The number of swings being detected is 59 and the percentage is 71.9% but 23 swings were not detected due to various reasons.

Table 10. The percentage of detection rate

	Detectable swings	Undetectable swings	Total
Number	59	23	82
Percentage(%)	71.9%	28.04%	100 %

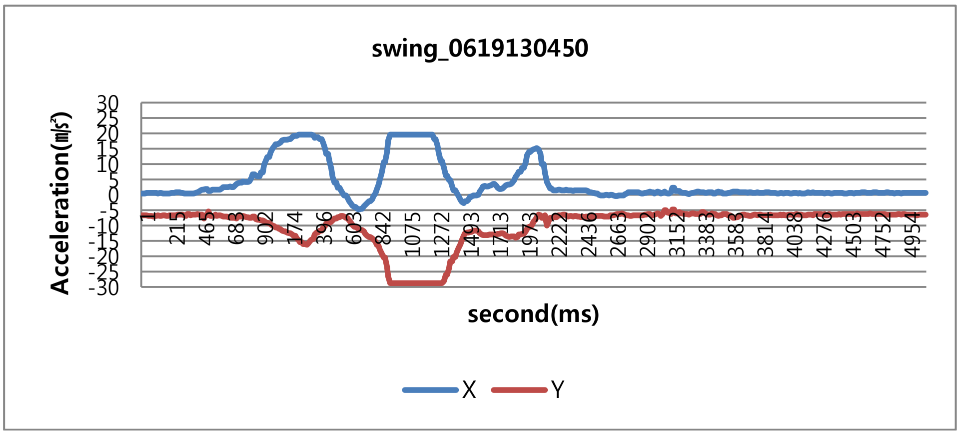
6.2.1 Reasons of detection failure

There are 23 swing data which cannot detect critical points using the software. The main reasons are investigated by the log messages of the software and the reasons are enumerated in Table 11. However, the detection rate can be different by changing the threshold values. In this evaluation, the two threshold values were set to -5 m/s^2 and +5 m/s^2.

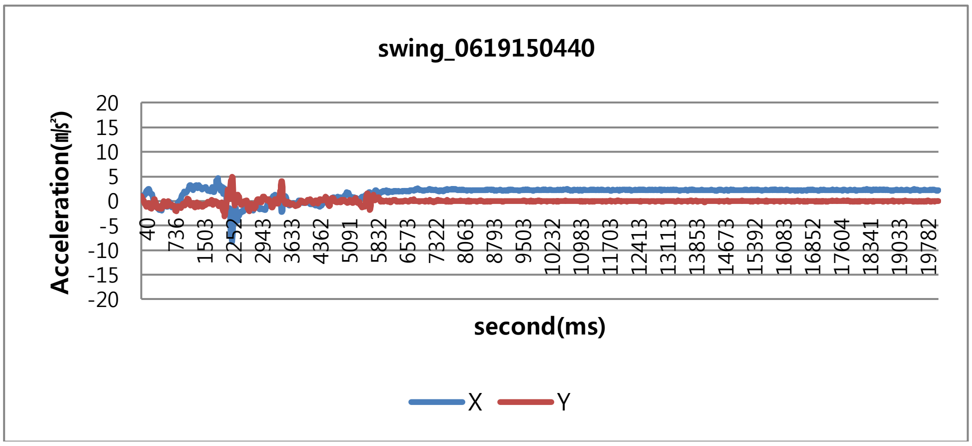
Table 11. The reasons of detection failure

Reasons	Number	Percentage
Weak swing (No data less than negative threshold (-5 m/s^2))	12	52.2 %
Weak swing (less than positive threshold (+5 m/s^2))	6	26.1 %
Undistinguishable swing	5	21.7 %
Total number	23	100 %

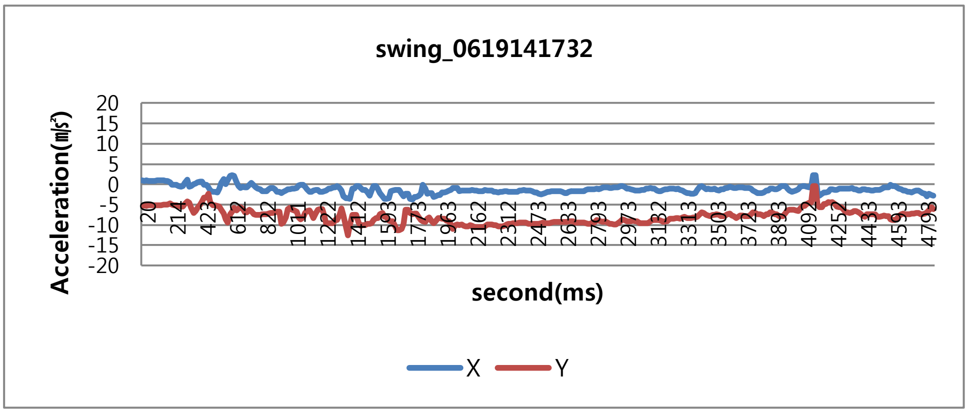
There are three types of reasons of detection failure and the examples are shown in Figure 35. The first graph (a) shows that the acceleration values of the x-axis are greater than the negative threshold. It seems that there are no negative values in the x-axis so the application cannot find the minimum peak point and it shows an error message about this swing. The second graph (b) represents that there is a minimum peak point but the positive values are less than the positive threshold value so it is also considered as a weak swing. The last graph (c) does not have any distinct peak points and the acceleration values are not clear.



(a) No negative values in the x-axis



(b) Less than the positive threshold in the x and y axes



(c) Undistinguishable swing

Figure 35. Three types of detection failure

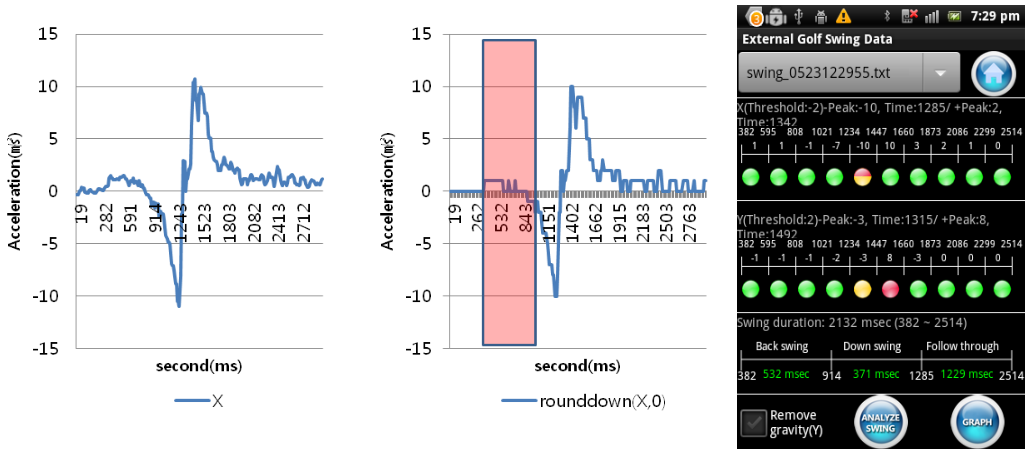
6.2.2 The result of detecting the backswing

Detecting the backswing period is still problematic because the movement of the golfer is smaller than other swing motions and it causes very undistinguishable acceleration values. Moreover, this period depends on the golfer's skill and swing patterns so the probability of detecting the backswing is 62.7% and the period is identified in 37 swings out of 59. Table 12 shows the percentage of detecting the backswing and other swing durations.

Table 12. The percentage of detecting swing durations

	Detectable swing data (total 59 swings)		
	Backswing	Downswing	Follow-through
Number	37	59	59
Percentage(%)	62.7%	100%	100%

Figure 36 shows the distinct backswing period which includes a non-zero area and a zero area. Accordingly, the software can identify its period from 382 msec to 914 msec as shown in Figure 33(c).



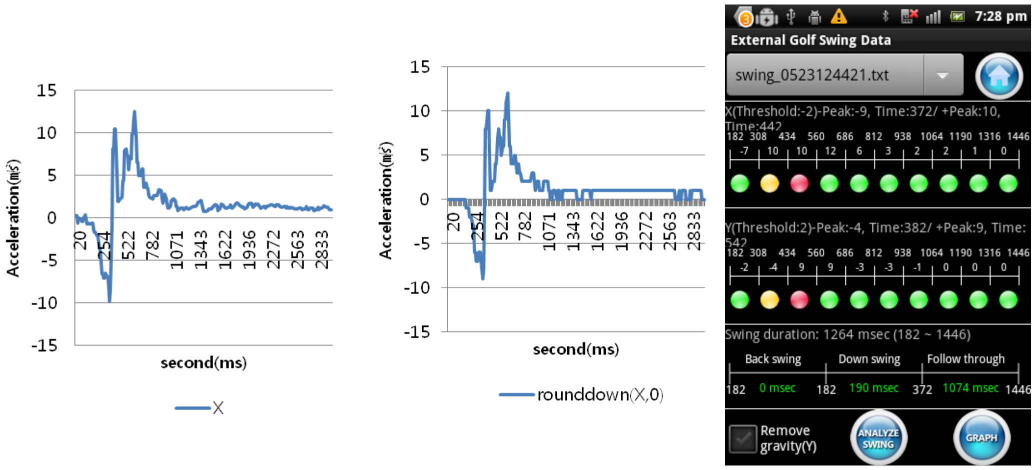
(a) Original swing graph

(b) Round-down graph (clear backswing period)

(c) backswing duration: (382~914 msec)

Figure 36. The case of successful detection of a backswing

On the contrary, Figure 37 represents that the swing data does not have an obvious backswing period unlike Figure 36. It does not have a non-zero area so the software does not consider the beginning part as the backswing duration.

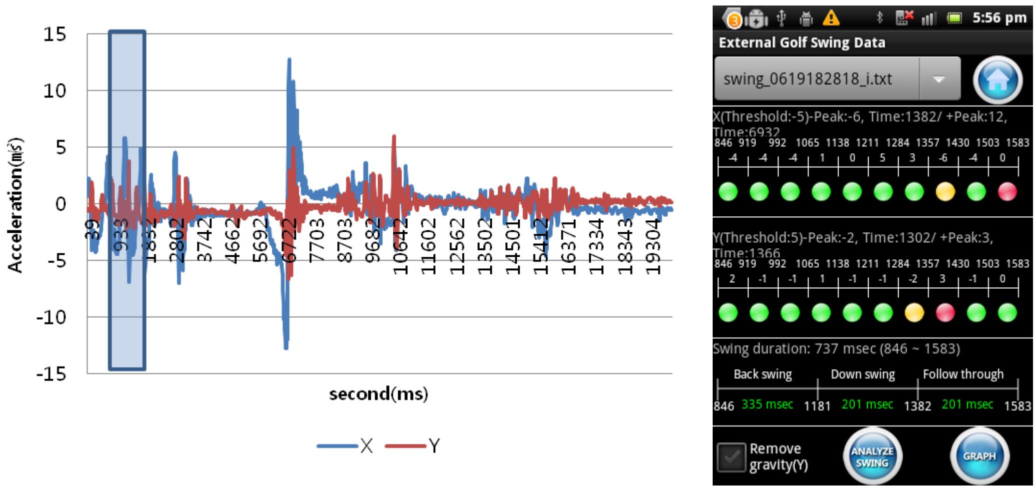


(a) original swing graph (b) round-down graph (c) backswing duration: 0 msec

Figure 37. The case of failure of detecting a backswing

6.2.4 The problem of choosing threshold values

The choice of threshold values for each golfer is an issue because the golfer's skills and power are very different. In addition, the software tries to detect critical points based on the pre-selected threshold values in the setting menu. According to the threshold values, the software identifies a different area as a real swing. If the thresholds are set to 5 and -5 ㎨, the colored area in Figure 38 is detected as a real swing because the movement looks like a golf swing and also the acceleration values are bigger than the threshold. Therefore, the software calculates swing period and gives feedback to the golfer based on the detected area even though there is a bigger swing motion later inside of the swing data.

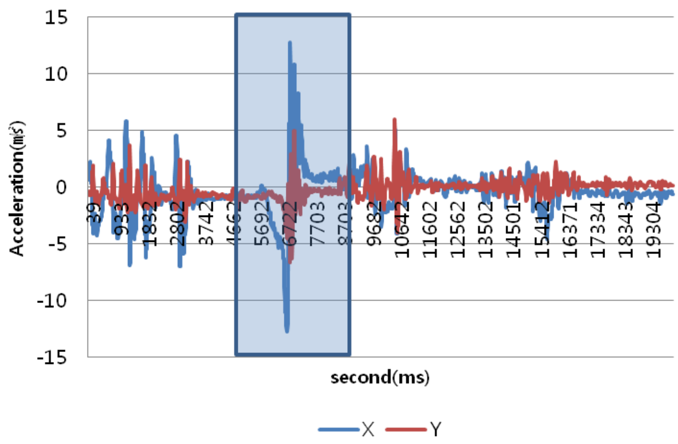


(a) The detected area (b) Result screen

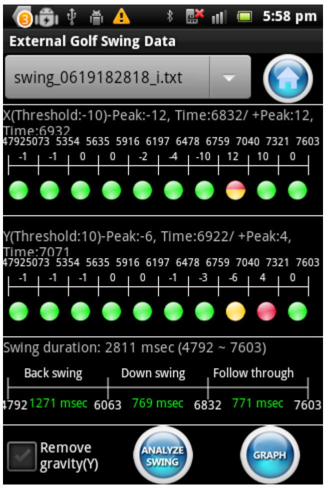
Figure 38. Threshold values are 5 and -5 ㎨.

In this case, the total swing duration is from 846 msec to 1583 msec, which consists of 335 msec for the backswing duration, 201 msec for the downswing and 201 msec for the follow-through as illustrated in Figure 38(b).

However, when the threshold values are set to 10 and -10 ㎨, the software detects the different area in the same swing data as described in Figure 39(a). This area is the real swing period and the calculated swing period is from 4972 msec to 7603 msec, which makes up 1271 msec for backswing duration, 760 msec for downswing and 771 msec for follow-through as shown in Figure 39(b). Therefore, golfers need to select the appropriate threshold values according to their strength and patterns.



(a) The detected area



(b) Result screen

Figure 39. Threshold values are 10 and -10 ㎨

7. Conclusion and future work

7.1 Conclusion

This study has shown the design and implementation of an Android application for measuring golfers' movements and identifying critical points of the golf swing using an accelerometer. Especially, this thesis focuses on the simple and novel ways for analyzing golfers' movement and the feedback related to the golfer's skill without any expensive or complicated devices.

The software evaluation was done by functional verification and performance assessment. The result showed that round 72 % out of 82 collected data were detected the critical points and calculated the swing durations successfully.

The main contribution of this thesis is that individuals who enjoy golf sport as entertainment have an opportunity of measuring and understanding their golf swing patterns without using expensive devices. In particular, other scientific research had aimed for elite and professional players, so, amateur players had difficulty in measuring their swing patterns and timings. However, this software can be accessible to them and be used conveniently using an Android smart phone.

In conclusion, this thesis provides a very simple and convenient way for golfers to comprehend their movement and improve their performance using an Android smart phone.

7.2 Future work

There are a few possibilities for improving this work. First, the selection of threshold values for each golfer needs careful attention. As mentioned in 6.2.4, the software can analyze different areas and give feedback to golfers with incorrect information according to the threshold values. Therefore, a heuristic algorithm of learning each golfer's suitable threshold values is required or some software functionalities for calculating golfers' appropriate threshold values are needed before measuring their movement.

Second, the identification of the backswing can be improved and other detection algorithms can be taken into consideration. Currently, the rate of detecting the backswing remains around 63% unlike other swing motions. What is worse, the backswing period has very small changes in the acceleration so it is very difficult to detect clearly. In addition, this movement is really dependent on the golfers. In order to improve the identification of the backswing, new approaches are needed such as using other sensors and measuring other parts of a body.

7.3 Social and Ethical Aspects

The contribution of this thesis is to provide entertainment to golfers during golf activities with a low-cost and convenient measurement system. All the tasks performed were considered to be beneficial to golfers and the sport itself. The author of this thesis has made the best effort to conform to the ethical policy of KTH in an upright and responsible manner.

7.4 Economic and Environmental Aspects

This thesis has been carried out not to cause any harm to the environment. Moreover, this thesis has focused on the design and implementation of an Android application to provide a reasonable measurement system to golfers.

7.5 Ecologically Sustainable Development

This study was completed by using an Android SDK which was opened to all developers and individuals. In addition, the important parts of the source code were shared in the appendix for future development. The source code can facilitate the development and implementation of some eco-friendly measurement systems in golf and other sports activities. The software was designed to utilize less power consumption and resource.

8. References

[1] Ed H.Chi, Gaetano Borriello, Guerney Hunt, Nigel Davies, "Pervasive Computing in Sports Technologies", Pervasive Computing, IEEE, 2005, pp 22-25.

[2] Isao Okuda, Philip Gribble and Charles Armstrong, "Trunk rotation and weight transfer patterns between skilled and low skilled golfers", Journal of Sports Science and Medicine, 2010, pp 127-133

[3] Heather Gulgin, Charles Armstrong and Phillip Gribble, "Hip rotational velocities during the full golf swing", Journal of Sports Science and Medicine, June, 2009, pp 296-299

[4] Kajiro Watanabe and Masaki Hokari, "Kinematical Analysis and Measurement of Sports Form", IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans, Vol.36, No. 3, May 2006

[5] Adrian M. Burden, Paul N. Grimshaw and Eric S. Wallace, "Hip and shoulder rotations during the golf swing of sub-10 handicap players", Journal of Sports Sciences, 1998, 16, 165-176

[6] Hiroshi Negoro, Masahiko Ueda, Kajiro Watanabe, Kazuyuki Kobayashi, and Yosuke Kurihara, "Measurement and Analysis of Golf Swing Using 3D Acceleration and Gyroscopic Sensors", SICE Annual Conference 2011, pp 1111-1114.

[7] Jake Crum, "Golf Swing Tempo a Result of Applied Physics", http://suite101.com/article/golf-swing-tempo-a-result-of-applied-physics-a143943, Engieering@suite101, August 30, 2009

[8] Robert D. Grober, "Measuring Tempo, Rhythm, Timing, and the Torques that Generate Power in the Golf Swing", arXiV: 1001.1137v1, January, 2010

[9] The definition of acceleration, Dictionary.com,

http://dictionary.reference.com/browse/acceleration

[10] Q4E case study 12-Acceleration,

http://www.quintic.com/education/case_studies/Acceleration.htm, Quintic 4 Education

[11] Sauvik Das, Latoya Green, Beatrice Perez and Michael Murphy, "Detecting User Activities using the Accelerometer on Android Smartphones", http://www.truststc.org/reu/10/Reports/DasGreenPerezMurphy_Paper.pdf

[12] Android Sensors overview,

http://developer.android.com/guide/topics/sensors/sensors_overview.html

[13] One Screen Turn Deserves Another,

http://android-developers.blogspot.se/2010/09/one-screen-turn-deserves-another.html

[14] Gang Chen, "How to Convert 3-Axis Directions and Swap X-Y Axis of Accelerometer Data within Android Driver",

http://cache.freescale.com/files/sensors/doc/app_note/AN4317.pdf,

Freescale Semiconductor Application Note, Document Number: AN4317

[15] Sonic Golf System-1, www.sonicgolf.com, Sonic Golf, Inc.

[16] Richard J. Jagacinski, Neil Greenberg and Min-Ju Liao, "Tempo, Rhythm, and Aging in Golf", Journal of Motor Behavior, 1997, Vol.29, No.2, pp 159-173

[17] Cheetham, P.J., Rose, G.A., Hinrichs, R.N., Neal, R.J., Mottram, R.E., Hurrion, P.D. and Vint, P.F. "Comparison of kinematic sequence parameters between amateur and professional golfers", Science and Golf V: Proceedings of the World Scientific Congress of Golf. Crews, D. and Lutz, R(Eds). Energy In Motion. Mesa Arizona, 2008, pp 30-36

[18] Math Strengthens the Swedish Olympic Cross-country Team, https://www.sics.se/files/SICS_AnnualReport_2011_FINAL.pdf, SICS Annual Report 2011-2012, pp 20-21

[19] Daniel Spelmezan, Mareike Jacobs, Anke Hilgers, and Jan Borcers, "Tactile Motion Instructions For Physical Activities", Proceedings of the 27th international conference on Human factors in computing systems, April 9th, 2009, pp 2043-2252

[20] SQLite, http://www.sqlite.org/

[21] Android SDK, http://developer.android.com/sdk/index.html

[22] AChartEngine, http://www.achartengine.org/

[23] Xperia™ Active, http://www.sonymobile.com/gb/products/phones/xperia-active/

[24] Koji Umemura, Koki Tachikawa, Yosuke Kurihara and Kajiro Watanabe, "Analysis of Human Walking by 3-D Acceleration Sensor and Its Application", SICE Annual Conference 2008, August, 2008, pp 1723-1726.

Appendix A: Detecting the minimum peak point of the x-axis

```
public int detectXMinPeakPoint(int index)
{
    int i=index;
    int minIndex = 0;
    int minValue = 0;
    boolean isFound = false;

    int x1, x2, x3;
    int arrSize = mSwingArrayList.size();

    x1 = x2 = x3 = 0;
    minValue = (int)mSwingArrayList.get(0).mXvalue;

    do
    {
        mCount++;

        x2 = getRoundDownValue(mSwingArrayList.get(i).mXvalue);

        if(i> 0)
            x1 = getRoundDownValue(mSwingArrayList.get(i-1).mXvalue);
        if(i < arrSize-1)
            x3 = getRoundDownValue(mSwingArrayList.get(i+1).mXvalue);

        if(x2 <= mMinThreshold)
        {
            if((x2 <= 0) && (x1 <= 0) && (x3 <= 0))
            {
                if(x2 <= minValue)
                {
                    minValue = x2;
                    minIndex = i;
                    isFound = true;
                }

                if(x2 == x3)
                {
                    i++;
                    continue;
                }

            }
        }

        if((isFound) && (x2<0) && (x3 >=0))
        {
            break;
        }
```

```
        i++;

    }while(i <= (arrSize-1));

    if(isFound)
    {
        return minIndex;
    }
    else
    {
        return DETECTION_FAIL;
    }
}
```

Appendix B: Detecting the maximum peak point of the x-axis

```
public int detectXMaxPeakPoint(int minIndex)
{
    int i = minIndex;
    int x1, x2, x3;

    int arrSize = 0;
    int maxIndex = 0;
    int maxValue = 0;
    boolean isPeakFound = false;

    x1 = x2 = x3 = 0;

    arrSize = mSwingArrayList.size();
    maxValue = (int)mSwingArrayList.get(minIndex).mXvalue;
    do
    {

        x2 = getRoundDownValue(mSwingArrayList.get(i).mXvalue);

        if(i> 0)
            x1 = getRoundDownValue(mSwingArrayList.get(i-1).mXvalue);
        if(i < arrSize-1)
            x3 = getRoundDownValue(mSwingArrayList.get(i+1).mXvalue);

        if((x2 >= mMaxThreshold))
        {
            if(x2 >= maxValue)
            {
                maxValue = x2;
                maxIndex = i;
            }
            if(x2 == x3)
            {
                i++;
                continue;
            }
            if((x2 > x3) && (x2 >= x1))
            {
                if(x2 >= maxValue)
                {
                    isPeakFound = true;
                    maxValue = x2;
                    maxIndex = i;
                }
            }

        }

        if((isPeakFound == true) && (x2 >= 0) && (x3 <= 0))
        {
```

```
                break;
            }

        i++;
    }while(i<= (arrSize-1));

    if(isPeakFound == false)
    {
        maxIndex = DETECTION_FAIL;
    }

    return maxIndex;
}
```

Appendix C: Detecting the downswing point

```
public int findDownSwingPoint(int peakIndex)
{
    int startIndex = 0;
    int i=0;
    int timestamp = 0;
    float x = 0;
    boolean isFound = false;
    int d = 0;

    i = peakIndex;

    do{

        timestamp = (int)mSwingArrayList.get(i).mTimestamp;

        x = mSwingArrayList.get(i).mXvalue;

        d = getRoundDownValue(x);

        if((x < 0) && (d==0))
        {
            isFound = true;
            startIndex = i;
            break;
        }

        i--;
    }while(i> 0);

    if(isFound)
    {
        return startIndex;
    }
    else
    {
        // If not found a start point, then return the first index
        return 0;
    }
}
```

Appendix D: Detecting the backswing point

```
public int findBackSwingPoint(int downIndex)
{
    int backSwingIndex = 0;
    int i=0;
    int timestamp = 0;
    float x = 0;
    boolean isBackSwingFound = false;
    boolean isTopSwingFound = false;
    int zero_count = 0;
    int non_zero_count=0;
    int d = 0;

    i = downIndex;

    do{

        timestamp = (int)mSwingArrayList.get(i).mTimestamp;

        x = mSwingArrayList.get(i).mXvalue;

        d = getRoundDownValue(x);

        if(d==0)
        {
            ++zero_count;

            if(zero_count >= TOP_SWING_CRITERION)
            {
                isTopSwingFound = true;
            }

            if(isBackSwingFound)
            {
                backSwingIndex = i+1;

                break;
            }

        }
        else
        {
            if(isTopSwingFound)
            {
                if(d > 0 || d < 0)
                {
                    ++non_zero_count;
                    if(non_zero_count >= BACK_SWING_CRITERION)
                    {
                        isBackSwingFound = true;
                        if(i == 0)
```

```
                        backSwingIndex = i;
                    }
                }
                else
                {
                    non_zero_count = 0;
                }
            }
            zero_count = 0;
        }

        i--;
    }while(i> 0);

    if(isBackSwingFound)
    {
        return backSwingIndex;
    }
    else
    {
        return downIndex;
    }
}
```

Appendix E: Detecting the end point

```
public int findSwingEndPoint(int peakIndex)
{
    int endIndex = 0;
    int i=0;
    int count = 0;
    int timestamp = 0;
    int arrSize = 0;
    float x, y;
    int zero_count_x = 0;
    int zero_count_y = 0;
    boolean isFoundEnd = false;
    int dx, dy;

    x = y = 0;
    dx = dy = 0;

    arrSize = mSwingArrayList.size();

    for(count = peakIndex; count< mSwingArrayList.size(); count++)
    {

        timestamp = mSwingArrayList.get(count).mTimestamp;

        x = mSwingArrayList.get(count).mXvalue;
        y = mSwingArrayList.get(count).mYvalue;

        dx = getRoundDownValue(x);
        dy = getRoundDownValue(y);

        if((dx == 0))
        {
            ++zero_count_x;

            if((zero_count_x == END_CRITERION))
            {
                endIndex = count-(END_CRITERION-1);
                timestamp =
                         (int)mSwingArrayList.get(endIndex).mTimestamp;
                isFoundEnd = true;
                break;
            }
        }
        else
        {
            zero_count_x = 0;
        }

    }


    if(isFoundEnd == false)
```

```
    {
        endIndex = arrSize-1;
    }

    return endIndex;
}
```

Appendix F: Round-down operation

```java
public int getRoundDownValue(float x)
{
    int value = 0;

    if(x > 0)
        value = (int)Math.floor((double)x);
    else
        value = (int)Math.ceil((double)x);

    return value;
}
```