

國立交通大學

網路工程研究所

碩士論文

建立以肢體感測網路為基礎之
虛擬社群應用服務平台



Building an Interactive Virtual-Physical Social Network
Platform

研究生：黃辰少

指導教授：曾煜棋 教授

易志偉 教授

中華民國 1 0 1 年 1 月

建立以肢體感測網路為基礎之虛擬社群應用服務平台
Building an Interactive Virtual-Physical Social Network Platform

研究生：黃辰少

Student : Chen-Shao Huang

指導教授：曾煜棋

Advisor : Yu-Chee Tseng

易志偉

Advisor : Chih-Wei Yi

國立交通大學
網路工程研究所
碩士論文



Submitted to Institute of Network Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

September 2010

Hsinchu, Taiwan, Republic of China

中華民國九十九年九月

建立以肢體感測網路為基礎之虛擬社群應用服務平台

學生：黃辰少

指導教授：曾煜棋 老師
易志偉 老師

國立交通大學網路工程研究所碩士班

摘 要

近來社群網路吸引了許多關注，以及許多的身體資訊可以使用無線感測網路擷取。此篇論文將肢體感測網路與社群網路結合，與傳統的社群網路相比，提供更多的使用者之間的肢體互動。我們提出一個創新的肢體感測網路為基礎之虛擬社群應用服務平台。在此平台上，我們設計了以肢體感測網路為基礎之應用程式介面，可以幫助開發者在社群平台上設計更多有趣的以肢體感測網路為基礎之服務，並且我們使用了這些應用程式介面開發了一些應用。基於會有更多的應用會在我們的平台上開發，由於肢體無線網路頻寬有限，傳統的資料收集協定不能提供高解析度的感測資料。所以在此平台上，我們設計了一個資料收集協定，依照感測器的資料分配每個感測器的重要性，再依照不同的重要性來收集感測器資料，而此協定不需知道上層的應用。最後，我們在 Facebook 實作了此平台，以及在實驗中，研究我們設計的資料收集協定的效能。

關鍵字：肢體感測網路、數位物理系統、行動運算、社群網路、無線感測網路

Building an Interactive Virtual-Physical Social Network Platform

Student : Chen-Shao Huang

Advisors : Dr. Yu-Chee Tseng
Dr. Chih-Wei Yi

Institute of Network Engineering
College of Computer Science
National Chiao Tung University

ABSTRACT

Web-based social networks have attracted a lot of attention. While more physical information can be captured by wireless sensor networks, this paper incorporates remote body sensor networks (BSNs) into such web-based social networks to provide more physical interaction between users than conventional social networks. An innovative interactive virtual-physical social network platform is proposed in this work. In our platform, we design a set of BSN-based APIs to help developers design more interesting BSN-based services on social networks. By adopting these APIs, some applications are designed in our platform. Since various BSN-based applications will be built on our platform, conventional data collection protocols can not provide high-resolution sensing data in a limited bandwidth BSN. We then design a data collection protocol for BSNs in our platform, where the weights (i.e., importance) of sensor nodes are assigned dynamically based on the distribution of sensing data and BSNs do not need to know what applications will be provided in the upper-layer social networks. Finally, we implement our platform on Facebook and show experimental results to investigate the performance of our data collection protocols.

Keywords: body sensor networks, cyber-physical systems, mobile computing, social networks, wireless sensor networks

誌 謝

在研究所近兩年的日子裡，承蒙曾煜棋老師與易志偉老師的教導，讓我得以順利完成本論文。與老師的討論以及細心的指導，使我在不斷的失敗中，讓我學習到正確的研究方法與態度。在口試期間承蒙林丁丙教授與周立德教授提供許多寶貴的建議，使此篇的內容可以更完整且細膩。

特別感謝芳璟學姊的細心指教，使我的研究方向與方法更加明確，在學姊有耐心且細心的帶領下，才能讓我順利的將碩士論文完成，實在萬分感激。感謝倫武學長及阿良提供先進的設備讓我順利的完成研究。並且感謝實驗室所有的學長姊在我遇到問題時給予我適時的建議。

而在我的同儕中，感謝大家陪我度過研究所生活。我永遠不會忘記，一個可以打十個的修齊時常給予我協助，還有勇甫許多經典的話，以及東薑、志偉、Nowfats，讓我學習到團隊合作的重要性。水哥的說書、PO 的網路不好、小豪的超強籃球，Pony 最快，大家陪我一起測試了一年的交大網路。還有能遇到同月同日生的 Duck，真是幸運。家瑜是大正妹。感謝毛毛能與我一起畢業，同甘共苦！

最後，感謝我親愛的家人，媽媽、阿公、阿嬤提供我衣食無缺的協助，以及媽媽天天的電話關心。特別感謝舅舅、舅媽與陸皮，每週末帶我去補充營養以及戶外踏青，使我在辛苦的研究生活中不至於營養不良。另外，要非常感謝我的女朋友靖雯不離不棄，能在我忙碌的研究所生活中，犧牲出去玩的時間陪我度過。謝謝爸爸的保佑，讓我順利的畢業。

黃辰少 謹誌

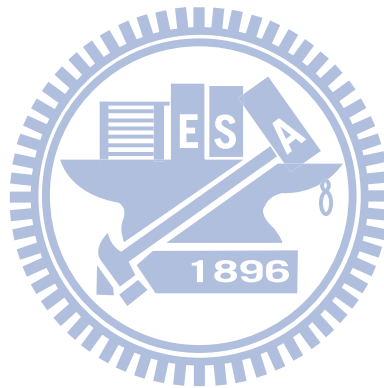
國立交通大學網路工程研究所

高速通訊與計算實驗室

中華民國九十九年九月

Contents

摘要	i
Abstract	ii
誌謝	iii
Table of Contents	iv
List of Figures	v
1 Introduction	1
2 Related Works	5
3 System Architecture	7
3.1 Software Architecture Overview	9
3.2 BSN APIs on Social Network Platforms	11
3.3 BSN Applications on Social Network Platforms	15
4 Data Collection Protocols	20
5 Implementation & Experimental Results	27
5.1 Implementation	27
5.2 Experiment Results	30
6 Conclusions & Future Works	35
Bibliography	36



List of Figures

1.1	Virtual-Physical Spectrum.	2
1.2	Different Data Collection Method.	4
3.1	System Architecture.	8
3.2	System Architecture.	10
3.3	A Walking Pattern.	10
3.4	The Definition of the Three Angles.	11
3.5	Set Rules in the Database.	12
3.6	The Flow Chart of Static Reporting API.	13
3.7	The Rotation Matrix of Rendering API.	13
3.8	The Flow Chart of Rendering API.	14
3.9	The Definition of A Pattern.	15
3.10	The Flow Chart of Pattern Recognition API.	16
3.11	The User Interface of Posture Messenger.	16
3.12	A Example of Static Reporting API.	17
3.13	The User Interface of Tai-Chi.	18
3.14	A Example of Rendering API.	18
3.15	The User Interface of Running Game.	19
3.16	A Example of Pattern Recognition API.	19
4.1	Different Variance for Different Human Body.	21
4.2	Our Method of Data Collection.	22
4.3	The Definition of A Polling Round.	22
4.4	A Polling Round Is Allocated Only One Sensor.	23
4.5	To Maximize the Polling Round of the Sensor.	24
4.6	Allocate Polling Rounds of Sensor 1, 2 and 3.	25

4.7	The Reporting Control Scheme.	26
5.1	The Data Flow of Servers with Three Applications.	28
5.2	The User Interface of Selecting Three Application.	29
5.3	The Hardware of the Sensor and the Sink.	29
5.4	The Wear Method of Nine Body Sensors and A Sink.	30
5.5	Our Data Collection Protocol Performance(Number of Samples vs. Variance).	32
5.6	Polling Protocol Performance(Number of Samples vs. Variance).	32
5.7	Pure Contention Protocol Performance(Number of Samples vs. Variance).	33
5.8	Polling Rounds vs. Number of Samples.	33
5.9	Compare Throughput.	34



Chapter 1

Introduction

Recently, technologies of wireless communications and mobile computing open up an opportunity for web-based social networks, e.g., Facebook [4], YouTube and Windows Live [14]. Many fancy web services on these social networks have attracted extensive attention, where users can share various content, e.g., photos, videos, messages and essays, in their daily life. However, the interaction on these conventional social network platforms are still *virtual* due to the following constraints. First, the shared content is generated mutually rather than automatically. Second, users only can have virtual interaction with each other by using mouses, keyboards, touch panels or cameras. Finally, since users must excel in the operation of computers to make a specific file format (e.g., jpg format for images or mpg format for videos), it is hard to older generation and may not join these social networks.

Instead of virtual information on these web-based social networks, the progress of wireless sensor networks (WSNs) opens up an opportunity for providing physical and real-time information in users' daily life. A WSN is composed of a sink node and a set of sensor nodes. Sensor nodes are capable of sensing *physical* information and communicating with each other, while the sink node is responsible for collecting sensing data from all the sensors. In recent years, body sensor networks (BSNs) have paid attention extensively, where sensors nodes are worn on users' body to monitor physiological information of human bodies, such as heartbeat and gestures. So, many BSNs research issues and applications are developed such as motion capture, virtual games and health monitoring [7][22][25]. However, in conventional BSNs, since data dissemination flows is one-way (i.e., from users to a centralized server), remote BSNs cannot have social interaction with each other.

Motivated by the above mentions, we design an *interactive virtual-physical social network*

platform, which integrates remote body sensor networks (BSNs) into conventional social networks. In the innovative platform, body information can be automatically captured by BSNs, transformed into specific knowledge (e.g., motions), and then shared in the social networks. By this way, users can have *physical* interaction with friends by sharing sensing data in a *virtual* social network. For example, a group of remote users can do exercise at the same time. Fig. 1.1 shows a spectrum of communities from physical to virtual worlds. In physical communities, we must meet together to have physical interaction with friends such as chatting, doing Tai-Chi exercise or having a race. On the contrary, in virtual communities, we do not meet together and can have virtual interaction with each other via social networks such as MSN [15] or on-line games. By combining the features of virtual and physical communities, users can have physical interaction with each other in virtual social networks without meeting together. For example, users can share their status based on gesture recognition, have a race by evaluating their motion, or do Tai-Chi exercise by transforming sensing data in a visual way. Compared to conventional sensor-based [20] or camera-based [8] motion capture systems, our system has less physical constraints, so users can interact with friends almost anytime anywhere.

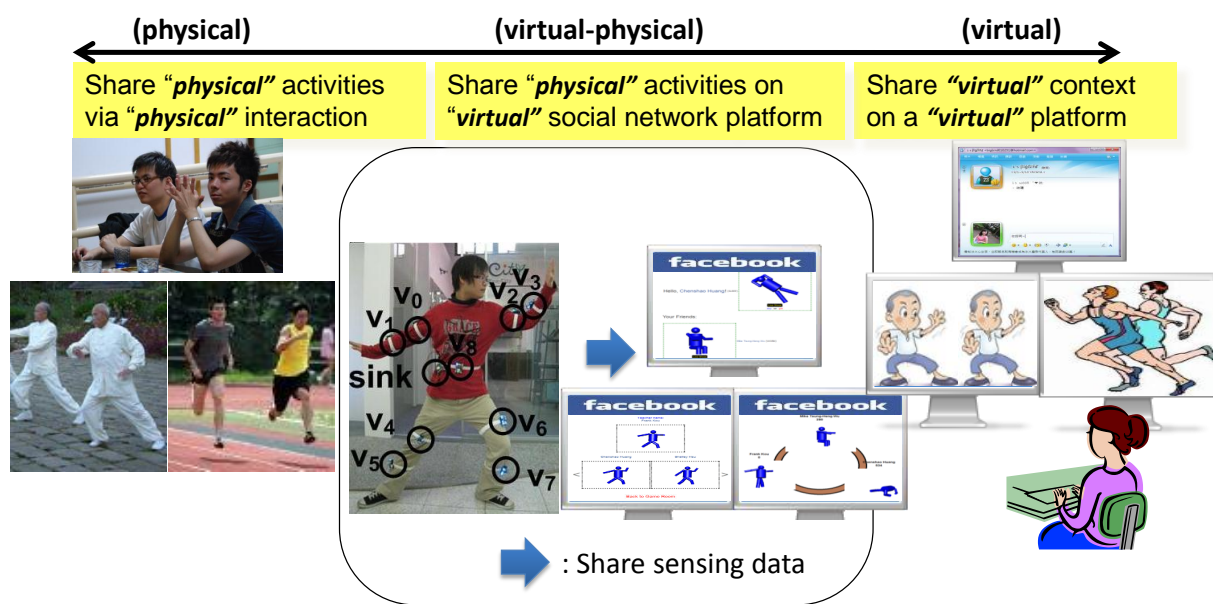


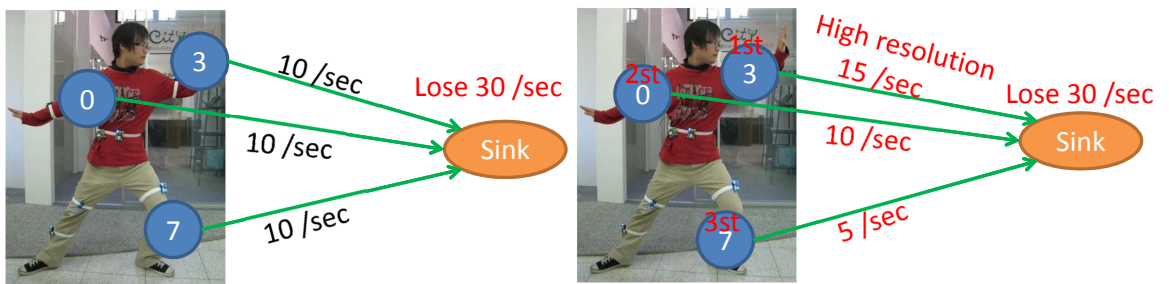
Figure 1.1: Virtual-Physical Spectrum.

In this paper, we propose a new framework to combine BSNs and social networks and design a set of application programming interfaces (APIs) including status identify, motion rendering and pattern recognition to help developers extend various BSN applications on social networks.

We also adopt these APIs to implement three applications on our platform. In our platform, each user wears nine sensor node and a sink node. Each sensor node includes a 3-axis accelerator and a digital compass to capture users' motions. Each user's sensing data will be collected by the sink node and then input to social networks via 3G/WiFi networks.

In our platform, *high-resolution* sensing data must be collected for representing users' motions. Lossless data collection is usually preferred, where all motion samples must be collected to the sink node. However, in a BSN, to capture smooth motions, a high sampling rate is necessary. In addition, all nodes in a BSN are usually fully connected. So, data loss is inevitable due to limited bandwidth in a BSN. Fig. 1.2(a) shows an example, where the sampling rate of sensors is 20 samples/sec and the available bandwidth is 30 samples/sec. When a conventional data collection mechanism is adopted (i.e. all sensors nodes have the same weights), each sensor node can only contribute sensing data at the rate of 10 samples/sec. On the other hand, to assign the same weight to sensor nodes cannot always collect more high-resolution sensing data. For example, when a user is doing Tai-Chi exercise, motions on his/her arms have higher variation than that on thighs/legs. This will be also reflected on sensors' sensing data and imply that the sensing data with higher variation is more important. If we allocate available bandwidth based on the importance of sensor nodes, higher-resolution data will be collected. Fig. 1.2(b) shows an example, where sensor node 1 has the most important data. Since various BSN-based applications will be built on our platform, a static weight assignment of sensor nodes cannot adapt to different applications. We then design a data collection protocol for BSNs in our platform, where the weights (i.e., importance) of sensor nodes are assigned dynamically based on the distribution of sensing data and BSNs do not need to know what applications will be provided in the upper-layer social networks.

The rest of this paper is as follows. In Chapter 2, we survey related work. The system design of our interactive virtual-physical social network platform will be detailed in Chapter 3. In Chapter 4, we design our data collection protocol. Chapter 5 shows the implementation and the experiment results of our system. Chapter 6 concludes this paper and points out our future work.



(a) The method of traditional data collection (b) The method of important data collection

Figure 1.2: Different Data Collection Method.

Chapter 2

Related Works

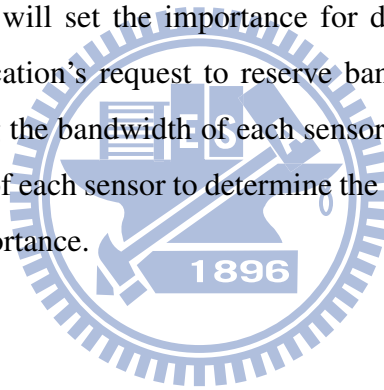
Related works are divided into three categories that includes social networks, body sensor networks and communication protocols. Social networks are introduced the recent main issue of social networks. Body sensor networks are introduced the related work of our paper technology. We divide body sensor networks into three parts following different areas that includes architecture, Pattern matching and Motion rendering. The communication protocol technology in the BSN is quite matured, we will introduce the related work of communication protocols which we use the more relevant.

The recent main issue of social networks, the main research is user behavior analysis and data mining such as [24][16]. In the reference [24], there is created generic APIs on social networks. There provides collected API of user's community information, and then uses these community information to calculate the similarity between two communities for the developer. Our work provides the API of social network with BSN for the developer.

The architecture of body sensor network areas, related works of BSN communication architecture have [13][3][5][1][6][9]. BSN platforms have [18]. In the reference [13], the main research is pattern detection of the body sensor sensor architecture. This architecture can detect the pattern by users defined. The pattern is generated by the feature extraction of the raw data, and each sensor will be matched the pattern of users defined. Unlike [13], our method of pattern detection is that the raw data is detected in the database. In the pattern matching area, in the reference [12], the main research is pattern recognition that can recognize the user's gesture. The main technology is that the user's sensing data compares the wave similarity with the template in the database. First, users use the wii remoter to read the raw data, and use DTW algorithm to calculate the minimum distance between current raw data and the template in the database.

Calculate all templates, and the minimum distance of all templates is recognized that template. We use pattern matching to recognize the user's gesture different from [12], in the reference [12] that only supports single sensor, we can support multiple sensors. In the motion rendering area, in the reference [21], the main research is using Extended Kalman Filter to offline rendering the human 3D model. This algorithm's contribution is that can combine three acceleration, compass, ultra wave to render the human 3D model. Our rendering technology is different from [21], in the reference [21], that stores raw data to the hard disk and offline renders the human 3D model, but we are real-time rendering the human 3D model. Furthermore, in the reference [21], their algorithm computing is very large, our sensors do not have the capacity of real-time computing, therefore, our designed rendering algorithm is simple.

The communication protocol of our related has [26][17][2][11]. The energy-efficient area has [23][19][17][10]. In the reference [26], the main research is to provide the QoS of priority data. First, the application will set the importance for different sensors, and then the QoS scheduler follows the application's request to reserve bandwidth for each sensor. Sink will collect sensor data following the bandwidth of each sensor. Our protocol is different from this work, we are using the data of each sensor to determine the sensor's importance, the application does not need to set the importance.



Chapter 3

System Architecture

This section introduces our proposed virtual-physical system architecture. Our system architecture is divided into hardware and software architecture. Our social network platform needs a software architecture to combine existing social network platform and our body sensor network needs a hardware architecture to collect user's sensor data. Through this combination of hardware and software system architecture, it combines the body sensor network and the social network and automatic sharing the sensor data through the application on our social network platform.

Fig. 3.1 is our system architecture. The system data flow is right to left. The system architecture consists of three parts that includes the client, the social network and the body sensor network. First, the human body should be worn body sensors and a sink, they used to sense the user's sensing data such as working, sleeping and running. The sink will collect the user's sensor data and then return to the social network via wireless or wired network such as 3G, Wifi or Ethernet.

The social network consists a group of different functional server. It divided into two parts, the first part is the social network server.

Social Network Server: It is provided by social network platform such as Facebook. This platform has the user data and the user's friend list. It allows us to develop the application on this platform.

The second part is three different functional servers for we combine social network and body sensor network.

Body Sensor Network Server: Use to receive user's sensing data and to write user's sensing data to database server.

Database Server: Use to record user's sensing data and to store user's information of applications.

Web Server: Web server provides client to use applications on our social network. It maintains user's information of applications in database server. We develop the application through our APIs in web server.

Client is a social network user. He/she can login social network platform via internet capability devices such as desktop, laptop and smart phone. Social networks platform will redirect client to our web server and then client can use the application.

The remain section is divided into three parts of the software architecture. In Section 3.1, the first part is the software architecture overview that illustrates overall structure of the software architecture and functions of each component in structure. In Section 3.2, the second part illustrates designed details and usages of BSN APIs in the software architecture. In Section 3.3, the third part introduces the application of using BSN APIs. The hardware architecture illustrates in Chapter 5.

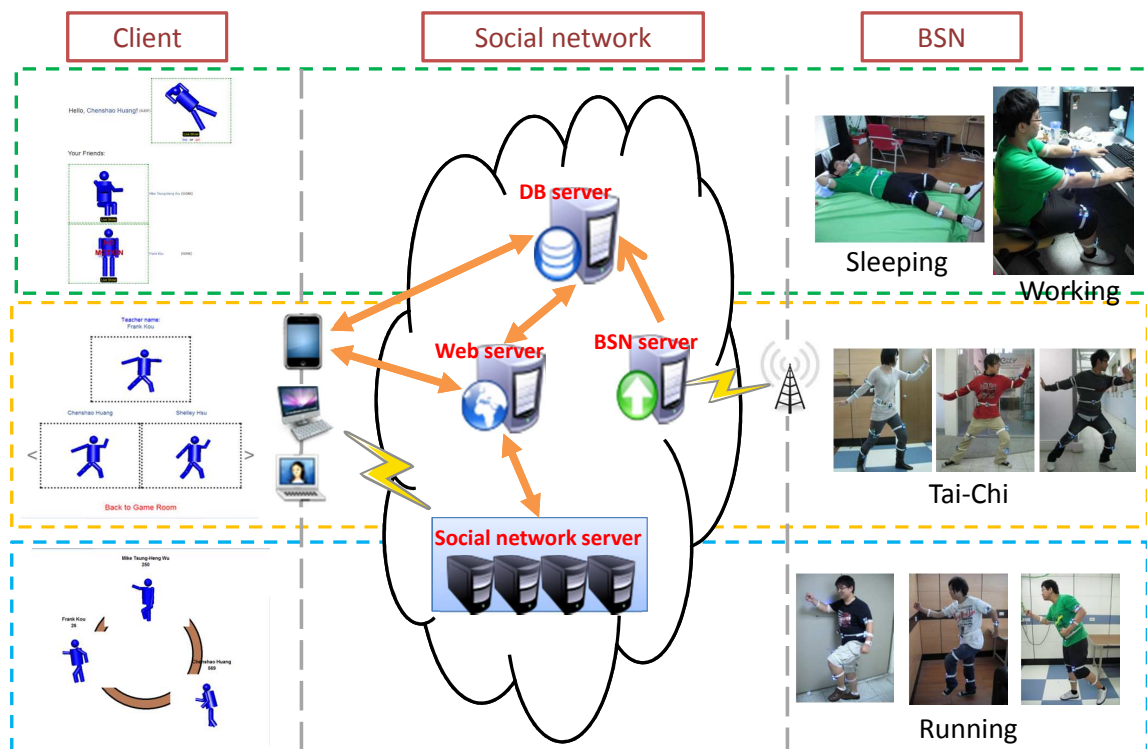


Figure 3.1: System Architecture.

3.1 Software Architecture Overview

In software architecture, we provide three different functional BSN APIs and three combined application based on these BSN APIs. Fig. 3.2 shows our software architecture, it is divided into two parts, the first part is the social network with BSN that contents three BSN APIs and three applications. The second part is the communication protocol that contents the method of sensor data collection. By this protocol, top of software architecture does not need to adjust the method of data collection. This protocol will introduce in Chapter 4.

The first part is the social network with BSN, the difference of this architecture with the traditional social network is different inputs. Inputs of the traditional social network are mouse, keyboard and webcam, but the input of the social network with BSN is user's sensing data.

We continue three roles of the traditional social network:

Social Network Platforms: It is provided by social network platform company. The platform provides platform APIs that developer can use these APIs to develop applications. Social network users can interact with friends by these applications.

Social Network Users: The social network users can interact with friends by using BSN applications.

Social Network Developers: The developer can develop applications by using our provided BSN APIs on social network platform and then users interacts with friends by these applications.

We provide BSN APIs to developer on our platform that called community service API. We design three BSN APIs that let us and developers to develop BSN applications on social networks.

Status Reporting API: This API allows developers to know the current status of the user such as working, sleeping, call phone and running. These statuses are defined by the user.

Rendering API: This API allows developers to show the current gesture of the user by 3D rendering engine.

Pattern Recognition API: This API allows developers to calculate how many times repeat-motion appears. A pattern is defined a sequence of many motions. Fig. 3.3 shows a walking pattern that is a sequence of three motions and this APIs will calculate how many times this pattern of the user appears.

The developer uses these APIs to develop applications. The detail of APIs will illustrate in

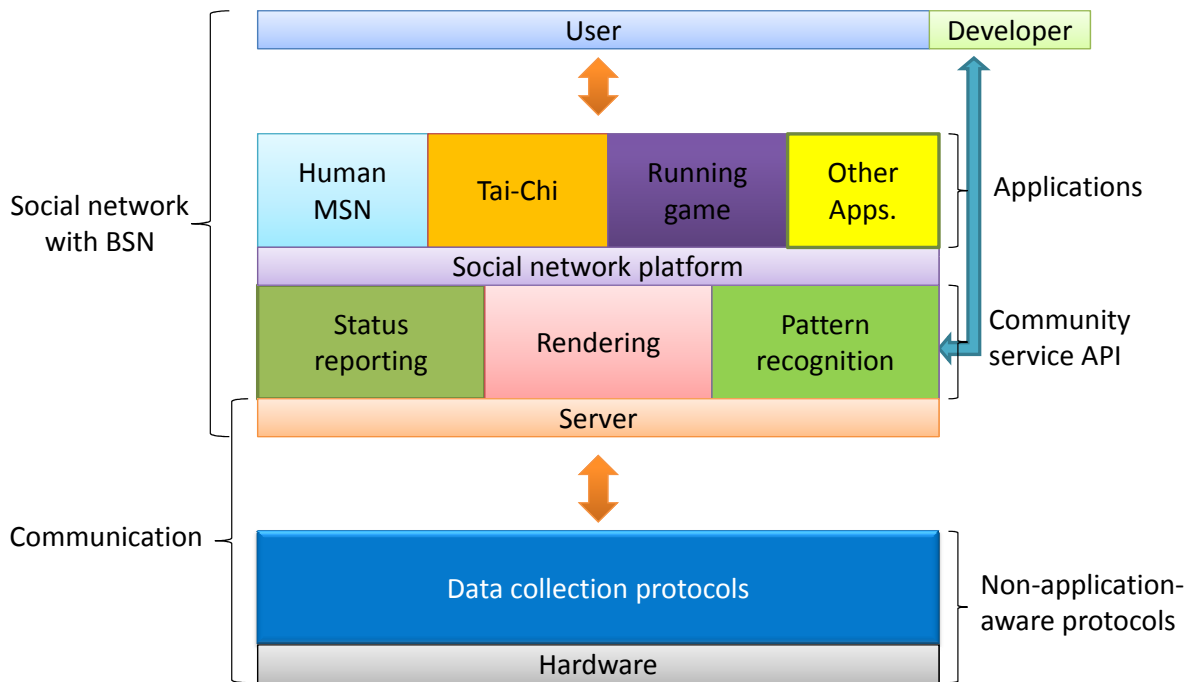


Figure 3.2: System Architecture.



Figure 3.3: A Walking Pattern.

Section 3.2. We develop three applications of using these APIs that show in Section 3.3.

We must know the meaning of the user's sensing data before we introduce BSN APIs. Each sensor senses Euler angle that includes Roll angle, Pitch angle, Yaw angle from human body. Each sensor has same fixed sensor coordinate, and Euler angle means three included angles with the axes of this coordinate. A object must follows the fixed rotation sequence, our rotation sequence is Y, X and Z. Follow the rotation sequence, the object will be rotated to correct position on this coordinate. The following shows the definition of the three angles:

Pitch Angle: This angle is right-handed rotation $\theta \in (-180, 180]$ about the y-axis by the roll angle.

Roll Angle: This angle is right-handed rotation $\phi \in [-90, 90]$ about the new (once-rotated) x-axis by the roll angle.

Yaw Angle: This angle is right-handed rotation $\varphi \in (0, 360]$ about the new (twice-rotated)

z-axis by the pitch angle.

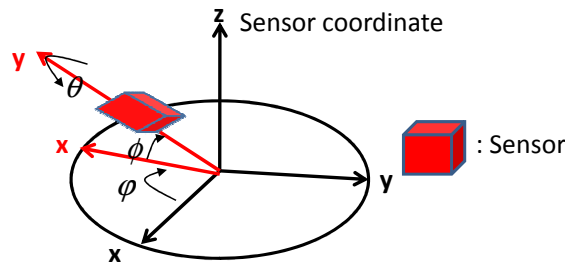


Figure 3.4: The Definition of the Three Angles.

We use Euler angle to restore three axes of human body and then our designed APIs uses Euler angle to reach different objectives such as status identification, motion rendering and pattern recognition.

3.2 BSN APIs on Social Network Platforms

This subsection will introduce the designed detail and usages of three BSN APIs. Three BSN APIs includes status reporting API, rendering API and pattern recognition API. The function of status reporting API provides developer to identify current user's status. The function of rendering API provides developer to show current user's gesture by 3D rendering engine. The function of pattern recognition API provides developer to calculate how many times the pattern of the user appears.

Our BSN APIs usages are divided into two phases. The first phase is offline input phase when developer setting BSN APIs before using BSN APIs. The second phase is online reporting phase when developer is using BSN APIs, and we will introduce the flow chart of the each BSN API.

The following will illustrate the details and the usage of the each API:

Status Reporting API: This API provides developer to identify current user's status. The developer should set the status into database server before using this API and then using the method of pattern matching identifies current user's sensing data whether is matching with all statuses in the database. If they are matching, then return the status name in the database. This API can be used: automatic detection of the state, danger detection.

Our identified method is using rule-based method that identify current user's sensing data

whether is matching with setting rules in the database. Fig. 3.5 shows the setting rule in database, we can set the Euler angle and the threshold of the corresponding node ID. If current user's sensing data is matching with one of the rule in database, it will return the name of the match rule. If there are matching multiple rules, it will return all the name of multiple match rules.

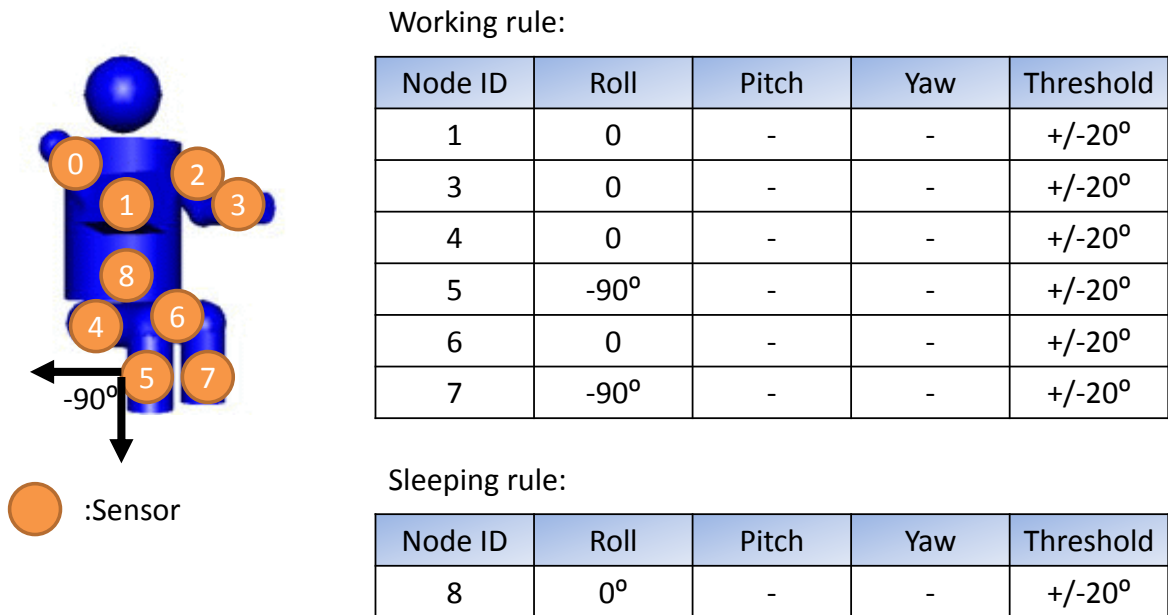


Figure 3.5: Set Rules in the Database.

Fig. 3.6 shows the developer using the method of status reporting API. First, the developer uses Set rule API to input the each rule on offline input phase. The developer inputs the status name that is the rule name, and inputs the Euler angle of the corresponding node ID. The developer calls Read status API to identify current user's status on online reporting phase, Read status API will match the user's sensing data and all the rule in database. If there have the match rule, then return the name of the match rule.

Set rule \langle Status name, Sensor ID, Roll, Pitch, Yaw \rangle

Status name = Readstatus \langle User ID \rangle

Rendering API: This API provides the developer to render the gesture of the user by 3D rendering engine. The developer can real-time render the gesture of the user using the user's sensing data. 3D model should be updated immediately when the user's sensing data updates. This API can be used: Show the user's 3D model.

Fig. 3.7 shows our the method of rendering the user's gesture. First, we draw a 3D model

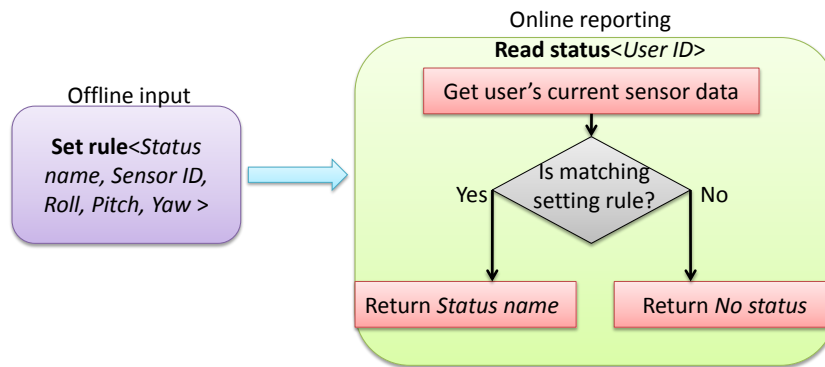


Figure 3.6: The Flow Chart of Static Reporting API.

on the graphics coordinate that is corresponding the sensor in the human body. For example: The sensor is worn on the arm, we draw a arm's 3D model on the graphics coordinate. Our goal is rotating the arm's 3D model to the same location with the sensor. We use rotation matrix to rotate the arm's 3D model, and following our rotation sequence Y ,X and Z rotates Euler angle of the sensor. The vector b means the original coordinate of the arm's 3D model, The vector b is multiplied by the rotation sequence Y ,X and Z. Finally, we get the vector b' after rotating the coordinate.

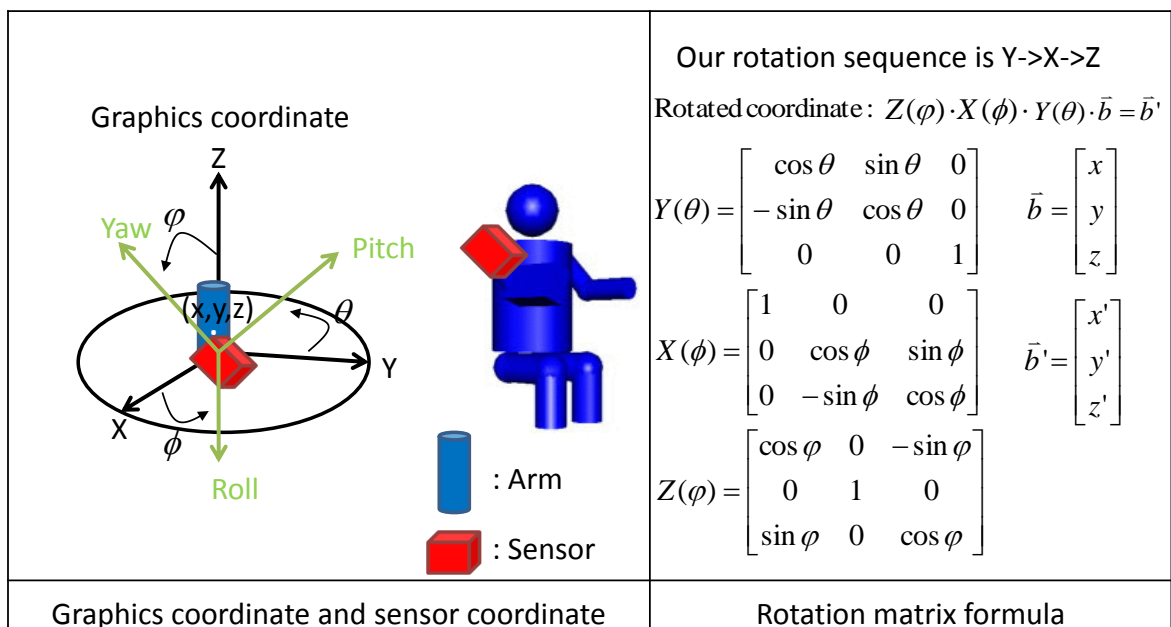


Figure 3.7: The Rotation Matrix of Rendering API.

Fig. 3.8 shows the usage of rendering API. Rendering API don't need setting so that this

API don't have offline input phase. We call the draw gesture API on online reporting phase, Draw gesture API will obtain the original 3D model coordinates of the vector b , and using the rotation matrix to get the vector b' after rotating the coordinate. Use this vector b' to update the arm's 3D model and the model is current user's gesture now.

Draw gesture $\langle User\ ID, Sensor\ ID, Roll, Pitch, Yaw \rangle$

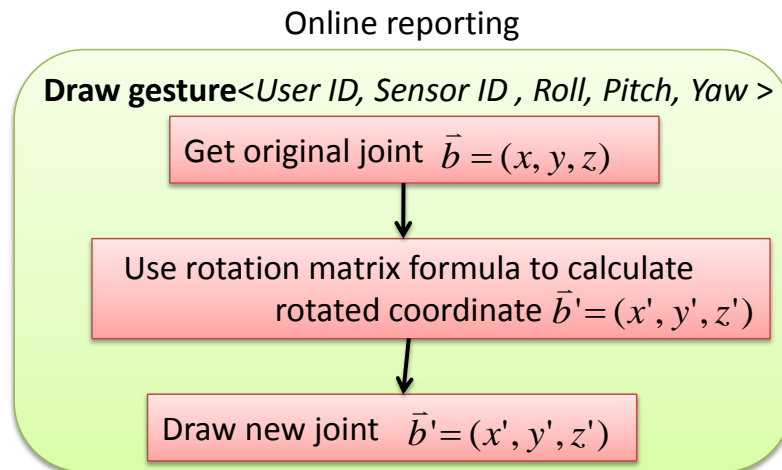


Figure 3.8: The Flow Chart of Rendering API.

Pattern Recognition API: This API provides the developer to calculate how many times the pattern of the user appears in period of time. First, the developer sets details of a pattern and then uses the method of pattern recognition to calculate how many times the pattern appears by the user's history sensing data. This API can be used: Steps recognition, repeat-motion recognition.

The definition of a pattern shows in Fig. 3.9, it is a walking pattern. The walking pattern is divided into four parts. The first motion is lifting up right thigh that is the roll angle of the node four to exceed -75 degree. The second motion is putting down right leg that is the roll angle of the node four to put under -75 degree. The third motion is lifting up left thigh that is the roll angle of the node six to exceed -75 degree. The fourth motion is putting down left leg that is the roll angle of the node four to put under -75 degree.

The usage of pattern recognition API shows in Fig. 3.10. On offline input phase, the developer should input the pattern name and the total of motions by using Set pattern API, and using Add states API to set states of the pattern, the sequence number of this state and the Euler angle of the sensor. On online reporting phase, the developer can use Get count API to calculate how

A walking pattern:

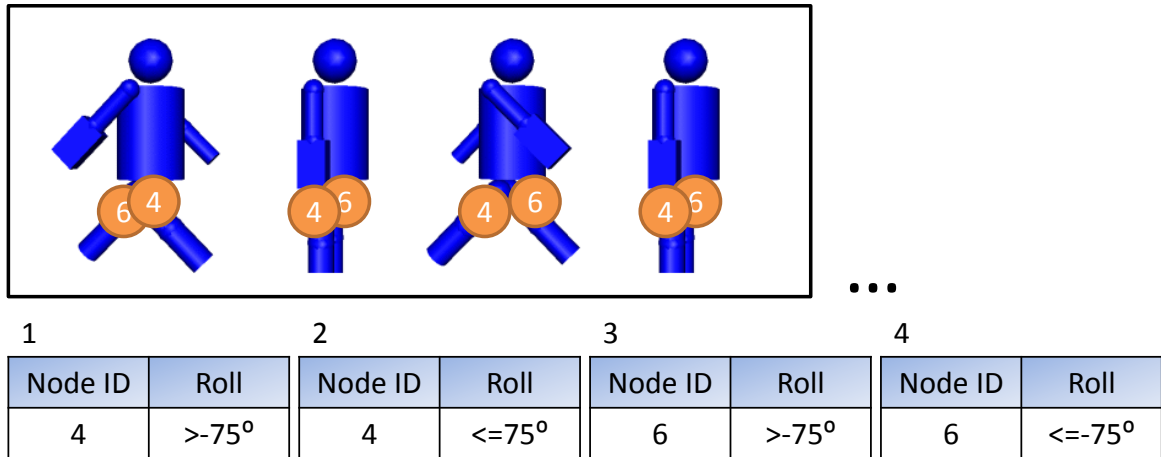


Figure 3.9: The Definition of A Pattern.

many times the pattern appears. Get count API will split the user's history sensing data into the each window. The each window contains a number of the user's sensing data, and Get count API will calculate how many times the pattern appears in the window, it matches the sensing data of the window by following the sequence of states. Each state must be the same sensing data more than once and then it enters the next state. The count of the pattern will add one if satisfies all states. The finally count of the pattern is calculated after all windows.

Set pattern < *Pattern name*, #*Sequence* >

Add states < *Pattern name*, *Sequence*, *Sensor ID*, *Roll*, *Pitch*, *Yaw* >

Count = *Get count* < *User ID*, *Pattern name* >

3.3 BSN Applications on Social Network Platforms

This subsection will introduce three applications by using BSN APIs, the BSN APIs usage and the flow chart of three applications. Three applications includes Posture messenger , Tai-Chi and Running game.

Posture Messenger: We can see real-time gesture and status of ours and friends in Posture messenger on Facebook. The gesture and the status is using user's sensing data. The traditional MSN only can manually set the self status. Posture messenger can real-time show the gesture and the status in Fig. 3.11.

Posture messenger uses two BSN APIs, the first one is Status reporting API that can identity

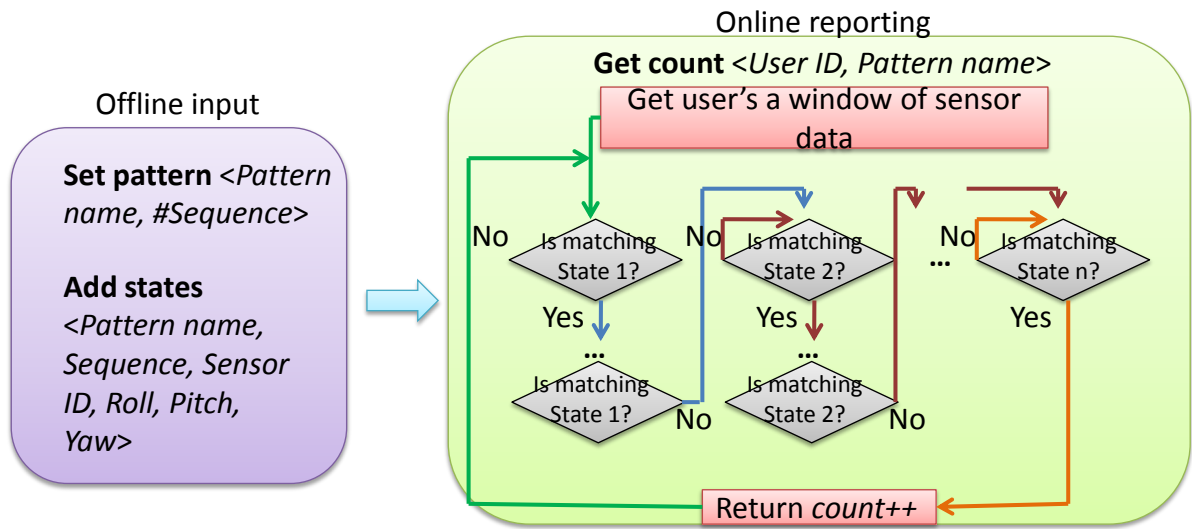


Figure 3.10: The Flow Chart of Pattern Recognition API.

how the status is. The second one is Rendering API that is using JAVA 3D to render the real-time gesture, we will introduce in the next application. When we first use Status reporting API, use Set status to set the status and details on offline input phase and then call Read status API to input the user's Facebook ID on online reporting phase. Read status API will get the current user's sensing data that matches with the rule of setting on offline input phase. If there have the match rule, then return the rule name. In Fig. 3.12, when Read status API gets a group of the user's sensing data, there only have sleeping of the rule matching, then return sleeping of the rule name.



Figure 3.11: The User Interface of Posture Messenger.

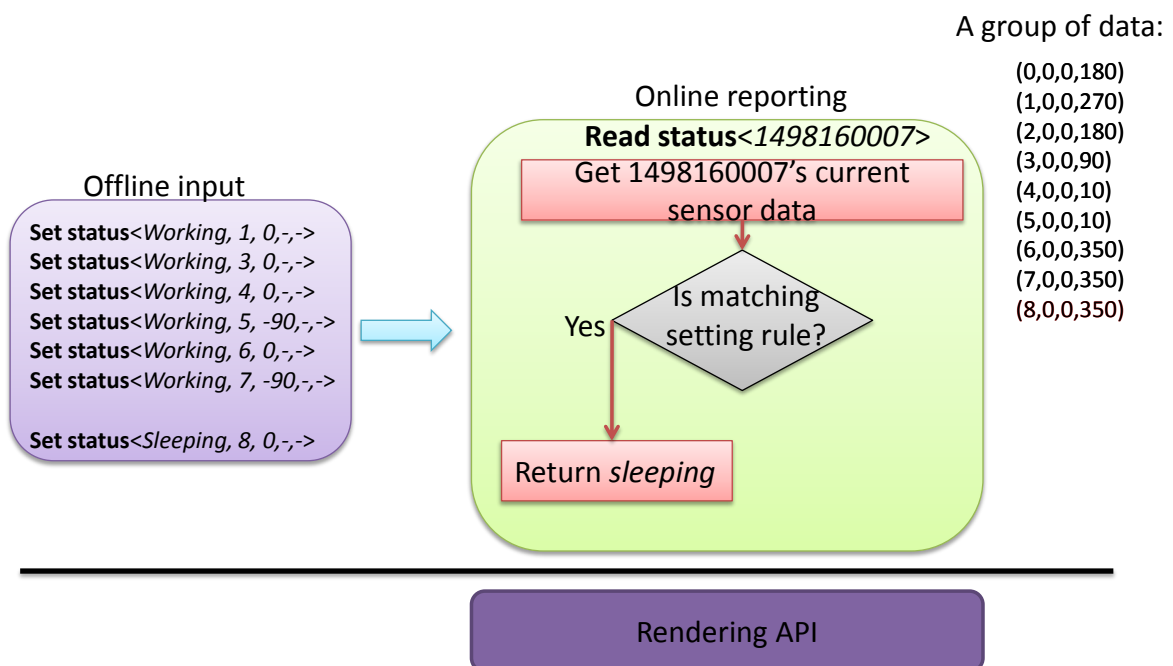


Figure 3.12: A Example of Static Reporting API.

Tai-Chi: The user and his/her friends can play Tai-Chi quan(Chinese traditional shadow boxing) together in Tai-Chi on Facebook. Although Tai-Chi teacher is in different location, Tai-chi students can learn the real-time gesture of Tai-Chi teacher on Facebook. Tai-Chi teacher can see the real-time gesture of Tai-Chi students to redress the gesture.

Tai-Chi uses Rendering API in Fig. 3.14. First, we use 3D rendering engine to render the user's 3D human model, then call Draw gesture API on online reporting phase, and input Euler angle of the user's sensing data. Draw gesture API will get the original coordinate of each joint such as arm's coordinate (10,10,10), and then it uses Rotation matrix to rotate Euler angle(-45,0,0) of the arm's sensor. The finally coordinate of calculated Euler angle is new coordinate (10,-3,13.7), update this coordinate to arm's 3D model.

Running Game: Running game provides the user and his/her friends running together on Facebook. Because the user and his/her friends maybe can not running together for the weather and the position. The user and his/her friends can real-time see the gesture and steps of each other.

Running game uses two BSN APIs that shows in Fig. 3.16 ,one of BSN APIs is Pattern recognition API, it can calculate steps of the user and his/her friends. Another one is rendering

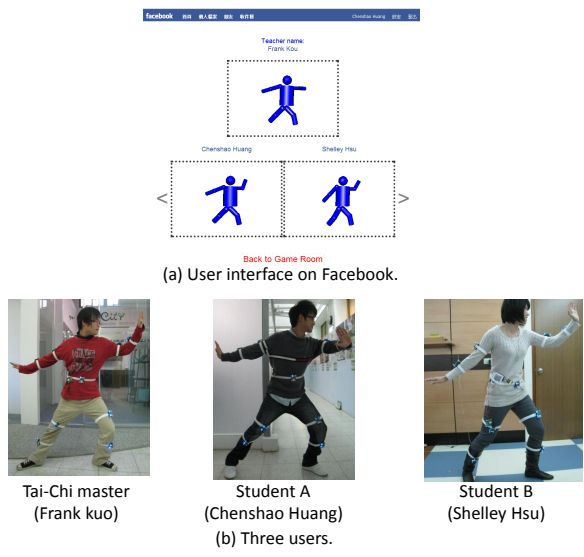


Figure 3.13: The User Interface of Tai-Chi.

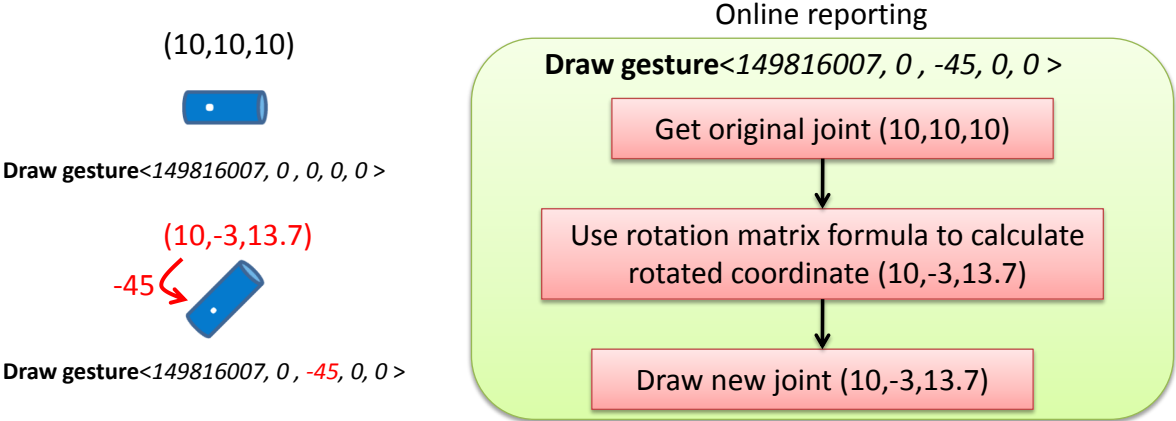


Figure 3.14: A Example of Rendering API.

API, it can render the running gesture of the user and his/her friends. First we set the running pattern on offline input phase, and we call Get count API to calculate how many times the pattern appears. Get count API should be input the user's ID and the pattern name what needs to be calculated, and then it will match the state of the pattern and the user's sensing data of the window. If all the states is matched, the step count will be added one. After it matches all windows, the step count is the user steps.

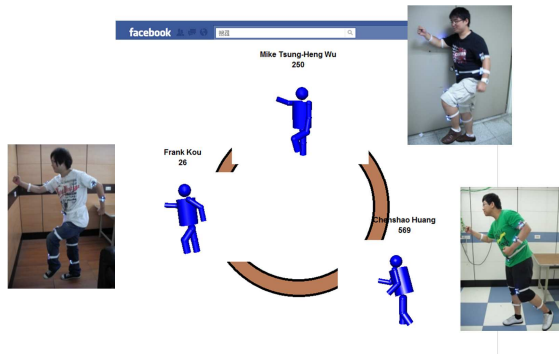


Figure 3.15: The User Interface of Running Game.

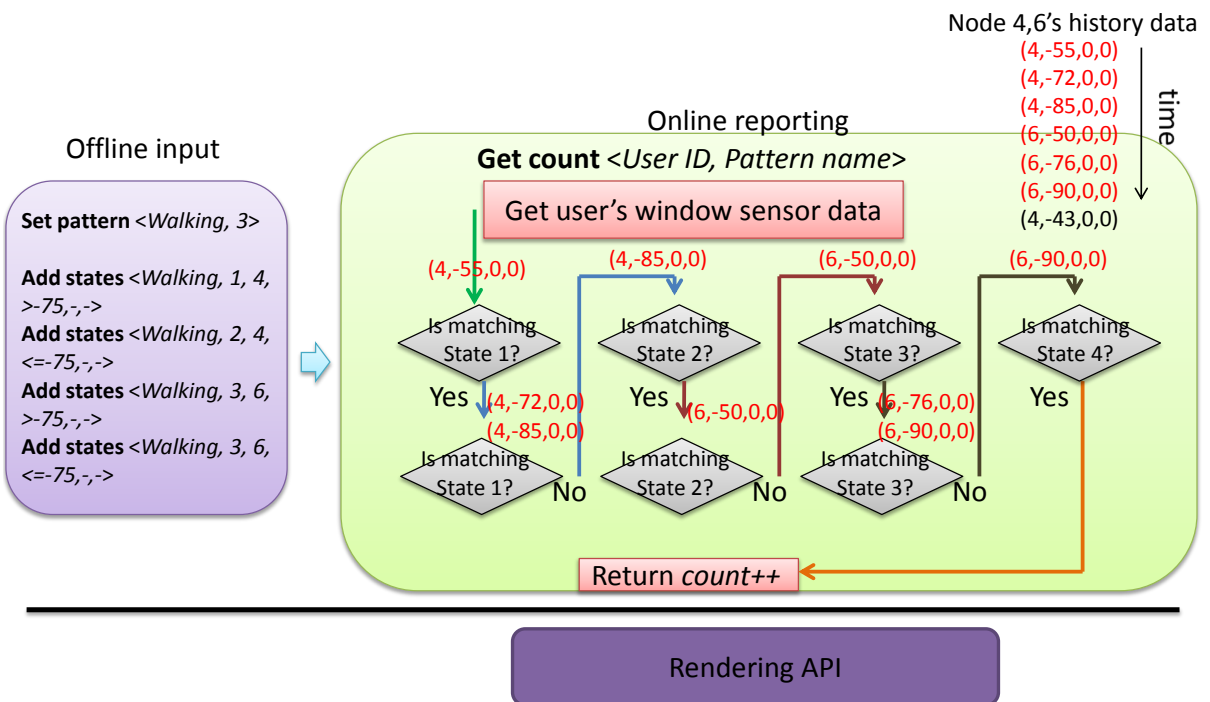


Figure 3.16: A Example of Pattern Recognition API.

Chapter 4

Data Collection Protocols

This section will introduce our data collection method. The definition of the data collection method is a process that the sink to collect the data of each body sensor. The goal of our data collection method is providing that the developer and the user do not need to adjust the data collection method, our data collection method collects the user's sensing data adjusting different applications.

We consider that for different applications, the collection importance of larger variance is large than the smaller variance in the user's sensing data. We observe that for different applications, different human body has different variance. In Fig. 4.1, we observe the variance of different human body from three our designed applications in Section 3.3, we divide into upper and lower human body for observation. We use standard deviation of the Euler angle as the variance for each sensor. We observe that the non-acute applications has the smaller variance in whole body. For example: The static motion has few variance in whole body. In some motions of Tai-Chi, user's upper body has the lager variance than lower body. In Running game, user's lower body has the lager variance than upper body. Therefore, we consider that for different applications must be collected more variance of the user's sensing data.

We hope that the more variance of the sensor has more number of data. Because the sensor has more number of data, applications have better resolution such as faster updates data ,high resolution rendering and high accurate pattern recognition. However, the total number of the data are fixed, because the transmission protocol of our body sensor is using IEEE 802.15.4 bandwidth. So the data collection will be lost when the data generation rate is more faster than the data transmission rate. We must to collect more important data to tolerate the data losing. The definition of more important data is that the sensor has more variance of Euler angle in a

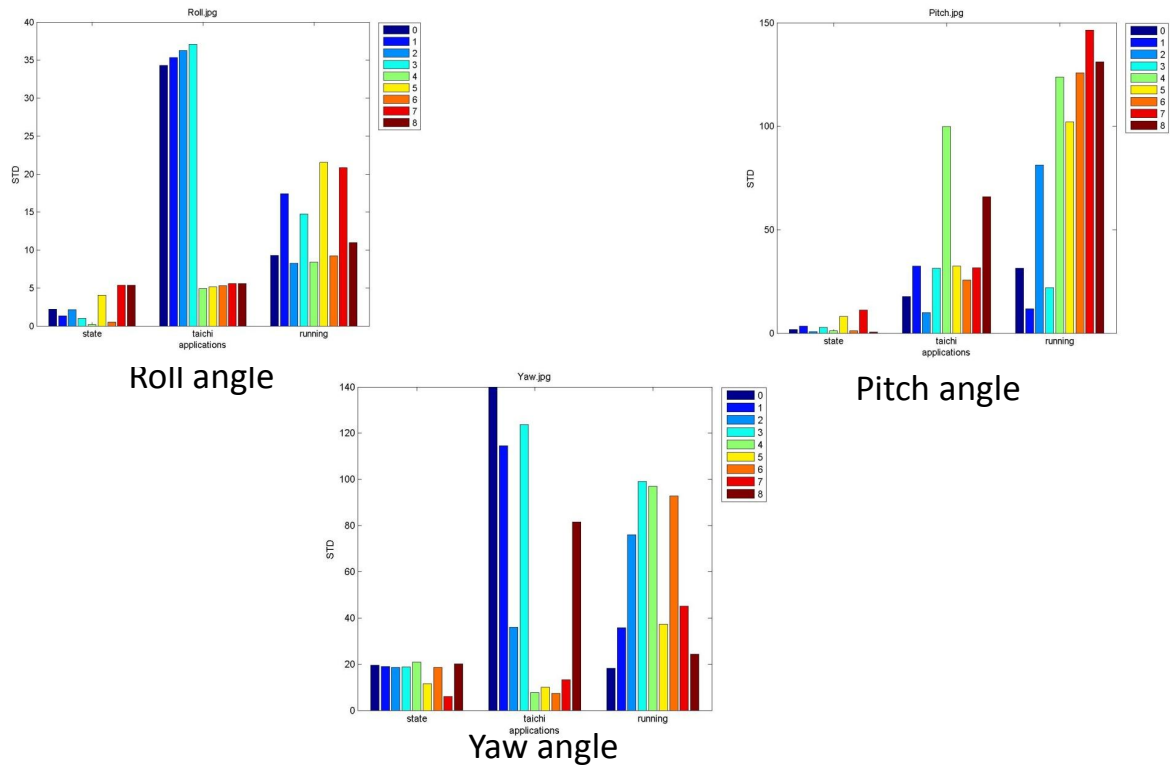


Figure 4.1: Different Variance for Different Human Body.

period.

The traditional methods of data collection protocol are contention and polling protocol. The number of collection data is not fixed by using contention protocol, the number of collection data of each sensor is same by using polling protocol. Above two methods of data collection protocol are not following important data to collect more number of data. We must to design a method of data collection to collect more important data, and we only know the variance of each sensor.

Our method of data collection is collecting the user's sensing data by sink, sink will collect the data flowing the variance of each sensor. For example: The more variance of sensor has more collection data. So, sink will allocate the sequence and number of collection data for each sensor. In Fig. 4.2, we divide the medium access time of sink into a Frame. We assume that the frame time is not fixed, a frame is composed by the number of Polling round. A polling round is a time that includes sink requesting time and sensor returning time. We assume that the data generation rate of each sensor is fixed $r = \text{samples}/\text{time}$, and the number of the data of a polling round is fixed $c = \text{samples}/\text{apollinground}$. Each frame must a dependency between

this frame and previous frame, this frame uses the information of previous frame.

A frame means a interaction of the data collection. we divide a interaction into three phase that includes sampling phase, bandwidth allocation phase and reporting control phase. Sampling phase is that we calculate the variance of each sensor. Bandwidth allocation phase is following the variance of each sensor to allocate the number and the position of the polling round for each sensor. Reporting rate control is that the sensor should be returned the data when each polling round needs to return.

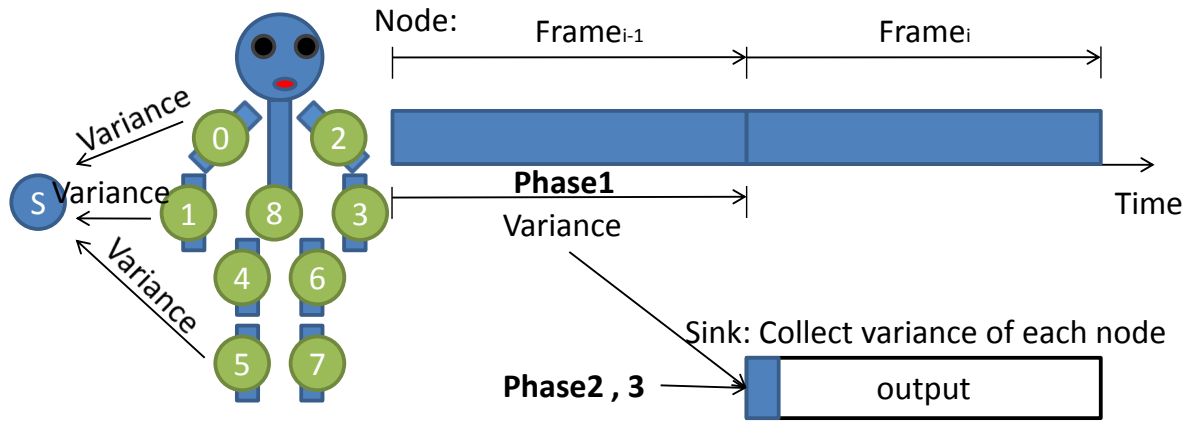


Figure 4.2: Our Method of Data Collection.

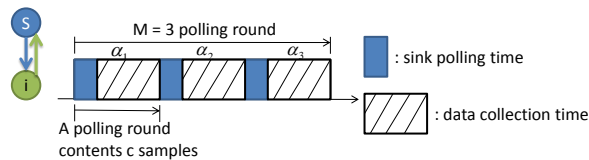


Figure 4.3: The Definition of A Polling Round.

Sampling Phase: The goal of this phase is calculating the variance of a frame about each sensor. Each sensor will sense data in every frame, the sensor will calculate the variance in a frame after sensing data. Our method of calculated variance is using standard deviation of Euler angle from each sensor. We assume that the sensor has same data format $(\varepsilon_1, \varepsilon_2, \dots, \varepsilon_k)$, and the variance of each data in a frame is $(D_i^1, D_i^2, \dots, D_i^k)$. The variance of each sensor is adding all variance of each data $\sigma_i = \sum_{j=1}^k W_j D_i^j$. W_j is a normalize factor, because the data format maybe is difference such as 0 360 degree and -2G 2G. Using this factor can be the same scale for each data.

Bandwidth Allocation Phase: This phase is allocating bandwidth for each sensor. We assume that the sink requests only one sensor in a polling round. A polling round is allocated only one sensor that shows in Fig. 4.4. 4 illustrates that a polling round allocation is a binary, zero means that j-th polling round is not allocated, one means that j-th polling round is allocated to sensor i.

$$x_j^i = \begin{cases} 0, & \text{no allocate} \\ 1, & \text{j-th polling round is allocated to sensor i for reporting, } j= 1 \sim M \end{cases}$$

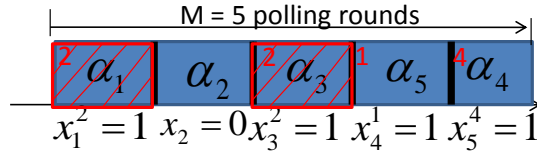


Figure 4.4: A Polling Round Is Allocated Only One Sensor.

We use a set A_i that means a sensor i has which polling rounds in a frame, that shows in 4. We calculate the number of a set A_i using the ratio about the variance of sensor i and total sensors. So, the more variance of the sensor will be allocated more polling rounds, the less variance of the sensor will be allocated less polling rounds.

$$A_i = \{ \alpha_j \mid \text{if } x_j^i = 1, j = 1 \sim M \}, \text{ where } |A_i| = \lceil \frac{\sigma_i}{\sum_{i=1}^n \sigma_i} \cdot M \rceil$$

We use T_i that means a interval from the sink requests sensor i to return the data, that shows in 4. The length of interval is a relation with the size of A_i , the more variance of the sensor has short interval, the less variance has long interval.

$$\text{Polling interval of sensor i: } T_i = \frac{\text{Number of total polling rounds}}{\text{Number of polling rounds of sensor i}} = \frac{M}{|A_i|}$$

Bandwidth Allocation Problem Definition: We want to bandwidth allocation, first we must to define the input and output. The input is our calculated variance of each sensor in a frame, and the output is the position of polling rounds of each sensor after bandwidth allocation.

Input: A frame has M polling round, and we already know the variance of each sensor $\sigma_1, \sigma_2, \dots, \sigma_n$.

Output: We use the variance from input to allocate the position of polling rounds for each sensor. Each polling round is collision free $\sum_{i=1}^n x_j^i = 1, j = 1 \sim M$ that means only to allocate a polling round to one sensor, and more variance of the sensor must to allocate more polling rounds $\sigma_i > \sigma_j \Rightarrow |A_i| > |A_j|$. We hope that the interval of the polling round is longer to distribute the polling round of each sensor, shows in Fig. 4.5. We hope that any two of near polling rounds is maximized, because that will let the transmission delay to shorter for each sensor. For example: If it allocates continuous polling rounds to sensor i, the application will require the data of the other sensor to wait a long time.

For each sensor i: $\sum_{i=1}^n \min(y_j^i - y_k^i)$ is maximized, where $y_j^i, y_k^i \in A_i$

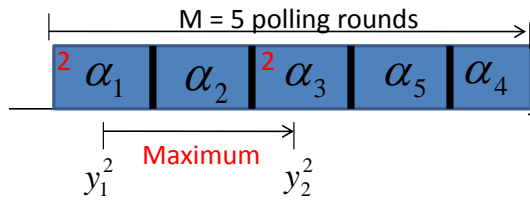


Figure 4.5: To Maximize the Polling Round of the Sensor.

Solution: Our method of bandwidth allocation is that the sink calculates number of polling rounds and polling interval for each sensor after each sensor calculates each variance. First, more number of polling rounds of the sensor will follow polling interval to allocate polling round. Because more number of polling rounds early allocates, the collision of allocation will be less. In Fig. 4.6, sensor 1 has more number of polling rounds, so the sink first allocates polling rounds of sensor 1 following its polling interval. To allocate polling round should begin the first polling round.

The sink next allocates polling rounds of sensor 2 following its polling interval. But it is occurred collision in the first polling round, the sink should shifts the polling round to no collision polling round. Sensor 3 has least polling rounds, and it is occurred collision, too. It should shift the polling round to no collision polling round.

Reporting Rate Control: The goal of reporting rate control is which data returns to the sink. The sampling rate of each sensor is fixed, therefore, each sensor are sampling the same way. In Fig. 4.7, each sensor knows the number of polling rounds in this frame. Each sensor will uniform sample in the previous frame. The number of sampling data is the capacity of this

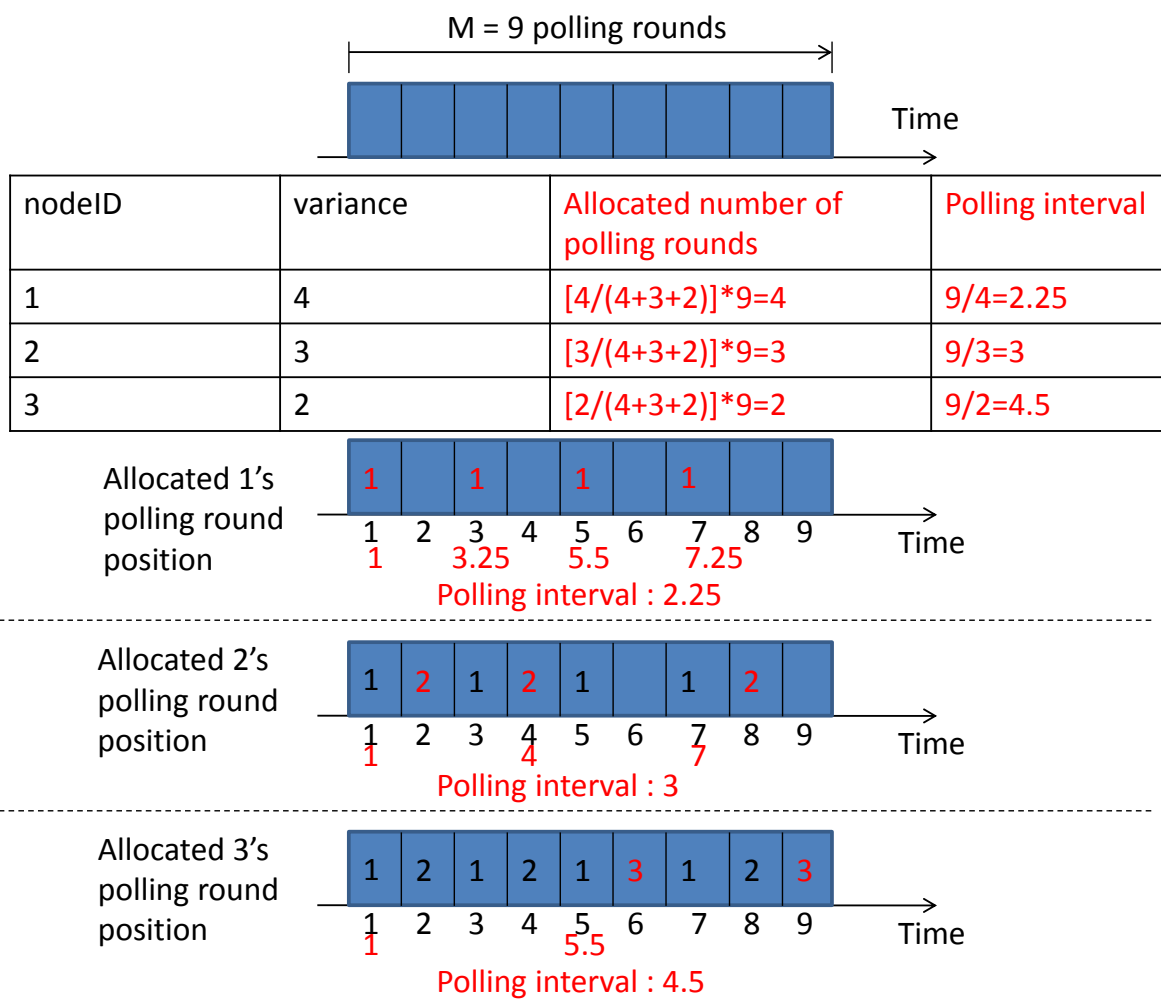


Figure 4.6: Allocate Polling Rounds of Sensor 1, 2 and 3.

frame. The signification of uniform sampling is that each sample has same importance in the same frame, because the variance of the frame is same. After uniform sampling, the sink will fill in the data following the sequence of polling rounds. For example : the number of allocation polling rounds is two, and the capacity of each polling round is two samples. Therefore, the sink should uniform sample four samples in the previous frame, and the sensor 2 fills in the data following the sequence of polling rounds.

Allocated polling rounds of node 2:

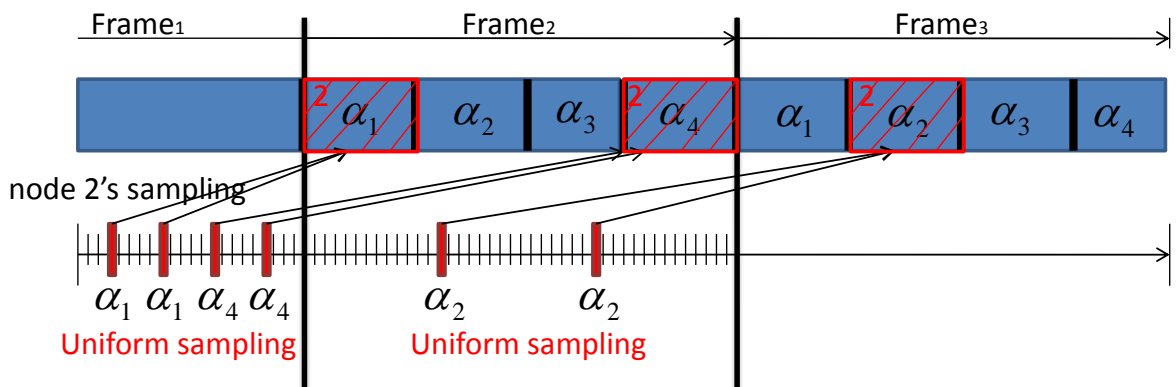


Figure 4.7: The Reporting Control Scheme.

Chapter 5

Implementation & Experimental Results

This section will introduce our implementation that includes social network server details, the body sensor and the sink designed method and wear method in Fig. 5.1. Experiment results will show the performance of our data collection protocol and experiment environment in Fig. 5.2.

5.1 Implementation

Our implementation device includes social network platform, servers, body sensors, the sink. We will illustrate in the following:

Social Network Platform: We are using Facebook as our Social network platform. Facebook currently has about 350 million of the population, and to growth 250000 of the population every days. Facebook provides Facebook API to developer to fast develop their application on Facebook.

Servers: Our servers includes web server, database server and BSN server. These servers mount on the same computer, and this computer OS is Linux. We are using Apache server as our web server, its usage is communicating with the user's browser and shows the user interface. Web server will transmit JAVA 3D to client that can show the human 3D model. We are using MySQL as our database server, its usage is providing that web server accesses the user's information in application and BSN server stores the current user's sensing data. We are using our designed UDP program as our BSN server, this program will write the current user's sensing time from the sink to database server.

The data flow of servers with three applications in Fig. 5.1. First, the BSN uses our data collection protocol to collect the user's sensing data to the sink, and then the sink transmits the data to BSN server via wireless/wired network such as 3G/Wifi/Ethernet. BSN updates the

received data to database server. Users login social network platform such as Facebook via the browser, and they can execute our designed three application that includes Posture messenger, Tai-Chi and Running game showing in Fig. 5.2. After selecting a application, social network platform directs to our web server, and web sever will load the application to the client's browser and the application accesses web server and database server with different user.

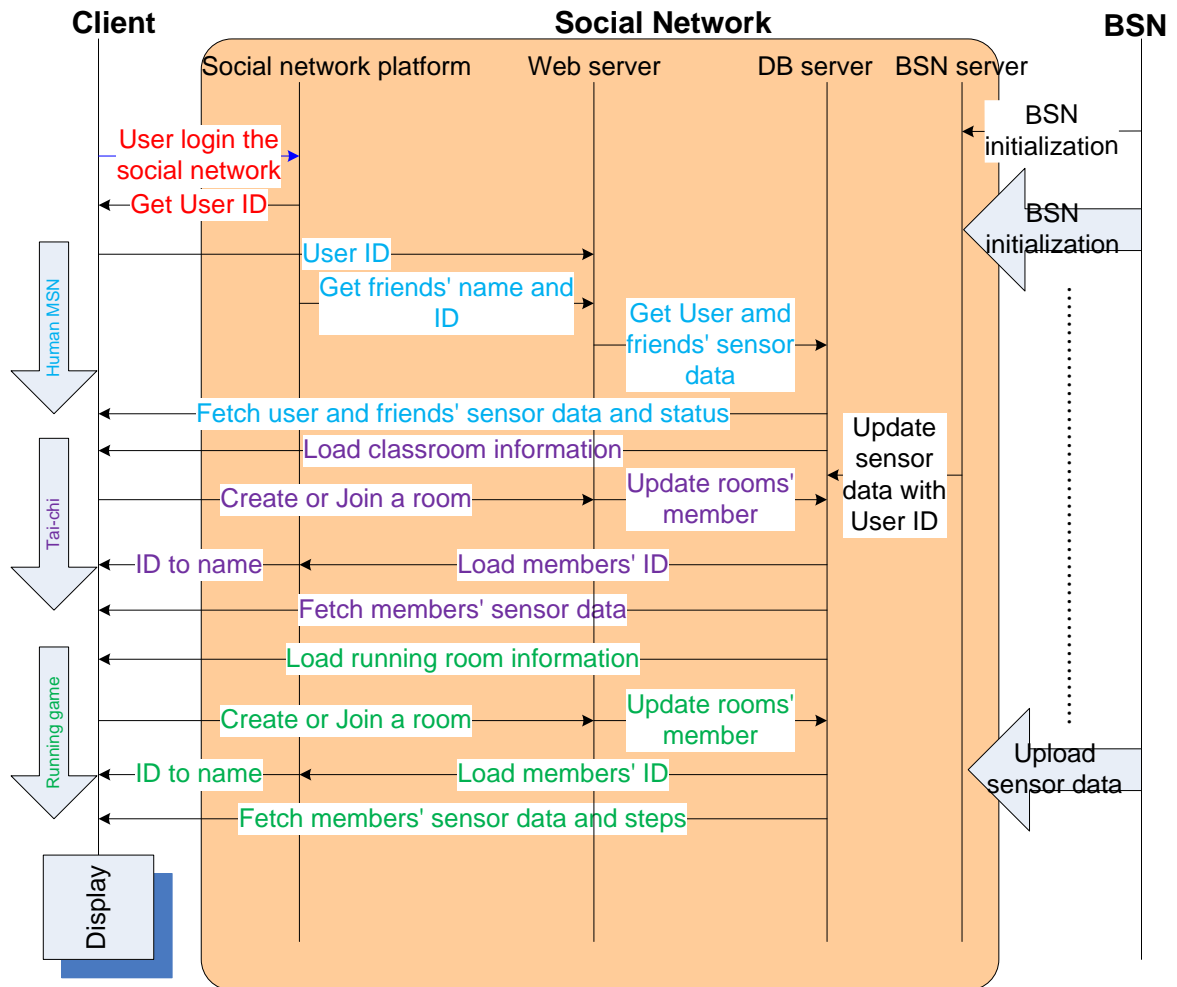


Figure 5.1: The Data Flow of Servers with Three Applications.

Body Sensors: Our body sensors can measure the user's Euler angle. We compose the body sensor to use modules that includes Jennic, G-sensor and a battery showing in Fig. 5.3. Jennic module is Jennic 5139, this module has the wireless capacity of IEEE 802.15.4 and a micro controller to control peripheral devices. G-sensor module is OS-5000, this module can sense the user's three acceleration, compass and Euler angle. The battery is using the 1000mA cell phone battery. The characteristic of our body sensor is light-weighted, well-portable and long



Figure 5.2: The User Interface of Selecting Three Application.

power. Executing our data collection protocol can use about two hours.

The step of measuring the user's sensing data is, Jennic first reads reading of OS-5000 via UART, and Jennic transmits the data to sink via wireless when sink requests.

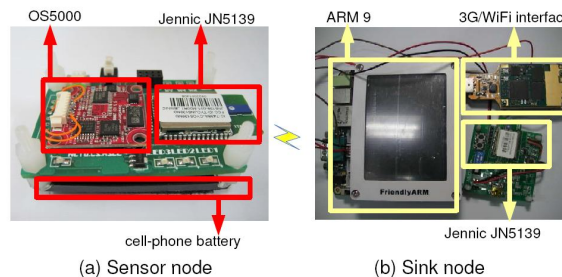


Figure 5.3: The Hardware of the Sensor and the Sink.

Our wear method of body sensors is showing in Fig. 5.4. we tie nine body sensors in user's body, each sensor needs to tie in the middle of the joint and the waist, and each sensor is downward direction when the user stands. The sink needs to tie in the waist or to hold in hand.

Sink: The function of the sink is uploading the user's sensing data to BSN server via internet after it collects the data. Our sink is using ARM9, its computing is equivalent to a smart phone,

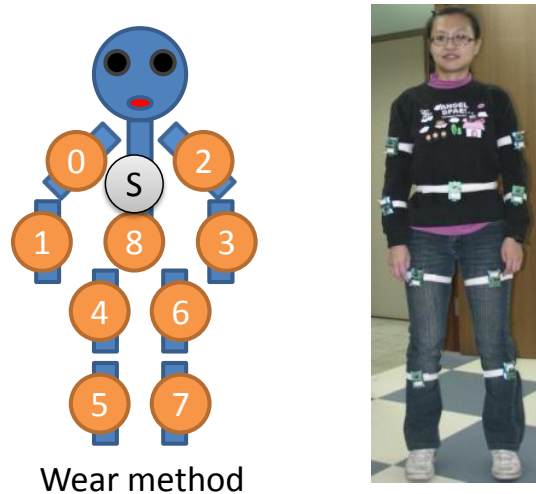


Figure 5.4: The Wear Method of Nine Body Sensors and A Sink.

and it can plug-in the USB device. The sink connects Jennic via comport, and the Jennic can execute our data collection protocol to collect the user's sensing data in body sensors via wireless. ARM9 can plug-in 3G/Wifi network card, and ARM9 has the internet capacity that can transmit the data to BSN server.

5.2 Experiment Results

We show the performance our data collection protocol in this section. We design our data collection protocol in Chapter 4, this protocol automatic adjusts the number of data for different applications. We compare the number of sample and throughput that includes our data collection protocol and two traditional protocols.

The explanation of two traditional protocols as follows:

Polling Protocol: The method of data collection is that the sink polls the sensor following the sensor ID and the sensor returns the setting number of the user's sensing data.

Pure Contention Protocol: The method of data collection is that the sensor immediately returns the data to the sink when the sensor senses a certain number of the user's sensing data. Because there will be collision, we use IEEE802.15.4 CSMA/CA to avoid collision and to retransmit.

Our motions of experiment includes three motions, static is testing the motion of the human all body, Tai-Chi is testing the motion of the human upper body and running is testing the

motion of the human lower body. Our environment is that the user wears nine sensors and a sink showing Chapter 5.1. Each sensor has same sampling rate 20 samples/sec. The data recording time is 30 sec/once. The setting of our data collection protocol is that a frame has fifty polling rounds, and the capacity of each polling round is one sample. The setting of polling protocol is that the capacity of each node is one sample. The setting of pure contention protocol is that the sensor immediately returns the data when it senses one sample.

Experimental Environment	
Motions	Static, Tai-chi, Running
Total sensors	9 sensors
Log time	30 sec
Sampling rate	20 samples/sec
Our Data Collection Protocol Setting	
Number of polling round per frame	50 polling rounds/frame
Capacity of a polling round	1 sample
Polling Protocol Setting	
Capacity of a polling round	1 sample
Pure Contention Protocol Setting	
Capacity of a polling round	1 sample

The performance of our data collection protocol shows in Fig. 5.5. It shows that running has the more variance in lower body, so lower body has the more number of data. Tai-Chi has more variance in upper body, so upper has the more number of data. Static has the less variance in all body, so the number of data is not obvious. It proves that our protocol follows the variance of sensors to collect the user's sensing data. In Fig. 5.6 and Fig. 5.7, polling protocol shows that the number of data of each sensor is the average, and contention protocol is contrary. The traditional protocols can not follow the variance of sensors to collect the user's sensing data.

We find that if the sink allocates more polling rounds to the sensor, the sink will collect more the number of data showing in Fig. 5.8. It proves that the number of data is changed by our allocating polling round.

We compare the throughput of our protocol and the traditional protocol showing in Fig. 5.9. We find that our throughput is lower than pure contention protocol, the reason is that our transmission method is the sink to collect each sensor, there should waste collection and transmission time. And our throughput is lower than protocol, the reason is that our protocol should use each sensor to calculate the variance, because our hardware computing of sensors is slow, there should waste computing time.

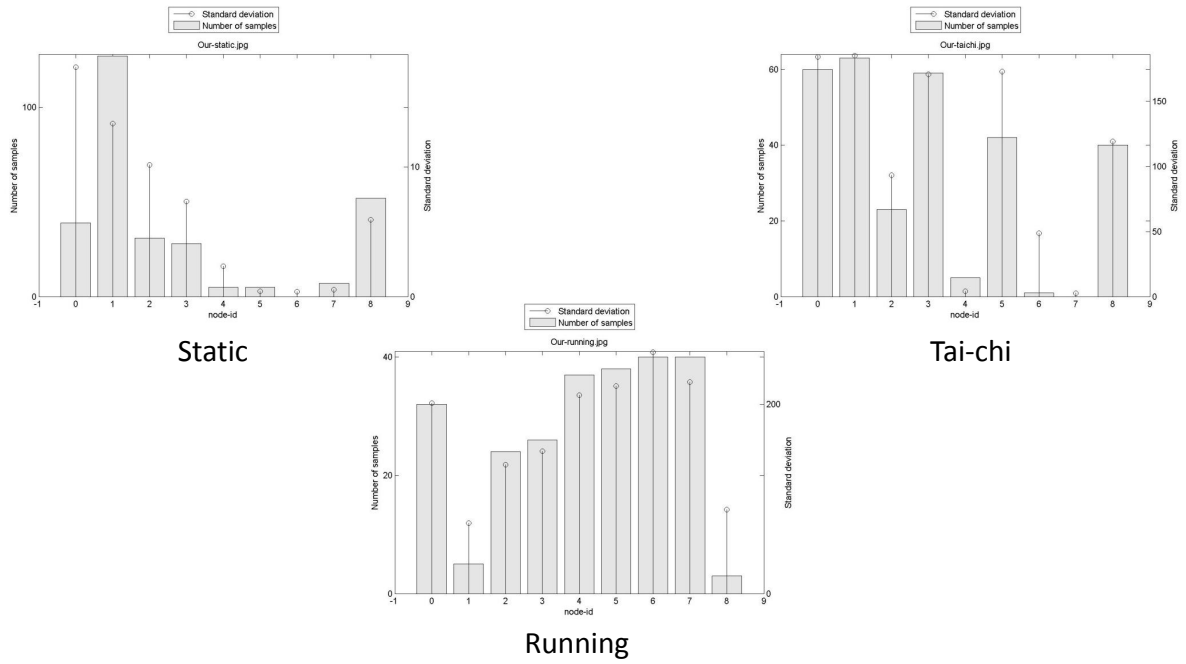


Figure 5.5: Our Data Collection Protocol Performance (Number of Samples vs. Variance).

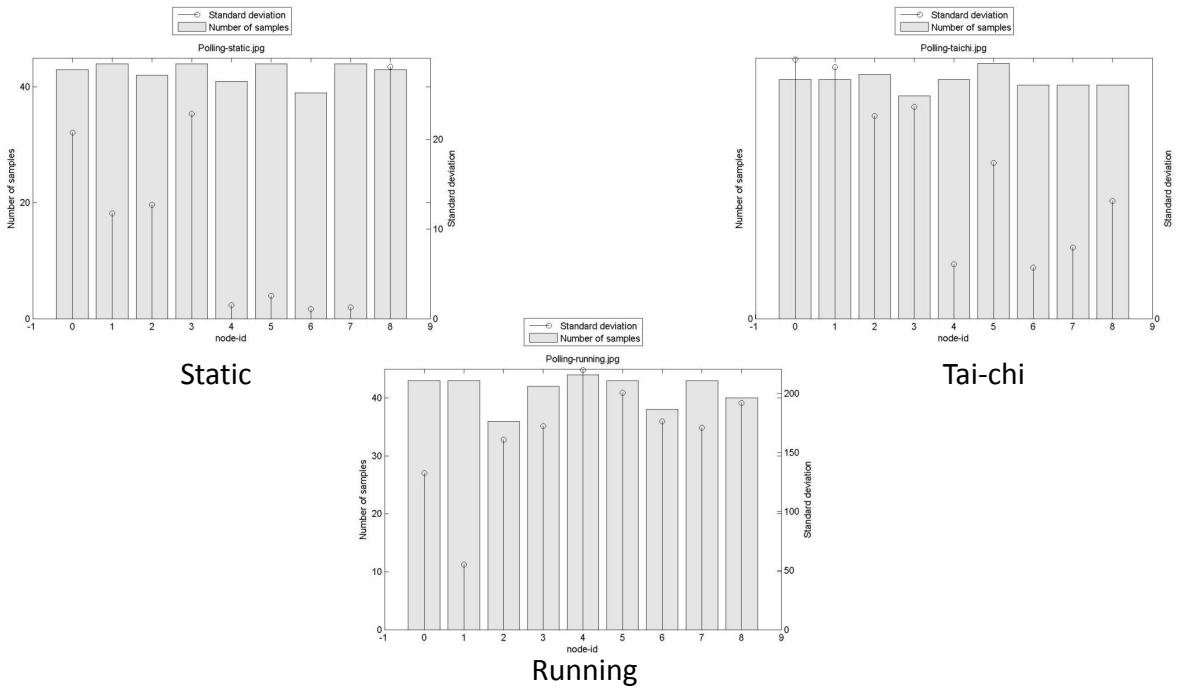


Figure 5.6: Polling Protocol Performance (Number of Samples vs. Variance).

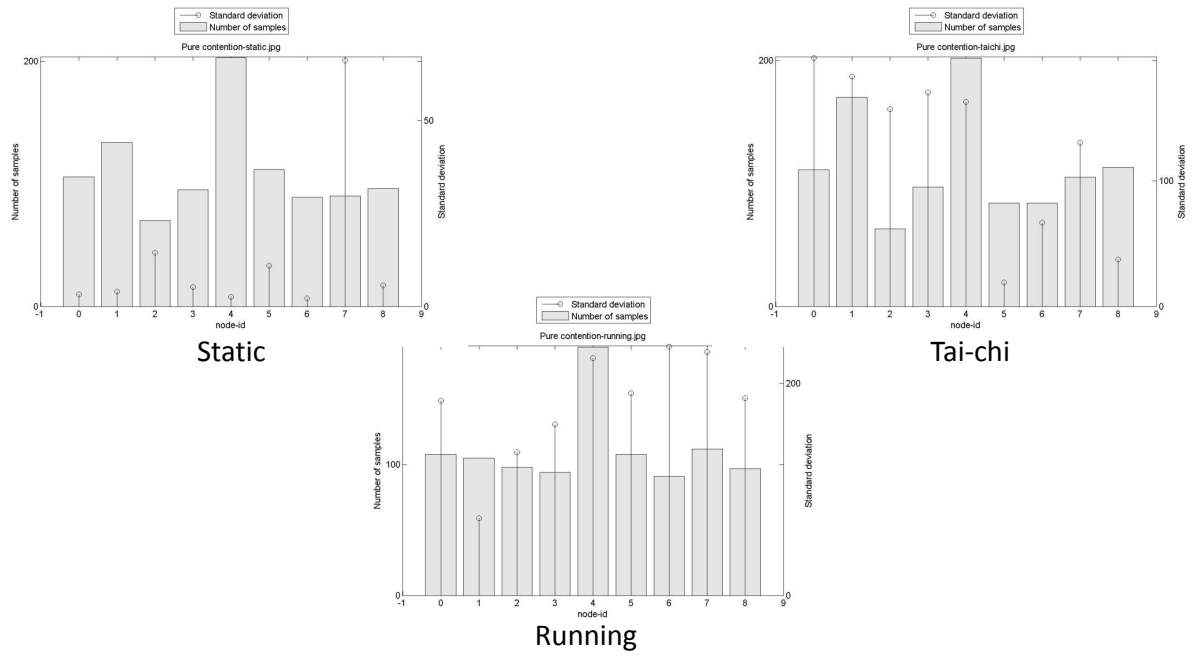


Figure 5.7: Pure Contention Protocol Performance (Number of Samples vs. Variance).

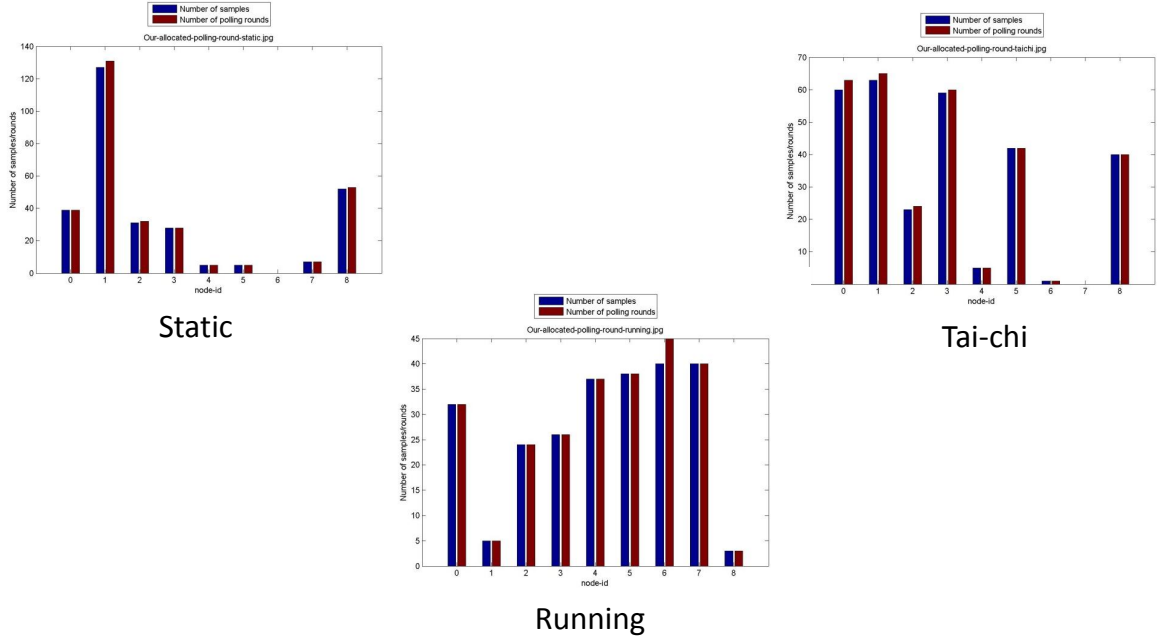


Figure 5.8: Polling Rounds vs. Number of Samples.

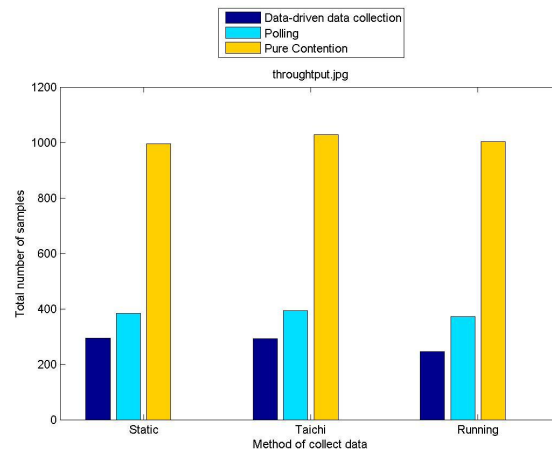


Figure 5.9: Compare Throughput.

Chapter 6

Conclusions & Future Works

We propose an innovative virtual-physical social network platform. This platform provides three of our designed APIs that include: Status reporting API, Rendering API and Pattern recognition API. The developer can use these APIs to quickly develop and extend the application on social network with BSN. We use these APIs to implement three applications that include: Posture messenger, Tai-Chi and Running game. The social network user can easily use these applications and automatically share their own sensing data to friends on social networks. We design our data collection protocol to collect the more important user's sensing data. Our protocol can follow the variance of each sensor to collect the suitable number of data, and it does not know that the user executes what applications. Experimental results show that our protocol can efficiently collect the more variance sensing data of the sensor.

In future works, we will extend BSN APIs and add more number of body sensors to increase the resolution for rendering. And add the kind of body sensors such as biomedical sensors, there will be more applications in this platform. We can improve our hardware of sensors to make better the throughput of our data collection protocol.

Bibliography

- [1] C. R. Baker, K. Armijo, S. Belka, M. Benhabib, V. Bhargava, N. Burkhart, A. D. Minassians, G. Dervisoglu, L. Gutnik, M. B. Haick, C. Ho, M. Koplow, J. Mangold, S. Robinson, M. Rosa, M. Schwartz, C. Sims, H. Stoffregen, A. Waterbury, E. S. Leland, T. Pering, and P. K. Wright. Wireless sensor networks for home health care. In *AINAW*, 2007.
- [2] H. Cao, S. González-Valenzuela, and V. C. M. Leung. Employing ieee 802.15.4 for quality of service provisioning in wireless body area sensor networks. In *Proc. IEEE advanced information networking and application*, 2010.
- [3] D. Curtis, E. Shih, J. Waterman, J. Gutttag, J. Bailey, T. Stair, R. A. Greenes, and L. Ohno-Machado. Physiological signal monitoring in the waiting areas of an emergency room. In *BodyNets*, 2008.
- [4] Facebook. Facebook. <http://www.facebook.com/>.
- [5] E. Farella, A. Pieracci, L. Benini, L. Rocchi, and A. Acquaviva. Interfacing human and computer with wireless body area sensor networks: the WiMoCA solution. In *Multimedia Tools and Applications*, 2008.
- [6] T. Gao, T. Massey, L. Selavo, D. Crawford, B. rong Chen, K. Lorincz, V. Shnayder, L. Hauenstein, F. Dabiri, J. Jeng, A. Chanmugam, D. White, M. Sarrafzadeh, and M. Welsh. The advanced health and disaster aid network: a light-weight wireless medical system for triage. In *IEEE Transactions on Biomedical Circuits and Systems*, 2007.
- [7] F. Hu, Y. Xiao, and Q. Hao. Congestion-aware, loss-resilient bio-monitoring sensor networking for mobile health applications. *IEEE Journal on Selected Areas in Communications*, 27(4):450–465, 2009.
- [8] B.-W. Hwang, S. Kim, and S.-W. Lee. A full-body gesture database for automatic gesture recognition. pages 243–248, 2006.
- [9] S. Jiang, Y. Cao, S. Iyengar, P. Kuryloski, R. Jafari, Y. Xue, R. Bajcsy, and S. Wicker. Carenet: an integrated wireless sensor networking environment for remote healthcare. In *Proc. of international conference on body area network*, 2008.
- [10] B. Latré, B. Braemt, I. Moerman, C. Blondiat, E. Reusens, W. Joseph, and P. Demeester. A low-delay protocol for multihop wireless body area networks. In *Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous)*, 2007.
- [11] H. Li and J. Tan. Heartbeat driven medium access control for body sensor networks. In *HealthNet*, 2007.
- [12] J. Liu, Z. Wang, J. Wickramasuriya, V. Vasudevan, and L. Zhong. uwave: Accelerometer-based personalized gesture recognition and its applications. In *Proc. of IEEE Int'l Conference on Pervasive Computing and Communications (PerCom)*, 2009.
- [13] K. Lorincz, B. rong Chen, G. W. Challen, A. R. Chowdhury, S. Patel, P. Bonato, and M. Welsh. Mercury: A wearable sensor network platform for high-fidelity motion analysis. In *Proc. of ACM Int'l Conference on Embedded Networked Sensor Systems (SenSys)*, 2009.
- [14] Microsoft. Windows Live. <http://home.spaces.live.com/>.
- [15] Microsoft. Windows live messenger. <http://www.msn.com/>.

- [16] A. Nazir, S. Raza, and C.-N. Chuah. Unveiling facebook: A measurement study of social network based applications. In *IMC*, 2009.
- [17] V. Rajendran, K. Obraczka, and J. Garcia-Luna-Aceves. Energy-efficient, collision-free medium access control for wireless sensor networks. In *Proc. of ACM Int'l Conference on Embedded Networked Sensor Systems (SenSys)*, 2003.
- [18] TinyOS. Tynyos for wireless embedded sensor networks. <http://www.tinyos.net>.
- [19] T. van Dam and K. Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. In *Proc. of ACM Int'l Conference on Embedded Networked Sensor Systems (SenSys)*, 2003.
- [20] D. Vlastic, R. Adelsberger, G. Vannucci, J. Barnwell, M. Gross, W. Matusik, and J. Popović. Practical motion capture in everyday surroundings. 26(3):35, 2007.
- [21] D. Vlastic, R. Adelsberger, G. Vannucci, J. Barnwell, M. Gross, W. Matusik, and J. Popović. Practical motion capture in everyday surroundings. In *ACM Transactions on Graphics (TOG)*, 2007.
- [22] C.-H. Wu, Y.-T. Chang, and Y.-C. Tseng. Multi-screen cyber-physical video game: An integration with body-area inertial sensor networks. In *Proc. of IEEE Int'l Conference on Pervasive Computing and Communications (PerCom)*, pages 832–834, 2010.
- [23] W. Ye, J. Heidemann, and D. Estrin. Medium access control with coordinated, adaptive sleeping for wireless sensor networks. In *IEEE/ACM Transactions on Networking*, 2004.
- [24] T. Yokoyama, S. Kashihara, T. Okuda, Y. Kadobayashi, and S. Yamaguchi. A generic api for retrieving human-oriented information from social network services. In *SAINTW*, 2007.
- [25] A. D. Young, M. J. Ling, and D. K. Arvind. Distributed estimation of linear acceleration for improved accuracy in wireless inertial motion capture. In *Proc. of Int'l Symposium on Information Processing in Sensor Networks (IPSN)*, pages 256–267, 2010.
- [26] G. Zhou, J. Lu, C.-Y. Wan, M. D. Yarvis, and J. A. Stankovic. BodyQoS: Adaptive and radio-agnostic QoS for body sensor networks. In *Proc. of IEEE INFOCOM*, 2008.