# Informationally Structured Space for Daily Life Monitoring

TOKYO METROPOLITAN UNIVERSITY

首都大学東京

Dalai Tang

Graduate School of System Design

Tokyo Metropolitan University

A thesis submitted for the degree of

*Doctor of Engineering*

March 2016

# Acknowledgements

I would like to present this thesis to my parents. Their struggle to raise and support me took me until this point. Furthermore, many thanks to my two older sisters and their families for their supports and spirit during my life in Japan.

I would like to extend my special thanks to my advisor, Prof. Naoyuki Kubota for his kind and strong supports, a lot of inspiring advice, and wonderful guidance. I could learn and experience a lot of things from him.

I would like to thank to Prof. Janos Botzheim, his lots of advice and supports gave me the opportunity to learn much from him.

I would like to thank to Dr. Takenori Obo and Dr. Takahiro Takeda for their advice and support during conducting researches and projects.

I would also like to thank to Prof. Honghai Liu, Dr. Hui Yu, Dr. Zhaojie Ju for their advice and hospitality during my stay in the UK.

Thanks to all the members of the laboratory for their support and advice. Ms. Tomoko Suzuki and Ms. Mizuho Shimada kindly supported my daily life in the laboratory; Mr. Woo Jinseok and Mr. Bakhtiar Yusuf for their warm supports and advice during my study in TMU; Mr. Yuichiro Toda and the sensor network team members, Mr. Noel Tay, Ms. Yuri Yoshihara, and Mr. Zheng Wei always supported me in programming and in technical issues; Mr. Mutsumi Iwasa, Mr. Shuo Nan, Ms. Noriko Takase, and Mr. Wei Quan for their support; Mr. Jun Shimazaki, Mr. Shogo Yoshida, Mr. Yusei Matsuo, and Mr. Junya Kusaka, Mr. Guo Wei long, and Ms. Saika Ono for their friendly support; Mr. Azhar Aulia, Mr. Shunsuke Miki, Mr. Takuya Mabuchi, Mr. Shin Miyake, Mr. Koji Tamatsu, Mr. Taichi Sato, Mr. Joji Sato, Ms. Sun Suqi, Ms. Chiaki Kasuya, and Mr. Shao for their daily support in the laboratory.

# Abstract

As the problem of the increased number of elderly people and the decreased number of children in Japan has arisen recently, the development of robot partner and intelligent room for monitoring and measurement system has become a main topic. In the research and development on robot partner, hardware technologies have been developed for human behavior passive measurement, and various methodologies have been proposed for human behavior active measurement through communication with human. In the research on intelligent room, methodologies for passive measurement by embedded visible sensors in the environment have been discussed in various research areas such as ubiquitous computing, ambient intelligence, Internet of Things (IoT), etc. However, the main issues of these studies are the construction of uninterrupted network, the specification of communication protocol, the distributed sensing, and the large-scale data measurement and collection, and so on. Nevertheless, methodologies and systematization are necessary to represent and use the information measured by individual robot and sensor.

Nowadays, various researches and developments on intelligent space and intelligent environments have been done, but the comprehensive design guidelines of information structuring and on the information sharing method between the intelligent room and the robot partner have not been discussed sufficiently. Furthermore, it is not systematically discussed how the system can flexibly adapt to changing environmental condition and system configuration. Therefore, in this thesis, I clarify the properties of a space where the information is structuralized for solving the abovementioned problems. Furthermore, I propose a methodology for information structuring by feature extraction based on bottom-up information collection, and information measurement based on top-down constraints. In such a space, the human behavior information is associated with location information in a certain environment, and we can access the temporal and spatial change of human behavior from the space. This space

is called informationally structured space. I also propose a human behavior measurement method, and a method for cooperation between the robot partner and the distributed sensing system composed of various sensors. Furthermore, I propose a method for the system to flexibly adapt to changing environmental condition and system configuration, and show the effectiveness of the proposed methodology through several experiments on human daily measurement and long-term monitoring. The thesis consists of six chapters.

Chapter 1 discusses the background and related researches. The research purposes and goals are also clearly explained in this chapter.

Chapter 2 introduces the basic properties of informationally structured space including information sharing among devices; information representation method by considering the familiarity with human; reversibility of informational conversion among devices; and information operation method by considering system versatility. Next, in order to clarify the design guidelines, I discuss the informationally structured space defined by three layers: the sensing layer, the feature extraction layer, and the monitoring layer; and I define the function and structure of each layer. The sensing layer is used for measurement; the feature extraction layer is used for extracting feature values from measurement data, and the monitoring layer is used for extracting temporal and spatial changes from feature values.

Chapter 3 presents informationally structured space for distributed sensing system. First, I explain the problem of outdoor and indoor measurement, global and local measurement in the sensing layer from the viewpoints of environmental conditions and measurement targets. Next, I propose human behavior measurement by applying a spiking neural network, and structuring and updating the informationally structured space based on the relationship between location and human behavior. The experimental results show that the proposed method is able to measure the human behavior in both indoor and outdoor flexibly.

Chapter 4 explains informationally structured space for robot partner to enable the active measurement of human behavior. First, I develop a gesture recognition system using evolutionary robot vision and conversation system based on time-dependent utterance, and propose human behavior measurement in the feature extraction layer. Next, I propose a method of complementarily using behavior information measured by the distributed sensing system and behavior information estimated by the robot partner. Experimental results through time dependent conversation show that the robot partner can estimate human behavior that is difficult to be measured by the distributed sensing system.

Chapter 5 discusses informationally structured space for the monitoring layer in order to realize long-term human behavior measurement. First, I propose daily life model estimation method using fuzzy modeling from long-term human behavior information. Moreover, I construct daily life models with different granularities such as day, week and so on. I also propose a method that specifies the difference between life models. Experimental results show, that the proposed method can specify the change of daily life models. In order to realize flexible adaptation to the change of environmental condition and system configuration, informationally structured space can flexibly change the way of signal processing and the network structure in the sensing layer according to the access state of the devices, in the case when a device is broken or the battery is empty. Experimental results show how the informationally structured space can flexibly handle such situations. Additionally, I develop a simulator based on the informationally structured space designed for a physical environment, and I use this simulator to perform numerical experiments in order to show the effectiveness of the proposed method.

Chapter 6 concludes the thesis and explains the future research directions. The thesis discusses the methodology for constructing informationally structured space from different points of view in order to show the efficiency of the proposed human behavior monitoring method.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

In this chapter, first the background and related researches are introduced. Then, the problems and challenges are discussed. Next, the overview of approaches and contributions are presented. This is followed by the explanation of the research purposes and goals. Finally, the outline of the thesis is presented.

## 1.1 Background

From the year of 2000 to 2050, the proportion of elderly people (60 years old or older people) in the world's population will double from 11% to 22%, and the absolute number of elderly people is expected to increase from 605 million to 2 billion over the same period[1]. Meanwhile, in Japan, according to the Statistics Bureau at the Ministry of Internal Affairs and Communication[2], the population of elderly people is expected to increase to 36 million that is about 31% of the population in the year of 2030. In Tokyo, the number of elderly people will reach 25.2% of the population in year 2015.

Along with the increasing number of elderly people, one must note that the number of those elderly people who are no longer able to look after themselves will also increase proportionally. For instance, many of them will lose the ability to live independently because of limited mobility, frailty or other physical or mental health problems [19, 99]. In Japan, the increasing number of elderly people who live alone or independently has required a large forms of nursing care to support them. However, since the number of caregivers is always limited, it is important to introduce another solution to tackle this problem. One of the solutions is the introduction of the human-friendly robot partner and intelligent room to support the elderly people in their daily life.

[1]See www.who.int/ageing/en/

[2]See www.stat.go.jp/english/data/handbook/c0117.htm

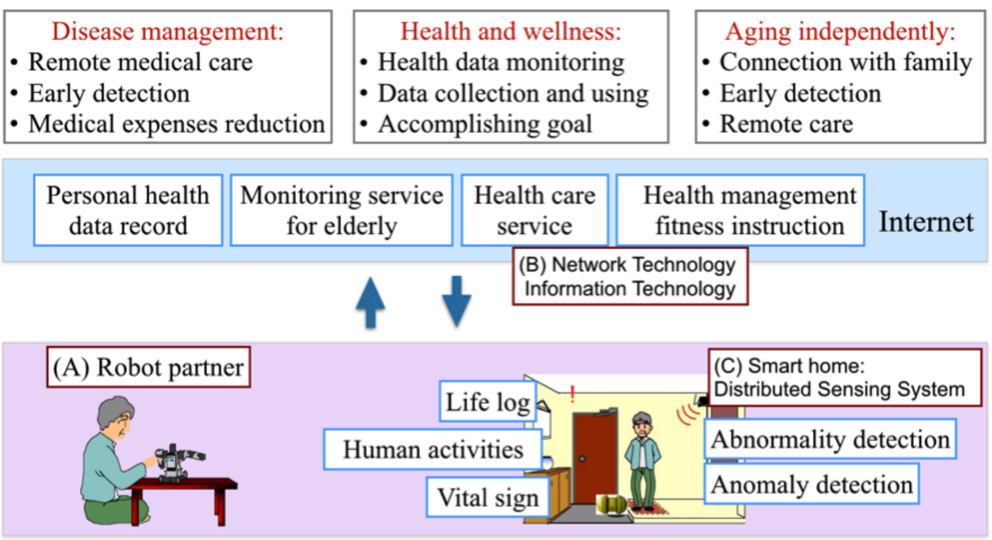



Figure 1.1: Health care monitoring

In the recent years, health care support system [2, 4, 92] based on application mainly for disease management, health and wellness management, and aging independently has developed rapidly. This condition is also affected by the result of cooperation between companies on the integration of the digitalization of health and medical equipment. Now, we can acquire various information, starting from low dimensional measurement sensor data until high dimensional data to produce life log data. However, by recent technology, it is shown that we only can acquire data, but it is also not less important that the foundation system which connects the measurement system until support system foundation is also needed. In order to realize this, the fusion between robot technology, smart home, information technology and network technology is very important.

## 1.2 Literature Review

In the literature review and presentation of the research background, three main research fields are overviewed. First, the intelligent space related researches are introduced. This section includes the overview of network technology, ubiquitous computing, and cloud computing. Then, the robot technology is presented. This includes the introduction of intelligent robotics and human-friendly robotics. Finally, the intelligence technology is introduced. This section details the three main sub-fields of soft computing, namely, fuzzy systems, neural networks, and evolutionary computation.

### 1.2.1 Intelligent Space

#### 1.2.1.1 Network Technology

In recent years, network communication has become an important part in our life. For example inside office building, there is Local Area Network (LAN), which connects limited number of computers and Wide Area Network WAN, which connects LAN to form a wider network. While LAN and WAN are used by limited people, Internet is a network, which can be used by anyone. Internet protocol is used to communicate among Internet users. With today's technology such as WIFI, everyone can use Internet without cables. In the fourth generation of wireless network, communication distance based technology can be divided into: WAN for distance above 10 km, WMAN (Wireless Metropolitan Area Network) for distance between 100 m–10 km, WLAN (Wireless Local Area Network) for distance between 10 m–100 m, WPAN (Wireless Personal Area Network) for distance between 1 m–10 m, and NPC (Near Field Communication) for distance below 1 m. WMAN is used to cover a city, WLAN is using IEEE802.11b/g/n standard inside a building. Meanwhile, WPAN is used as personal network communication such as low energy cost Bluetooth (IEEE802.15.1) with 1 Mbps communication speed and ZigBee with only communication speed 20–250 kbps, but using low energy, economy, and high reliability. For NFC, Sony has built IC card Felica, while JR East Japan has developed Suica. Others technology such as RFID (Radio Frequency Identification) known as electrical tag, has been installed in a lot of devices and used for wireless communication technology. Apart from these technologies, more recent technologies for NFC such as BAN (Body Area Network), HEMS (Home Energy Management System), and HAN (Home Area Network) have been developed to support the needs of communication system.

With these technologies of both wire and wireless network that can be installed everywhere, the important aspect in today's technology is the intercommunication between devices rather than the type of device or computer.

#### 1.2.1.2 Ubiquitous Computing

Ubiquitous computing [124, 125, 78] is a concept in software engineering and computer science where computing is made to appear anytime and everywhere. In contrast to desktop computing, ubiquitous computing can occur using any device, in any location, and in any format. The underlying technologies to support ubiquitous computing include Internet, advanced middleware, operating system, mobile code,

Table 1.1: Ubiquitous terms

| | |
|---|---|
| Ubiquitous network | This term was first quoted by Nomura research center in 1999. They defined that ubiquitous network as an environment that can access internet anywhere and anytime using private mobile network. |
| Ubiquitous society | This term means that information communication tools are distributed in the environment around us and by using its communication feature, various services can be generated based on the recognized condition. |
| Ubiquitous computing | This term means that computers distributed in the real world environment which enable users to gain information based on their preferences using their own computer or other devices such as smartphone or tablet computer. |

sensors, microprocessors, new I/O and user interfaces, networks, mobile protocols, location and positioning and new materials. Below, I will explain related technologies to ubiquitous computing.

Definition of ubiquitous computing: The word ubiquitous came from the Latin word “ubique” means existing everywhere. The phrase ubiquitous computing was quoted first by Mark Weiser during his tenure as Chief Technologist of the Xerox Palo Alto Research Center (PARC) when explaining about human interface. The word ubiquitous [85] can be combined with other words to form new terms such as ubiquitous network, ubiquitous society, and ubiquitous computing as presented in Table 1.1.

By integrating these terms, Mori defined ubiquitous to have three layers such as infrastructure layer, interaction layer, and service layer. Infrastructure layer pointed to ubiquitous network. Sensor network, network, and data communication technology are included in this layer. The interaction layer explains the methods on how the services and contents are presented to the users. The service layer explains what kind of services and contents should be offered to the users. Related technologies are presented in Table 1.2.

#### 1.2.1.3 Cloud Computing

In contrast with the previous computer management concept which used private facility to keep the software and data, cloud computing uses internet to store data

Table 1.2: Related technologies

| | |
|---|---|
| Everywhere computer | Before Mark Weiser introduced ubiquitous computing, in 1984 Ken Sakamura from University of Tokyo introduced everywhere computer [103]. This concept leads to the development of TRON project. Based on Ken Sakamura, the essence of everywhere computer is that computer and network recognize the context awareness of life space condition. |
| Pervasive Computing | Pervasive computing [80, 102] goes beyond the realm of personal computers, it is the idea that almost any device, from clothing to tools to appliances to cars to homes to the human body to your coffee mug, can be imbedded with chips to connect the device to an infinite network of other devices. The goal of pervasive computing, which combines current network technologies with wireless computing, voice recognition, Internet capability and artificial intelligence, is to create an environment where the connectivity of devices is embedded in such a way that the connectivity is unobtrusive and always available. |
| Calm Computing | Calm technology [126] is a type of information technology where the interaction between the technology and its user is designed to occur in the user's periphery rather than constantly at the center of attention. Information from the technology smoothly shifts to the user's attention when needed but otherwise stays calmly in the user's periphery. Mark Weiser and John Seely Brown describe calm technology as "that which informs but doesn't demand our focus or attention". The use of calm technology is paired with ubiquitous computing as a way to minimize the perceptible invasiveness of computers in everyday life. |
| Ambient Intelligence | This phrase was introduced by Philips research laboratory in the late 90s for explaining the next generation of communication society and broadly used in Europe. The main difference between ubiquitous computing and ambient intelligence [1, 20] is, that ubiquitous computing uses passive method for accessing human information through computer, while in ambient intelligence the machines work autonomously when interacting with human. |

Table 1.3: Historical computer operation

| | |
|---|---|
| Mainframe operation | Computer operation done with 1 computer |
| Client-server operation | Distributed computer operation |
| Network operation | Internet based computer operation |
| Cloud computing | Unrecognized server operation |

Table 1.4: Concept of computing

| | |
|---|---|
| Utility computing [95] | Computing resources are threated as electricity and water supply, when needed the important parts only are used. |
| The network is the computer [53] | Computers are connected by a network, where computer is the network itself. |
| Ubiquitous computing | Computers spread ubiquitously in a network that operates autonomously. |
| Grid computing [30] | Big scale of computer resources connected in a network similar to a supercomputer, where important part is allocated to be used. |

and software, which can be used whenever needed.

The company offering cloud computing service provides big scale of data center and numerous servers which can be controlled remotely to manage software and data. When users register to that service, they can use software directly and also can store data on the server.

Historical computer operation over time is presented in Table 1.3 and the concept of computing is presented in Table 1.4.

By using these concepts, combined with virtual and automatic operation technology as well as the spread of Internet, cloud computing was born. The types of cloud computing based on service model are presented in Table 1.5. Cloud deployment models are presented in Table 1.6 [22, 6].

Advantages:

- Data management can be done in one place
- Personal computer operation becomes easy, since software installation or updates are not required anymore
- Decreasing the operational cost
- Risk management can be realized, since the system is distributed

Disadvantages:

Table 1.5: Types of cloud computing based on service model

| | |
|---|---|
| SaaS (Software as a Services) | This type of cloud computing offers software package through Internet as well as e-mail, groupware and CRM. Sample of this kind of services include Sales force CRM, Microsoft Online Service, and Google App. |
| PaaS (Platform as a Service) | This type of cloud computing offers application execution platform. This service provides virtual application server and database where we can put our application to operate. Sample of this services include Salesforce dotcom, Windows Azure, Google App Engine, and Amazon. |
| HaaS or IaaS (Infrastructure as a Service) | This kind of service offers hardware and infrastructure through Internet. For example virtual server and virtual desktop. |

Table 1.6: Cloud deployment models

| | |
|---|---|
| Public cloud | This is the service that uses cloud computing as a tool and public community as a service target. |
| Private cloud | This is the service used inside a company which connects departments or groups inside the company through cloud computing. |
| Hybrid cloud | This service combined public and private cloud which operates based on the characteristics of the system. |
| Community cloud | This service targets a specific community in conducting cloud computing. Comparing to hybrid cloud, community cloud offers higher level of reliability, and comparing to private cloud, community cloud has a higher construction cost, but also has higher flexibility. |

Table 1.7: Robotics applications of cloud computing [94]

| | |
|---|---|
| Remote Computation | Cloud computing enables remote computation by enabling massively parallel computation on demand. Public domain cloud service providers such as Amazon Web Services Elastic Compute Cloud, known as EC2, Google Compute Engine, and Microsoft Azure allow computing resources to be rented for various computing tasks. |
| Shared Knowledge-base | Autonomous robots are typically equipped with multiple sensors, cameras etc. Data collected from these sensors are normally used for making decisions on localization, movement, path determination etc for the robot. Sharing this data among multiple robots increases the volume to an extent that managing it would overwhelm limited capacities onboard a robot. |
| Collective Learning | Cloud computing provides an excellent platform for networked robots to share data, process information and improve perception by various machine learning methods. |
| Cloud Architectures | Cloud provides scalable resources for sharing information between humans, systems and robots. Recently many works have provided middle-wares, frameworks and platforms for Cloud robotics. |

- Difficulty to change specific features
- When the service is finished it cannot be used
- Safety issues, since the data management is done in one place
- Privacy issue

Cloud computing has robotics applications as well as presented in Table 1.7 [94]. Since standalone robot operation could have an expensive cost, therefore by using cloud computing small cheap robot with limited features can be used for optimized application.

### 1.2.2 Robot Technology

#### 1.2.2.1 Intelligent Robotics

In my opinion, a robot, which can acquire and apply knowledge or skill, can be called intelligent. Many methodologies have been applied for intelligent capabilities such as learning, reasoning, predicting, communicating, and decision making. Various methodologies concerning artificial intelligence (AI) have been developed in order to describe and build intelligent agents that perceive an environment, make appropriate decisions, and take actions. In a classical point of view, an intelligent agent was designed based on symbolic representation and manipulation of explicit knowledge. A search algorithm using symbolic approach finds a path from an initial state to final state in a graph or search tree generated as a world model. This kind of search algorithm can be divided into three types of breadth-first search, depth-first search, and best-first search. The breadth-first search expands the shallowest node in a graph, while the depth-first search expands the deepest node in a graph. The best-first search expands the node that minimizes an evaluation function based on task-dependent information. A useful best-first search is A* algorithm. Recently, human intelligence and life itself have been discussed in cognitive science, soft computing, artificial life, and computational intelligence. Soft computing proposed by L.Zadeh is a new concept for information processing. Its objective is to realize a new approach for analyzing and creating flexible information processing of human being such as sensing, understanding, learning, recognizing and thinking. In addition, neural computing, fuzzy computing, and evolutionary computing have been used as intelligent techniques. Neural computing and fuzzy computing is based on the mechanism of human brain. While neural computing simulates physiological features of human brain, fuzzy computing simulates psychological features of human brain (Fig. 1.2). Each technique is not complete for realizing all features of intelligence, and therefore, hybridized or combined methods have been proposed for building intelligent robots.

The usual approach to building control architectures for a mobile robot in a classical approach is functional decomposition: First, there is sensing and perception. In this step, information from different sensors including vision, auditory, and tactile is integrated into a central representation. As internal processing, building or updating a model of the environment often called a world model is performed. Next, decision making determines action to actually execute. Finally, some actions are executed in the real world. In this way, functional decomposition of a task leads to the sense-think-act cycle of the traditional information processing approach. Various methods using this architecture have been proposed so far, but map building and collision avoidance problems are still dealt with as application examples.

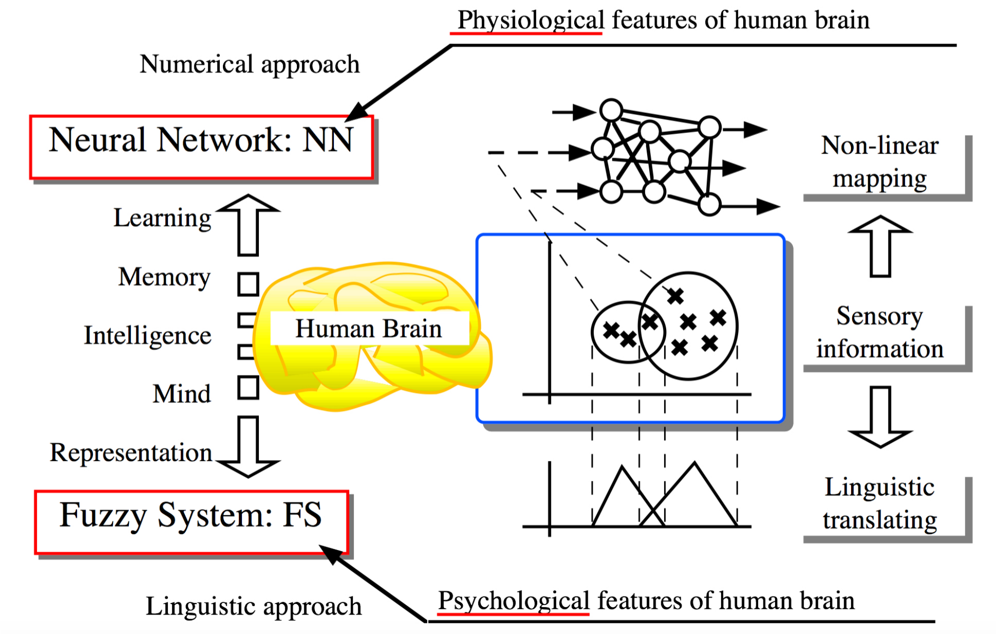



Figure 1.2: Neural computing and fuzzy computing

#### 1.2.2.2 Human-Friendly Robotics

The expectation for the human-friendly and functional robots is getting increased, such as care for aged and sick people, or a substitute for personal service that helps human people with their daily lives and their activities. Robotics can widely impact quality of life, serving as assistants to the elderly or incapacitated (Fig. 1.3).

In recent years, the progress of technological improvement has enabled us to develop communicative robots. A human-friendly robot can enjoy basic conversation with human people by using natural language processing and image processing for extracting human expression. Especially, social communication between human people and robots is very important. These robots are expected to co-exist and work together with human people in the everyday scene such as offices, hospitals and homes. These everyday-robots should be designed not for achieving a single task, but for generating various types of behavior according to the situations. Therefore, such robots must have a variety of sensors for perceiving its environment and actuators for realizing flexible and adaptive behaviors. Figure 1.3 shows some human-friendly robots. Hubot has various sensors for detecting human people around the robot. The robot should have multi-modal interface holding diverse channels of communication, and should be able to recognize the environment including human behavior in order to share the current situation with human people. Furthermore, robots are expected

(a) PALRO
(©PR TIMES, Inc.)

(b) PaPeRo petit
(©NEC Corporation)

(c) Jibo
(©Cond Nast Japan )


(d) Pepper
(©Impress Corporation.)

Figure 1.3: Robot partners

to interact with humans directly in a safe and comfortable way.

One task for such human-friendly robot is rehabilitation. A crucial factor in realizing the rehabilitation will be the ability of the robotic technology to move away from conventional materials and actuators, which are felt cold and stiff, and use innovative solutions for compliant, soft-moving hands and manipulators.

Development of human-like robot is still a dream of human people. A robot of the next generation can perform various tasks such as driving assistance, healthcare, and tutoring assistance. In order to realize such a robot, various technologies and concepts in electronics, mechanics, informatics, computer science, brain science, psychology, and sociology are required.

### 1.2.3 Intelligence Technology

Computational intelligence (CI) [26, 100] is set of computational methodologies and approaches such as neural network computing, fuzzy computing, evolution strategy, swarm intelligence, etc., to address complex real-world problems. Artificial Intelligence (AI) is a top-down approach related to symbol representation and manipulation. In contrast to AI, CI is a bottom-up approach related to numerical computation.

Meanwhile, soft computing is proposed by Lotfi Zadeh, the developer of fuzzy sets. It was a concept of information processing, and aimed to realize a flexible information processing for huge size and complex problem in the environment which have uncertain and vague information. While CI is discussing intelligence from the view of symbol processing and numerical computation, soft computing is discussing intelligence from the view of human information processing mechanism. However, in recent years, there is a trend to consider CI and soft computing in the same way. Both of these methodologies are called intelligence technology. In this section I will explain neural network computing, fuzzy computing, and evolutionary computation. Neural network computing is imitating a psychological aspect of the human brain. Fuzzy computing is imitating a physiological aspect of brain. Evolutionary computation is imitating the evolutionary processing in the nature.

#### 1.2.3.1 Fuzzy Systems

Fuzzy computing [40, 43] is the methodologies related to fuzzy theory and fuzzy information processing. Although computers have high accuracy and high-speed calculation capability, however it is difficult to deal with situations and data containing complex fuzziness. Human can flexibly deal with these situations. In order to quantitatively express and process human subjective judgment and fuzziness, Lotfi Zadeh proposed the fuzzy set theory in 1965 [133, 7]. Since then, until the 1980s fuzzy system was an active research field. Then, the research in fuzzy systems experienced a dark age in the 1980s, but it was reborn by Japanese researchers in the late 1980s. Today, it is a very active research field again with many successful applications, especially in control systems.

In the original crisp set theory (Fig. 1.4, 1.6), a set $A$ in the universe $X$ is defined by its characteristic function which can have two values as follows:

$$\mu_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases} \tag{1.1}$$

On the other hand, in fuzzy set theory, a set $A$ is defined by its membership function (Fig. 1.5):

$$\mu_A : X \to [0,1] \tag{1.2}$$

Fuzzy set describes a possibility distribution, and it can effectively handle things with vague state. There are different types of membership functions. The most common types are triangular, trapezoidal, and Gaussian (Fig. 1.7, 1.8, 1.9).

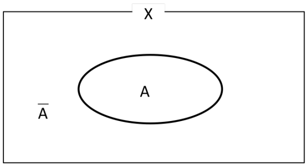



Figure 1.4: Crisp set

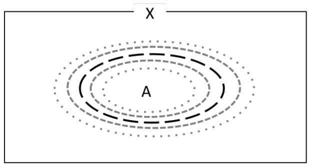



Figure 1.5: Fuzzy set

The generalization of set operators to fuzzy sets is necessary. The basic set operators such as intersection, union, and complement can be defined in many ways in the case of fuzzy sets. The most common definitions (as proposed by Zadeh) are:

Intersection:

$$\mu_{A\cap B}(x) = \mu_A(x) \wedge \mu_B(x) = min[\mu_A(x), \mu_B(x)] \tag{1.3}$$

Union:

$$\mu_{A\cup B}(x) = \mu_A(x) \vee \mu_B(x) = max[\mu_A(x), \mu_B(x)] \tag{1.4}$$

Complement:

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \tag{1.5}$$

where $\mu_A(x)$ and $\mu_B(x)$ are the membership functions of fuzzy set $A$ and fuzzy set $B$, respectively.

Fuzzy inference is executed using fuzzy If-Then inference rules. It can roughly be divided into direct method and indirect method. Indirect method is a method that indirectly asks the inference result through the truth space. In the direct method,

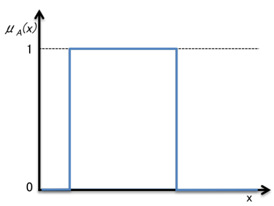



Figure 1.6: Crisp function

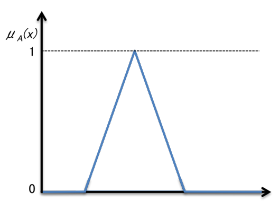



Figure 1.7: Triangular function

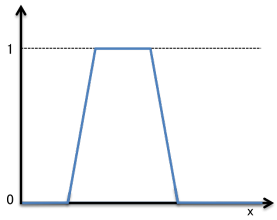



Figure 1.8: Trapezoidal function

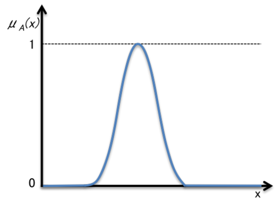



Figure 1.9: Gaussian function

when the fuzzy sets are $A_{j,k}$, $B_{j,i}$ of the $j$th fuzzy rule, the fuzzy rules can be expressed as follows:

IF $x_1$ is $A_{j,1}$ and $x_2$ is $A_{j,2}$ and $\cdots$ $x_n$ is $A_{j,n}$

THEN $y_1$ is $B_{j,1}$ and $y_2$ is $B_{j,2}$ and $\cdots$ $y_m$ is $B_{j,m}$

where $x_1, \cdots, x_n$ are input variables, $y_1, \cdots, y_m$ are output variables.

Typically, the procedure of fuzzy inference is as follows.

1. Calculate the degree of matching of each rule with the given input.
2. Calculate each inference result based on step 1.
3. Calculate the final inference result based on each inference result.

Various calculation methods can be applied in each step, but their computational cost and inference result are different. Typical methods are as follows:

1. Fuzzy min-max center of gravity methods
2. Product-Sum-Gravity method
3. Simplified reasoning method
4. Functional-type reasoning method

#### 1.2.3.2 Neural Networks

In the middle of 80's, the interest in artificial neural networks started to increase, making artificial neural networks (ANN) a very important research field in CI. Several different architectures have been proposed, trying to exploit the available knowledge of the mechanisms of the human brain.

Based on the biological characteristics of neuron, McCulloch and Pitts proposed a basic model of neuron with a binary threshold unit as a computational model as depicted in Fig. 1.10, where we can see that in this model a neuron is composed of

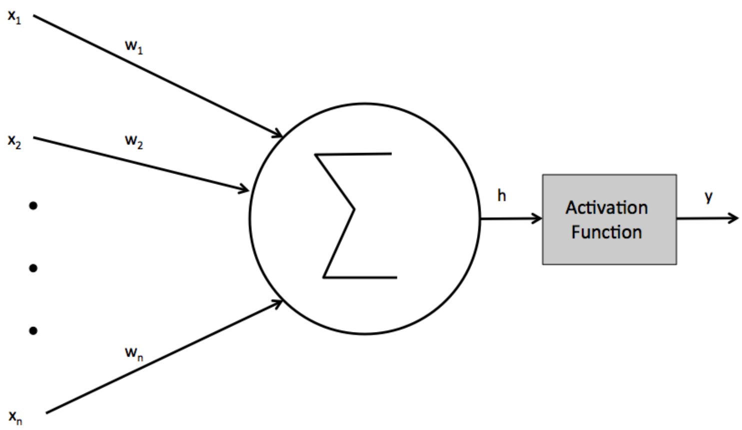


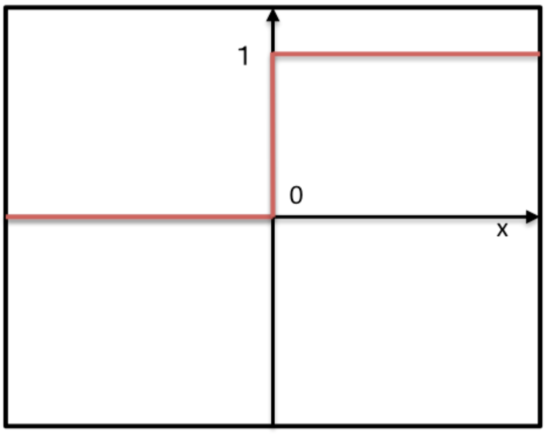
Figure 1.10: McCulloh-Pitts model of neuron

input, output, synaptic strength, and activation function. The output of neuron is produced based on activation function after the summing of inputs. The output of neuron before and after the activation function can be calculated as follows.

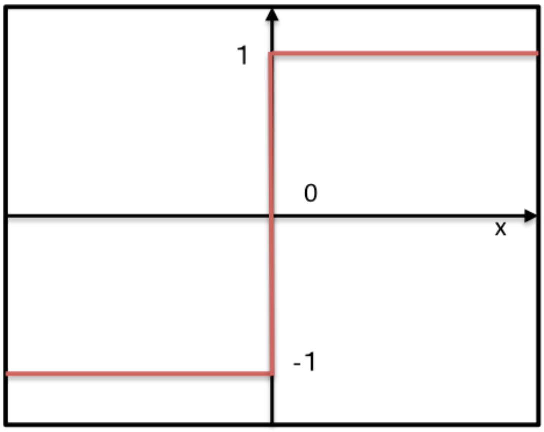
$$h = \sum_{j=1}^{n} w_j x_j - \theta \tag{1.6}$$

$$y = f(h) = f\left(\sum_{j=1}^{n} w_j x_j - \theta\right) \tag{1.7}$$

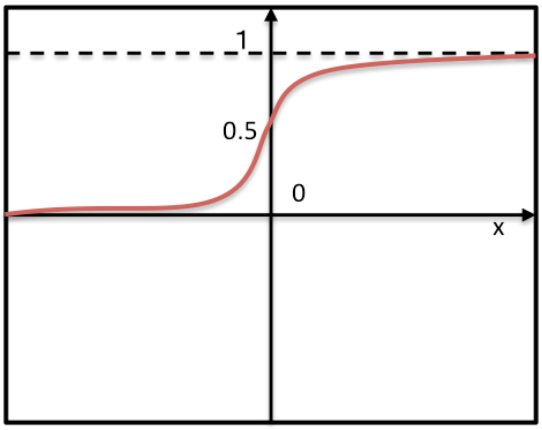
where $x$ and $y$ are input and output, respectively; $w$ is the weight parameter, $\theta$ is the threshold, and $n$ is the number of neurons; $f$ depicts the activation function, where basically step, sign, sigmoid, and hyperbolic tangent functions are often used as shown in Fig. 1.11.



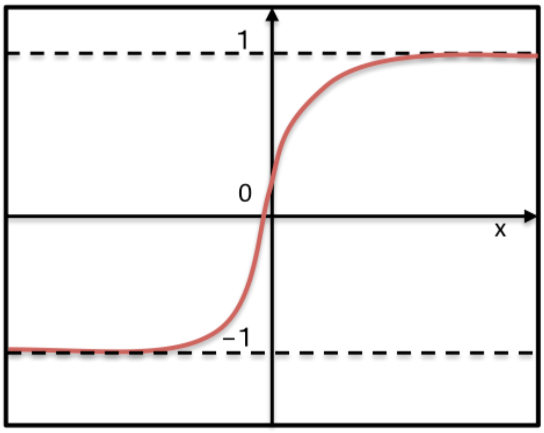

(a) A step function




(b) A sign function




(c) A sigmoid function




(d) A hyperbolic tangent function

Figure 1.11: Activation function

**Single-Layer Perceptron:**

The Perceptron, proposed by Rosenblatt, is composed of sensory unit, $S$, association unit, $A$, and response unit, $R$, where a step function is used as the activation function. The learning algorithm only adjusts the weights between $A$ and $R$. The original Perceptron is used for pattern recognizer with linear separability. Currently, the name of perceptron is used as a single layer feedforward network. A perceptron is trained by the delta rule based on the least square error minimization method. The output of the perceptron is calculated by Eq. (1.7), where a logistic function is used as the activation function. The error function is calculated as follows

$$E_p = \frac{1}{2}\sum_{i=1}^{n}(Y_{p,i} - y_i)^2 \tag{1.8}$$

where $Y_{p,i}$ is the desired value for the $i$th output of pattern $p$. By the chain rule, the derivative is calculated as follows

$$\frac{\partial E_p}{\partial w_{j,i}} = \frac{\partial E_p}{\partial y_i}\frac{\partial y_i}{\partial z_i}\frac{\partial z_i}{\partial w_{j,i}} = -(Y_{p,i} - y_i)f'(z_i)x_j \tag{1.9}$$

Furthermore, because a logistic function is used, the above equation becomes as follows

$$\frac{\partial E_p}{\partial w_{j,i}} = -(Y_{p,i} - y_i)f'(z_i)(1 - f(z_i))x_j \tag{1.10}$$

Therefore, the weight is updated by gradient descent:

$$\Delta_p w_{j,i} = -\eta\frac{\partial E_p}{\partial w_{j,i}} = -\eta(Y_{p,i} - y_i)f'(z_i)(1 - f(z_i))x_j \tag{1.11}$$

where $\eta$ is a learning rate within (0,1). The delta rule tries to minimize the squared errors iteratively. The architecture is very simple, but it is pointed out that the single layer perceptron cannot be used for learning the XOR problem, because the XOR problem is not linearly separable.

**Multilayer Perceptron:**

Multilayer perceptron is often applied to complex problems. In general, a multilayer perceptron is composed of many perceptrons in a hierarchical structure with one or more hidden layers. The number of layers in the most widely used multilayer perceptrons is usually 3 or 4. The thesis deals with a multilayer perceptron with $l$

layers ($l > 2$). Its calculation method is a simple extension of the perceptron. The most popular learning method is the backpropagation learning algorithm that is also known as a generalized delta rule. To simplify, this thesis assumes the number of neurons in each layer is $n$. Using the inputs in the $l-2$ layer, the output of a neuron in the $l$-th layer can be calculated recursively as follows

$$z_i^l = \sum_{j=1}^{n} w_{j,i}^l \cdot x_j^{l-1} - \theta_i^l \tag{1.12}$$

$$y_i^l = f(z_i^l) = f\left(\sum_{j=1}^{n} w_{j,i}^l \cdot x_j^{l-1} - \theta_i^l\right) = f\left(\sum_{j=1}^{n} w_{j,i}^l \cdot f\left(\sum_{k=1}^{n} w_{k,j}^{l-1} \cdot x_k^{l-2} - \theta_i^{l-1}\right) - \theta_i^l\right) \tag{1.13}$$

where the $x_j^{l-1}$ and $x_k^{l-2}$ are the $j$-th input in the $(l-1)$-th layer and the $k$-th input in the $(l-2)$-th layer, respectively ($j = 1, 2, \ldots, n$; $k = 1, 2, \ldots, n$). If the $l$-th layer is the output layer, the partial derivative with respect to $w_{j,i}^l$ is derived by the chain rule,

$$\frac{\partial z_i^l}{\partial w_{j,i}} = -(Y_{p,i} - y_i^l) f'(z_i^l) x_j^{l-1} \tag{1.14}$$

The error signal in the output layer is defined as

$$\frac{\partial E_p}{\partial w_{j,i}^l} = \frac{\partial E_p}{\partial y_i^l} \frac{\partial y_i^l}{\partial z_i^l} \delta_{p,i}^l = -(Y_{p,i} - y_i^l) f'(z_i^l) \tag{1.15}$$

Then, the weight is updated according to

$$\Delta_p w_{j,i}^l = -\eta \frac{\partial E_p}{\partial w_{j,i}^l} = -\eta \delta_{p,i}^l x_j^{l-1} \tag{1.16}$$

If the $(l-1)$-th layer is a hidden layer, the partial derivatives are

$$\frac{\partial E_p}{\partial w_{k,j}^{l-1}} = \frac{\partial E_p}{\partial z_j^{l-1}} \frac{\partial z_j^{l-1}}{\partial w_{k,j}^{l-1}} = \frac{\partial E_p}{\partial z_j^l} \frac{\partial z_i^l}{\partial x_j^{l-1}} \frac{\partial x_j^{l-1}}{\partial z_j^{l-1}} \frac{\partial z_j^{l-1}}{\partial w_{k,j}^{l-1}} = \sum_{i=1}^{n} \delta_{p,i}^l w_{j,i}^l f'\left(z_j^{l-1}\right) x_k^{l-2} \tag{1.17}$$

Furthermore, the error signal in the $(l-1)$-th layer is defined as,

$$\delta_{p,j}^{l-1} = \frac{\partial E_p}{\partial z_j^{l-1}} = \sum_{i=1}^{n} \delta_{p,i}^l w_{j,i}^l f'\left(z_j^{l-1}\right) \tag{1.18}$$

This equation includes the error signals of the $l$-th layer. This indicates the error signals can be calculated backward after the error signals of the output layer are obtained. Therefore, this procedure is called a backpropagation learning algorithm. The weight is updated according to

$$\Delta_p w_{k,j}^{l-1} = -\eta \frac{\partial E_p}{\partial w_{k,j}^{l-1}} = -\eta \delta_{p,j}^{l-1} x_k^{l-2} \tag{1.19}$$

**Spiking Neural Network:**

Generally, ANN can be divided into pulse-coded and rate-coded neuron models from the view point of abstraction [38]. The pulse-coded neuron approximates the dynamics created from the ignition phenomenon of neuron and also simulates propagation mechanism of the pulses between neuron. The pulse-coded neuron also known as spiking neuron. One of the classical neuronal spiking model is Hodgkin-Huxley model. This model uses four differential equations. On the other hand, the rate-coded neuron neglects the pulse structure and is considered to have higher level of abstraction. The McCulloch-Pitts model of ANN is well known as the rate-code model, while perceptron was proposed as a rate coded neural network [5].

One of the important features of spiking neurons is the capability in conducting temporal coding. Spiking neural networks have been implemented to memorize spatial and temporal context. Therefore, spiking neural networks are used to represent the time series of perceptual information. In the thesis always a modified simple spike response model is used to reduce computational cost.

Basically, the calculation in spiking neural network is conducted through the following steps. The membrane potential, or internal state $h_i(t)$ of the $i$-th neuron at the discrete time $t$ is given by:

$$h_i(t) = \tanh(h_i^{syn}(t) + h_i^{ref}(t) + h_i^{ext}(t)) \tag{1.20}$$

where $h_i^{syn}(t)$ includes the pulse outputs from the other neurons, $h_i^{ref}(t)$ is used for representing the refractoriness of the neuron, $h_i^{ext}(t)$ is the input to the $i$-th neuron from the environment. The hyperbolic tangent function is used to avoid the bursting of neuronal fires.

The first term $h_i^{syn}(t)$ is calculated as follows:

$$h_i^{syn}(t) = \gamma^{syn} \cdot h_i(t-1) + \sum_{j=1, j\neq i}^{N} w_{j,i} \cdot h_j^{PSP}(t-1) \tag{1.21}$$

where $\gamma^{syn}$ is the temporal discount rate, $w_{j,i}$ is a weight from the $j$-th neuron to the $i$-th neuron, $h_j^{PSP}(t)$ is the presynaptic action potential (PSP) approximately transmitted from the $j$-th neuron at the discrete time $t$, and $N$ is the number of neurons. When the internal state of the $i$-th neuron reaches the predefined threshold, a pulse is outputted as follows:

$$p_i(t) = \begin{cases} 1 & \text{if } \; h_i(t) \geq \theta, \\ 0 & \text{otherwise}, \end{cases} \tag{1.22}$$

where $\theta$ is a threshold for firing. When the neuron is fired, $R$ is subtracted from $h_i^{ref}(t)$:

$$h_i^{ref}(t) = \begin{cases} \gamma^{ref} \cdot h_i^{ref}(t-1) - R & \text{if } \; p_i(t-1) = 1, \\ \gamma^{ref} \cdot h_i^{ref}(t-1) & \text{otherwise}, \end{cases} \tag{1.23}$$

where $\gamma^{ref}$ is a discount rate and $R > 0$.

The presynaptic spike output is transmitted to the connected neuron according to PSP through the weight connection. The PSP is calculated as follows:

$$p_i(t) = \begin{cases} 1 & \text{if } \; p_i(t-1) = 1, \\ \gamma^{PSP} \cdot h_i^{PSP}(t-1) & \text{otherwise}, \end{cases} \tag{1.24}$$

where $\gamma^{PSP}$ is a discount rate and $(0 < \gamma^{PSP} < 1)$. Therefore, the postsynaptic action potential is excitatory if the weight parameter, $w_{j,i}$ is positive. If the condition $h_j^{PSP}(t-1) < h_i^{PSP}(t)$ is satisfied, the weight parameter $w_{j,i}$ is trained based on the temporal Hebbian learning rule as follows:

$$w_{j,i} \leftarrow \tanh(\gamma^{wgt} \cdot w_{j,i} + \xi^{wgt} \cdot h_j^{PSP}(t-1) \cdot h_i^{PSP}(t)) \tag{1.25}$$

where $\gamma^{wgt}$ is a discount rate and $\xi^{wgt}$ is a learning rate.

**Self Organizing Map (SOM):**

Unlike the previous section, unsupervised learning or sometimes also called clustering is used for grouping or segmenting data into subsets or clusters based on similarity [41]. Unsupervised learning is performed by using data without any teaching signals [60, 83, 31, 32]. k-Means and Gaussian mixture model are two of the most used clustering methods that use all data in the learning phase (batch learning). Meanwhile, Self-Organizing Map (SOM) , Neural Gas (NG) [83], Growing Cell Structures (GCS) [31], and Growing Neural Gas [32] are some well known unsupervised learning methods that use the competitive learning approach. Two unsupervised learning methods are discussed in this thesis, which are Self Organizing Map (SOM) and Growing Neural Gas (GNG).

SOM was developed by Kohonen [60]. SOM is a special type of competitive learning network that defines a spatial neighborhood for each output unit. SOM is often applied for extracting relationship among observed data, since SOM can learn the hidden topological structure from data. Regarding as topological structure extraction, SOM preserves the relative distance between the points while conducting mapping. Points that are near each other in the input space are mapped to nearby map units in the SOM. The SOM can thus serve as a cluster analyzing tool of high-dimensional data. The algorithm of SOM is presented in Algorithm 1.

**Algorithm 1** Self-Organizing Map

**Step 1:** Initialize the reference vector $m_i(0)$ randomly.

**Step 2:** Present the input data $\mathbf{x(t)}$.

**Step 3:** Select the winning vector of the reference vector using the following equation

$$c = \arg\min_i ||\mathbf{x}(t) - \mathbf{m}_i(t)||, \tag{1.26}$$

where $c$ is the node number of the winning vector, $\mathbf{x}(t)$ is the input data at time $t$, and $\mathbf{m}_i(t)$ is the reference vector $i$ at time $t$.

**Step 4:** Update the reference vector $\mathbf{m}_i(t)$ using the following equation

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + h_{ci}(t)[\mathbf{x}(t) - \mathbf{m}_i(t)], \tag{1.27}$$

where $h_{ci}(t)$ is the neighborhood function. An example of neighborhood function can be shown as,

**(1)**

$$h_{ci}(t) = \begin{cases} \alpha(t) & \text{if} \quad i \in N_c \\ 0 & \text{otherwise} \end{cases} \tag{1.28}$$

**(2)**

$$h_{ci}(t) = \alpha(t) \cdot \exp\left(-\frac{||\mathbf{r}_c - \mathbf{r}_i||}{2\sigma^2(t)}\right), \mathbf{r}_c \in R^2 \quad \text{and} \quad \mathbf{r}_i \in R^2 \tag{1.29}$$

where $r$ is the node vector, $N_c$ is the neighborhood set, and $\alpha(t)$ is the learning rate.

**Step 5:** Repeat steps 2 through 4 until the convergence condition is fulfilled.

#### 1.2.3.3 Evolutionary Computation

There are several optimization methods inspired by processes in the nature. The advantage of these algorithms is their ability to solve and quasi-optimize problems with non-linear, high-dimensional, multi-modal, and discontinuous character. It has been shown that evolutionary algorithms are efficient tools for solving non-linear, multi-objective and constrained optimization. These algorithms have the ability to explore large admissible spaces, without demanding the use of derivatives of the objective functions, such as the gradient-based training methods. Their principles are based on the search for a population of solutions, which tuning is done using mechanisms similar to biological recombination. The original Genetic Algorithm was developed by Holland [44] and was based on the process of evolution of biological organisms.

Evolutionary algorithms imitate the abstract model of the evolution of populations observed in the nature. They represent candidate solutions for a given optimization problem as individuals of the particular population (set of candidate solutions). Their aim is to change the individuals in the population by applying operators imitating evolutionary effects in order to obtain better and better individuals, this way the algorithms try to find the optimal solution for the problem. The goodness of an individual can be measured by its fitness. Thus, the individuals represent elements of the search space and the fitness function serves as a transformation of the objective function. If an evolutionary algorithm uses an elitist strategy [39], then the best ever individual will always survive and appear in the next generation. As a result, at the end of the algorithm the best individual will represent the best discovered, i.e (quasi-)optimal element of the search space.

Genetic algorithms were used originally as a mechanism of natural adaptation. In a standard genetic algorithm, a candidate solution called individual is encoded into a finite length string called genotype, and the search is done in the genotype space. Therefore, the mapping from the genotype space into the phenotype space is required to evaluate a candidate solution. Binary coding or gray coding has been often used as the genotype space. Generally, a candidate solution is represented as an individual composed of genes $\{0, 1\}$. The position of a gene in the individual is called locus. A genetic algorithm uses simple symbolic operations on genotype space. The procedure of a standard GA is as follows;

begin

    Initialization

repeat

Roulette wheel selection

Crossover

Mutation

Evaluation

until Termination_condition = True

end.

One iteration of this procedure is called a generation. Initialization randomly generates an initial population of candidate solutions (individuals) and each individual is assigned fitness according to an objective function. Selection is used for reproducing the fitter individuals. Roulette wheel selection was originally proposed as a fitness-proportionate selection. This selection selects an individual with the following probability;

$$p_i = \frac{fit(x_i)}{\sum_{j=1}^{n} fit(x_j)} \tag{1.30}$$

where $fit(x_i)$ is the fitness value of the $i$-th individual $x_i$ and $n$ denotes the population size. Therefore, a fitter individual can reproduce more offspring. Other selection methods include Boltzmann selection, rank selection, and tournament selection. The selection probability of each individual in Boltzmann selection is as follows,

$$p_i = \frac{\exp\left(fit(x_i)/T\right)}{\sum_{j=1}^{n} \exp\left(fit(x_j)/T\right)} \tag{1.31}$$

where $T$ is temperature. The selection pressure can be adapted by updating $T$. However, the best individual in a population might be eliminated in the above stochastic selections. In order to solve this problem, elitism is introduced into a genetic algorithm. Elitist selection is used for maintaining the best fitted individual into the next population.

Crossover generates new individuals by exchanging the substrings between two individuals chosen randomly from a population. Crossover is designed according to the coding method, because crossover is a problem-dependent operator. Mutation changes a string by flipping some of the bits in a string at random. The generated offspring are evaluated by a fitness function or simulation-based evaluation. If it is very difficult to design fitness function, human evaluation can be directly used for evaluating candidate solutions. The method using human evaluation is called an interactive genetic algorithm. The series of the above processes is repeated until the termination condition is satisfied.

**Steady-State Genetic Algorithm (SSGA):**

Genetic algorithms modify a population of potential solutions during the course of a run, using both the application of operators such as crossover and mutation, and the application of a reproductive technique. There are two reproductive techniques. The first one is, which is probably the most widely used, generational reproduction, and the second one is steady-state reproduction [114]. Briefly, generational reproduction replaces the entire population at once, while steady-state reproduction replaces only a few members at a time. The two techniques are actually quite different.

According to the schema theorem [34], the performance of a genetic algorithm is highly dependent on its reproductive behavior. The schema theorem states that the rate of increase of above average schema in a population is directly dependent on the reproductive rate of individual population members.

## 1.3 Problems and Challenges

Although there are many researches about smart house and robot partner, however the discussion of their generalization, reliability and flexibility has to be extended and detailed. There are needs for general access methods and techniques for robot partner, human, and environment to the various types of information such as environmental state, Internet information, people, and devices. However, there are hardware and computational limitations of robot partner, which makes it difficult to acquire information by itself. Another difficulty is the complexity of environmental system when integrating a lot of different types of sensors. Environmental system can handle the sensing error and noise, and the sensing can be adapted by state of human and environmental changes. The importance of early anomaly and abnormality detection in daily life monitoring, as well as the needs to share that information to robot partner and related people are also essential. Because of above things there is a need to design, develop, and apply a concept for solving these challenges. This concept is the informationally structured space.

## 1.4 Overview of Approaches and Contributions

I discuss methodologies for information structuring by feature extraction based on bottom-up information collection, and information measurement based on top-down constraints. I also discuss the informationally structured space defining three layers: sensing layer, feature extraction layer, monitoring layer; and I define the function and structure of each layer. Sensing layer is for measurement, feature extraction layer is for extracting feature values from measurement data, and monitoring layer is for extracting temporal and spatial changes from feature values.

I explain the elements constructed the system such as sensor network, web system, and robot partner including gesture recognition and emotional model. In further, the conversational architecture which allows the natural communication to be realized is discussed. This includes the discussion of database structure and conversation selection algorithm.

In terms of the theoretical contribution, several theories are applied in the thesis such as:

- Relevance Theory – Realizing natural communication between human and robot is an important topic of the thesis. Natural communication can be realized when robot can understand human intention or thought. I implemented the theory of relevance proposed by Sperber and Wilson (1995) to build mutual cognitive environment between human and robot partner into the ISS system to handle this problem. According to Sperber and Wilson (1995), relevance theory is very useful to discuss the multimodal communication, where each person has his or her own cognitive environment that make their communication restricted. Therefore, usually humans use their utterances or gestures to expand their cognitive environment by extracting person's attention into specific target object, event, or person. When human's cognitive environment became wider, they can share each other intention or thought. The implementation of this theory into our system can be observed in the structure of database.

- Rasmussen's Behavior Theory - For conducting daily conversation with human, robot partner has to have enough knowledge and contents. However, to get enough knowledge and contents, the conversation system contents and task will become bigger. At this state, flexibility and simplicity of the system become a new issue. To deal with this, I propose new conversation system architecture based on Rasmussen's behavior model. Using this model, I divided the conversation into three types based on complexity.

- Soft Computing - In the gesture recognition, I implement structured learning involving growing neural gas for information extraction and spiking neural network to recognize the gesture. In the human state estimation and human interaction mode I apply evolution strategy and fuzzy spiking neural networks.

- Boltzmann Selection - For acquiring non-monotonic or lively conversation between human and robot partner I used Boltzmann selection by controlling the value of temperature to perform word selection when the robot partner communicates with the human.

## 1.5 Goal of the Thesis

The goal of this thesis is to propose informationally structured space for human daily life monitoring, and to discuss a human behavior measurement method, and methodologies for cooperation between the robot partner and the distributed sensing system composed of various sensors based on ISS. Moreover, in this thesis I will present the application of the system in the monitoring system through data measurement and analysis based on ISS. In order to clarify the design guidelines, I discuss the informationally structured space defining three layers: sensing layer, feature extraction layer, monitoring layer; and I define the function and structure of each layer. Sensing layer is for measurement, feature extraction layer is for extracting feature values from measurement data, and monitoring layer is for extracting temporal and spatial changes from feature values. Based on this structure I discuss methodologies for information structuring by feature extraction based on bottom-up information collection, and information measurement based on top-down constraints. Furthermore, I will also discuss methodologies for the system to flexibly adapt to changing environmental condition and system configuration.

## 1.6 Outline of Thesis

The thesis is organized as follows (Fig 1.12). Chapter 1 discusses the background and related researches. In this chapter I also explain the research purposes and goals. Chapter 2 introduces the basic properties of informationally structured space including information sharing between each device; information representation method which considers the affinity with human; reversibility of informational conversion between each device; and information operation method which considers system versatility. In this chapter, I also discuss the methods for connecting various information such as environmental condition, the Internet, and human condition acquired from informationally structured space. I discuss the informationally structured space defining layers and the function and structure of each layer. At the end of this chapter I discuss the database system for informationally structured space built as the result of interaction between robot partner and environmental system for estimating human behavior. Chapter 3 explains informationally structured space for distributed sensing system. This chapter begins with the explanation of the problem of outdoor and indoor measurement, global and local measurement in sensing layer from the viewpoint of environmental condition and from the target of the measurement. I propose human behavior measurement by applying spiking neural network, and structuring and updating the informationally structured space based on the relationship between location and human behavior. The experimental results show that the proposed method can flexibly measure the human behavior in indoor and outdoor. Chapter 4 presents informationally structured space for robot partner to enable human behavior active measurement. Here, I develop a gesture recognition system using evolutionary robot vision and conversation system based on time dependent utterance, and propose human behavior measurement in feature extraction layer. In this chapter, I also propose methodologies that complementarily use behavior information measured by the distributed sensing system and behavior information estimated by the robot partner. The experimental result through time dependent conversation shows, that the robot partner can estimate human behavior that is difficult to measure by the distributed sensing system. Chapter 5 discusses informationally structured space for monitoring layer in order to realize long-term human behavior measurement. Here, I propose daily life model estimation method using fuzzy modeling from long-term human behavior information. I construct live models with different granularities by day, week and so on. I also propose a method that specifies the difference between the life models. Moreover, I propose an abnormality detection method. At the end of the chapter, I develop a simulator system to imitate the informationally structured space, and I use this simulator system to perform numerical experiments in order to show the effectiveness of the proposed method. I explain the use of fuzzy modeling to model the long period of data for estimating daily life activities. I also explain the usage of smart device to visualize the simulation of system. This visualization leads to abnormality recognition as the result of the change in the life pattern. Chapter 6 concludes the thesis and explains the future research directions.

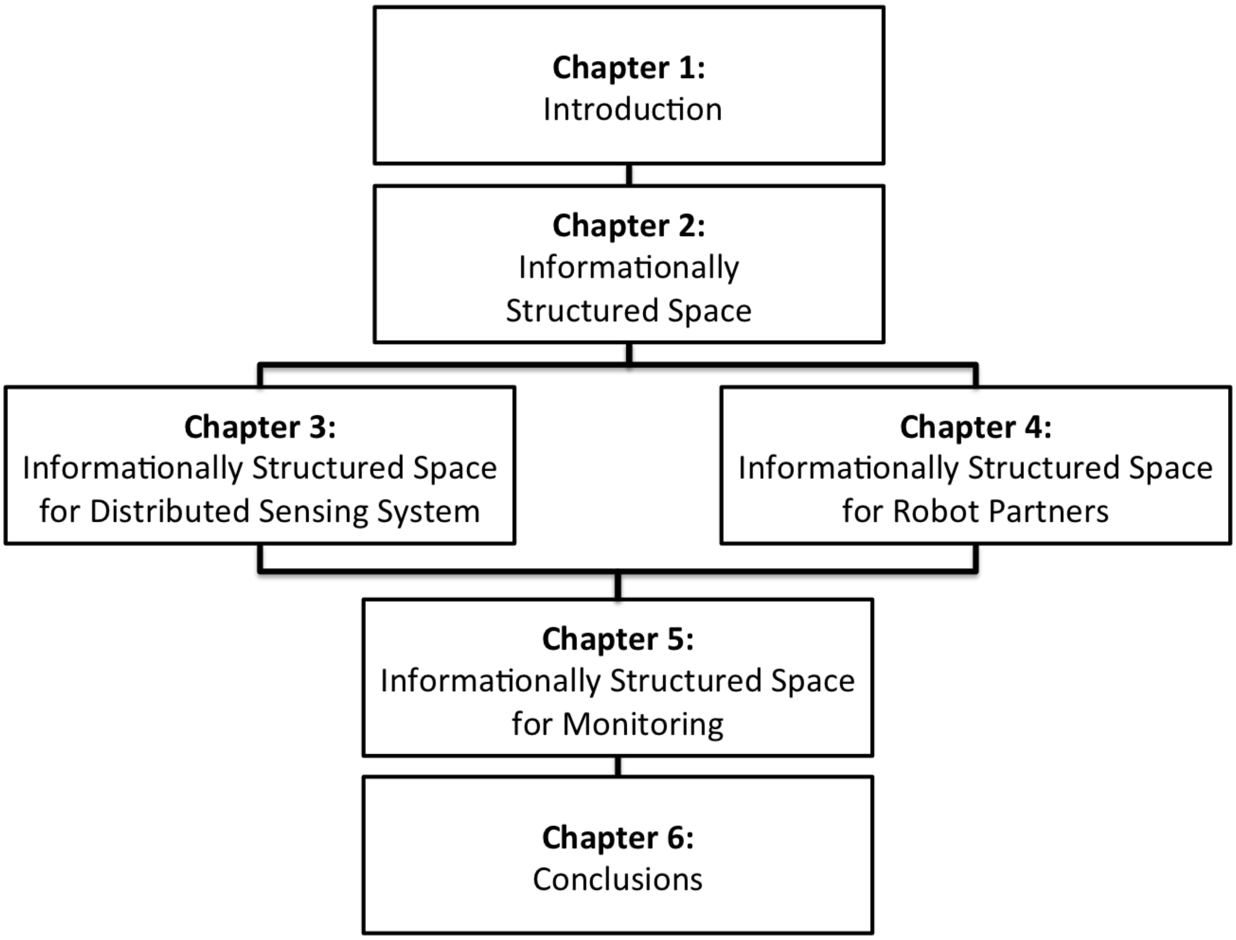



Figure 1.12: Outline of the thesis

# Chapter 2

# Informationally Structured Space

## 2.1 Introduction

Recently, ubiquitous computing [93, 81, 18, 47] has become one of the main attentions in the development of information technology. Ubiquitous computing can be defined as the opposite of virtual reality. While virtual reality puts people inside a computer-generated world, ubiquitous computing forces the computer to integrate the world with people [49, 74, 51, 98]. This technology can also be described as pervasive computing and ambient intelligence. Ambient intelligence is an emerging discipline that brings intelligence to our everyday environments and makes those environments sensitive to us [21]. The concept of sensor network and ubiquitous computing integrated into robotics can be called as network robotic and ubiquitous robotic [65, 58]. The network robotic is basically divided into three parts: visible robots, unconscious robots, and virtual robots [62]. The visible robots use their body to act with human. The unconscious robots are used to acquire environmental data, and the existence along with human is invisible. The virtual robot points out an agent or a software package in the cyber world. Based on these, I can make a conclusion that a robot can be used not only as a human-friendly life-support system, but it can also become an interface connecting the physical world with the cyber world [23, 61, 66].

### 2.1.1 Computational Systems Care

Nowadays, research concerning quality of life (QOL) and quality of community (QOC) has been developed in order to monitor and preserve the health of elderly people [25, 111, 52, 112, 48]. QOL means "… not merely the absence of disease, but physical, psychological and social well-being …" [91], Meanwhile QOC is defined as the preservation of a community in order to give an appropriate support which is not only focusing on individual, but also on the interaction between the members of the community itself. However, the quantitative evaluation of both physical and spiritual health has become one of the main issues. Moreover, it is also difficult to conduct the evaluation of elderly people's life condition. Recently, the research on the measurement and evaluation of Activities of Daily Living (ADL), which includes meal, taking a bath, and movement, as well as Instrumental Activities of Daily Living (IADL), which includes shopping, cleaning, and washing have been developed [127, 101, 9, 73, 122, 88]. By clearly defining the connection between ADL and QOL, we can develop various support for elderly people.

Recently, the importance of community-centric systems is increasing as a new paradigm in the aging society. In the American Heritage Dictionary, community is defined as [45] (1) A group of people living in the same locality and under the same government, (2) A group of people having common interests, (3) A group viewed as forming a distinct segment of society (4) A group of organisms interacting with one another and with the environment in a specific region. Each member of a community gains a personal and social identity by sharing common beliefs, values and standards. Social media have played an important role in creating, sharing, and exchanging information and ideas within a community. For example, if social media is available in a disaster, we can exchange and share disaster information in local community very quickly. Various types of assistive technologies should be developed from the human-centric and community-centric points of view to realize such information support. Human-centric systems can enhance the accessibility and usability of complicated systems and devices supporting human activities, communication, and interactions. Furthermore, the concept of human-centric approaches can improve QOL, especially, information support, physical care, and mental care to maintain autonomy and independency in ADL. QOL is deeply related with QOC. If QOC has been improved, the members of a community could feel better the improvement of QOL. This means, that the coupled improvement of QOC and QOL is very important. If the improvement of QOL has been done by the bottom-up construction, QOC can be considered as top-down constraint to human daily life. The coupled improvement of QOL and QOC enables to extend the healthy life expectancies of elderly people.

Information and Communication Technology (ICT) was introduced to improve the quality of life. The implementation of ICT can bind the personal relationship in the social community to increase the quality of community. When implementing ICT for elderly people we have to consider at least two situations, such as, the difficulty for elderly people to manage personal computers (PC) and the capability of elderly people

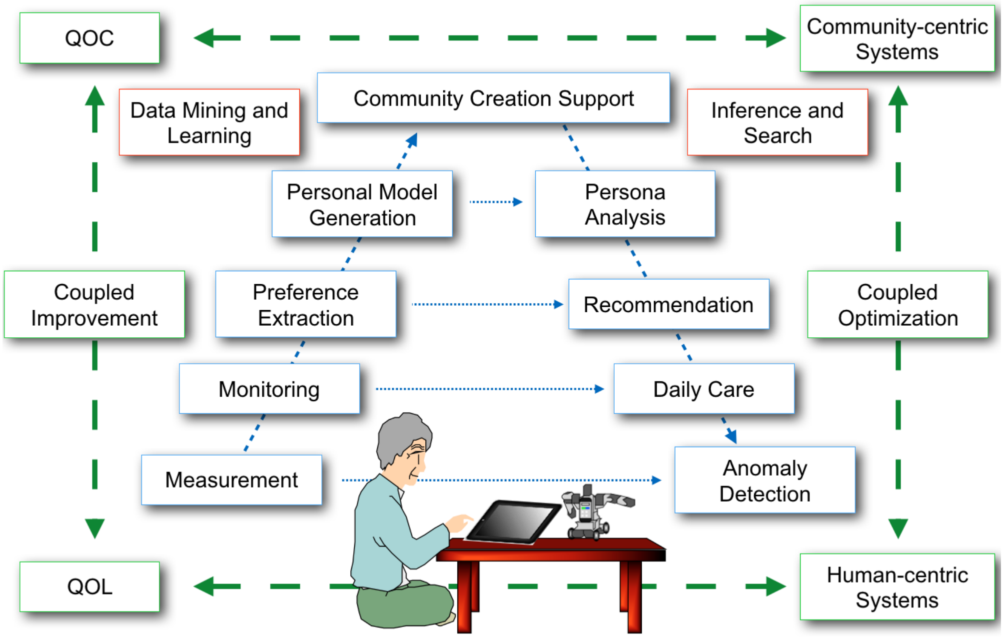



Figure 2.1: Concept of computational systems care

to use only basic functions of ICT, even if he or she is provided with an ICT device to control home appliances and to monitor the state of a living house. Implementing a smart device and a human-friendly robot partner can solve these problems. However, we need a synthetic system for elderly care. The synthetic approach for measurement, monitoring, anomaly detection, personal model generation, persona analysis, and care support, is called computational systems care (Fig. 2.1). This is composed of three main subsystems of distributed sensing system, informationally structured space servers, and robot partners. In the next section, I will explain informationally structured space and how to implement it for computational systems care.

### 2.1.2 Concept of Informationally Structured Space

A Distributed Sensing System can measure the number of people, human motions and behaviors as well as environmental state surrounding people. However, we have to deal with a large size of measurement data gathered from different types of sensors simultaneously. A robot partner should have human-like intelligence and cognitive capabilities to co-exist, communicate, and interact with people. According to the relevance theory proposed by Sperber and Wilson [109], each person has his or her own cognitive environment, and the communication between two people is restricted by their cognitive environments. Furthermore, an important role of utterances or gestures is to make a person pay his or her attention to a specific target object, person, or event. As a result, the cognitive environment of the other can be enlarged and shared with each other. The shared cognitive environment is called a mutual cognitive environment. A human-friendly robot partner also should have such a cognitive environment, and the robot partner should keep updating the cognitive environment according to the current perception through the interaction with a person in order to realize natural communication. This means that such robot partner should deal with a large size of environmental data. Therefore, the environment surrounding people and robots should have a structured platform for gathering, storing, transforming, and providing information. Such environment is called Informationally Structured Space (ISS). A robot can directly receive and share the environmental information through a wireless network without any measurement process. Information gathered in the environment is transformed by robot partner and sensor network device into a qualitative information. This information is uploaded to ISS using its own rule and reversibly can transform the information downloaded from ISS to measurement data by its own rule. This sharing information process within the environment, which realizes natural communication between human and robot. Figure 2.2 illustrates the concept of informationally structured space. In order to understand ISS deeply, I will discuss the ISS concerning life hub and robot partner in detail.

The information in ISS can be categorized into four classes from the viewpoint of access or measurement methods such as (1) personal information, (2) environmental information, (3) intelligent kernel information (sensor, robot, etc.), and (4) Internet information. (4) The Internet information (Fig. 2.3) is composed of public information and private information, while (2) the environmental information is composed of stable information (object, place, etc.) and dynamic information (condition, state, etc.) (1) The personal information is composed of three types of information; stable information (attribute information, medical information, etc.), dynamic information of raw level data (biomedical signal, human motion, and human location measurement data), and dynamic information of state and condition (currently human behavior state, health condition, etc.) (3) The intelligent kernel information consists of sensor, robot, etc. Through the intelligent kernel, we can acquire information from as well as provide information for physical space and cyberspace. The intelligent kernel can be treated as a hub connecting human to environment. In this paper, the intelligent kernel is called Life Hub.

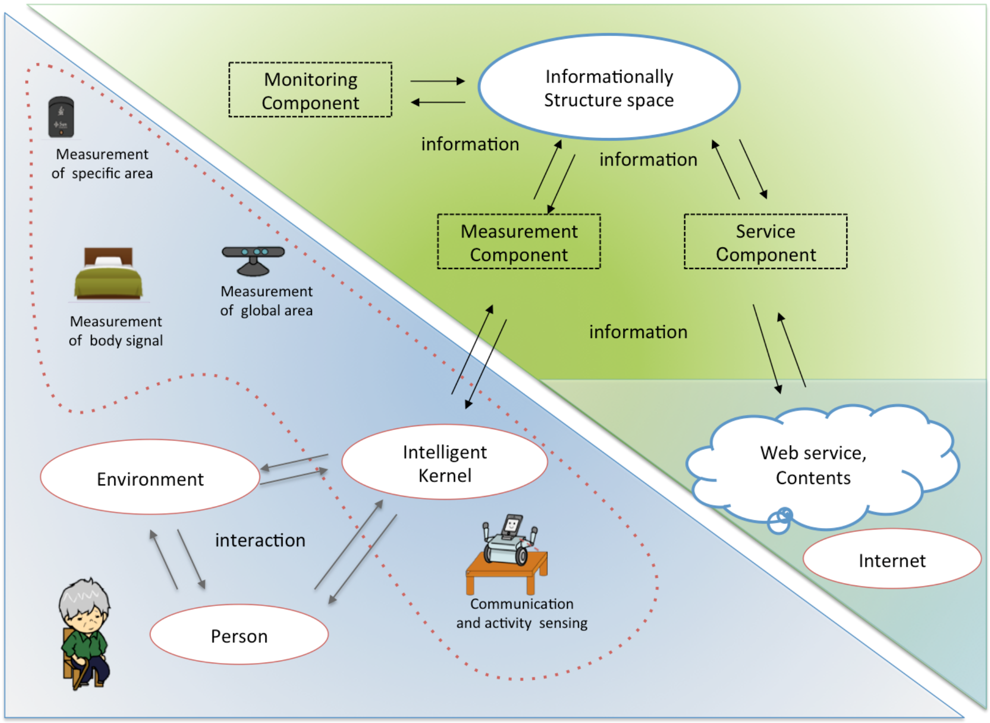



Figure 2.2: Concept of informationally structured space

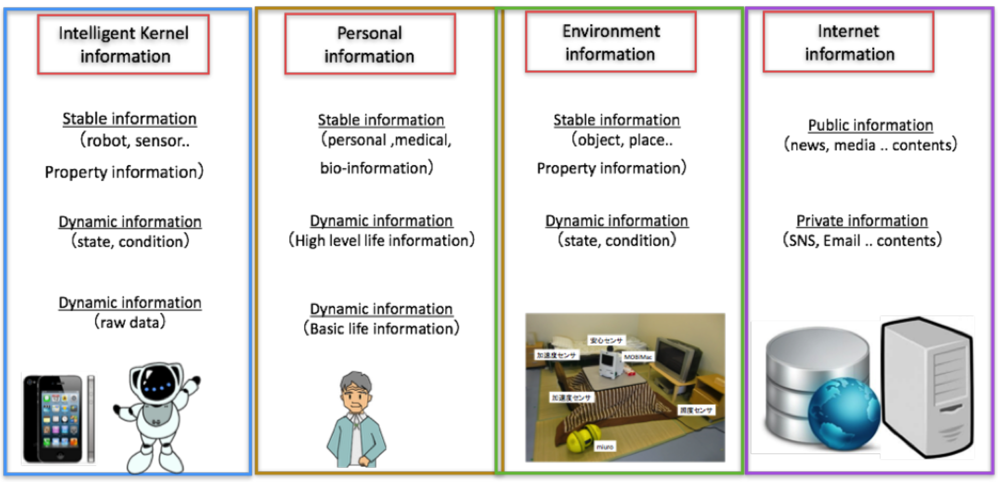



Figure 2.3: Information of informationally structured space

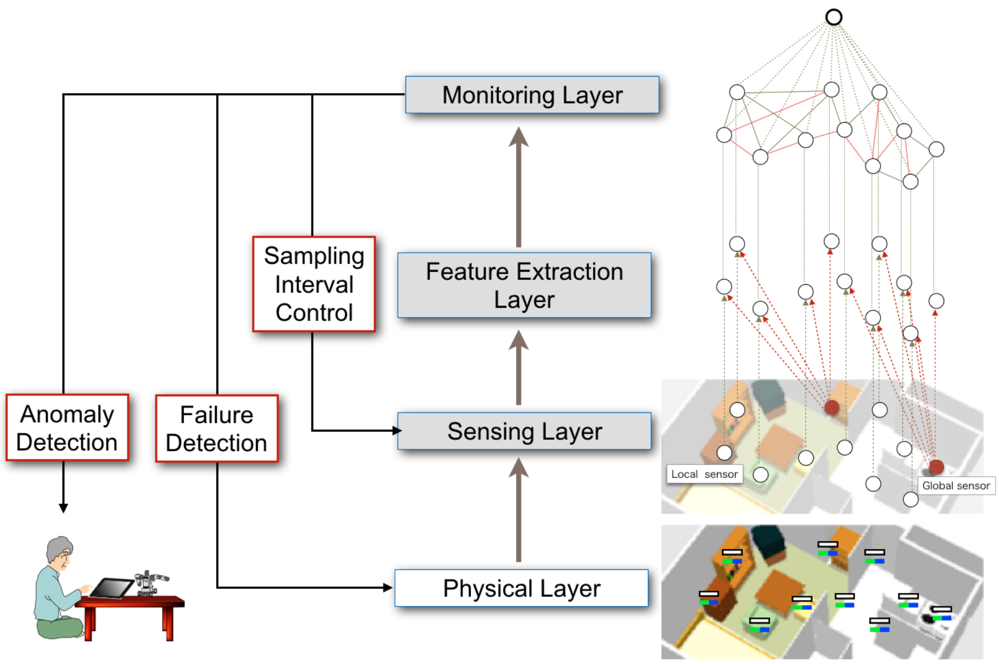



Figure 2.4: Layers of informationally structured space

### 2.1.3 Design of Informationally Structured Space

Here, in order to clarify the design guidelines, I will discuss the informationally structured space which consists of three layers : 1) sensing layer, 2) feature extraction layer, and 3) monitoring layer (Fig. 2.4).

1) In sensing layer, each sensor node measures human and environmental states while changing its sampling interval. 2) In feature extraction layer, human behavior estimation is conducted according to sensing data from the sensor node and robot. 3) In monitoring layer, spatiotemporal changing patterns are extracted from time-series of behaviors. This layer also generates individual daily life model and detects sensor failure as well as human anomaly life pattern. Additionally, the monitoring layer is able to control sampling interval of sensor node according to the change of the environmental condition.

## 2.2 Informationally Structured Space for Life Hub

The word "life hub" is the extended concept of "digital hub" explained by the late Steve Jobs. He explained that Macintosh in a short time could serve as the Digital Hub that unites those disparate points in our digital life (January 9, 2001). In Life Hub, I unite people with physical and virtual information including (1) personal

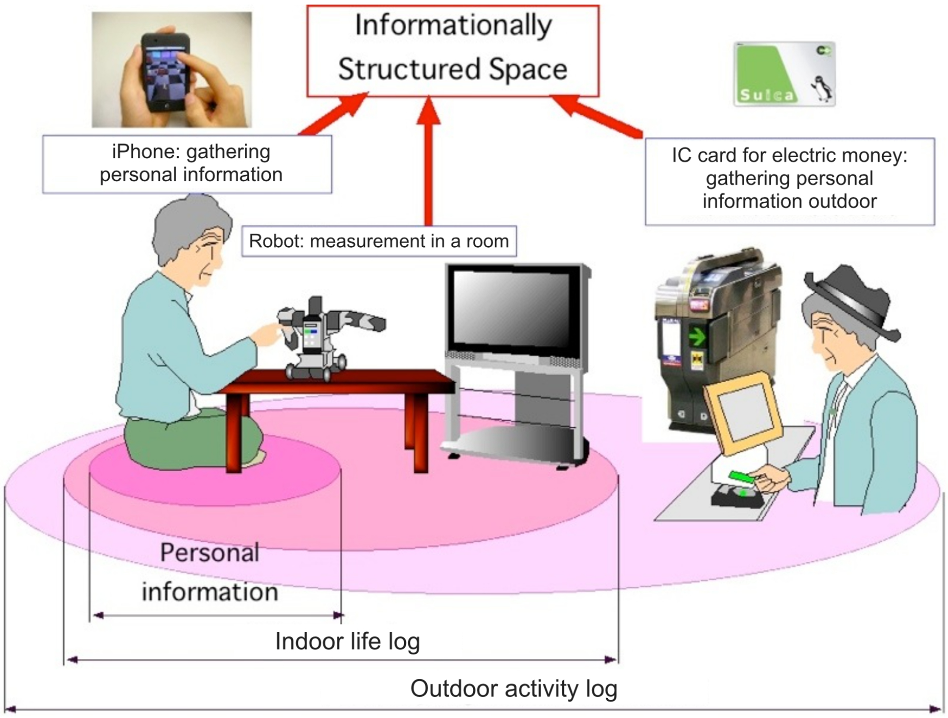



Figure 2.5: Data gathering in Informationally Structured Space

information, (2) environmental information, (3) Internet information, (4) people, (5) place, (6) goods, and (7) events, in addition to real world. These information will be structured and recorded in the database, which can be accessed by the robot partner.

Figure 2.5 illustrates the ISS when gathering personal information to produce daily life log. This figure shows different levels of information supports, such as personal information, indoor life log, and outdoor activity log. Personal information can be gathered by a smart phone. Indoor life log can be created by sensor network. The next level is when the human activity log considers also outdoor information. As shown in Fig. 2.5 a rechargeable IC card is used to trace the shopping and traveling activities of people.

The gathered information can be used on different levels as well. The first level is the information support for family, or in the case of elderly people for the caregivers. The other level is social network in the sense of a local community on the Internet. The last level is the complete information support using Internet.

Connection With Personal Information: Personal information includes personal data such as a bank account, medical history, and daily life log. For example, from the same device application I can easily get personal schedule, address book, etc. A rechargeable IC card, which is almost always carried on the person, can store electric money, and can be used for purchasing and for public transport as a ticket. These tools can help in creating the daily life log of a person.

Connection with Environmental Information: Generally, a robot is equipped with many sensors. However, if the robot wants to express emotions the equipped sensors are not enough to grasp the necessary input information about the surrounding environmental situation. The sensor network devices can measure environmental information, and this environmental information can be applied by the robot partner as input data for example to an emotional model.

Connection with Internet Information: Internet information is used for information service based on personal preference and hobby. I can use various web services such as RSS (Rich Site Summary), yahoo web API for extraction and integration of web information. For example weather, news, local event and traffic information etc.

Connection with People: The robot partner can upload utterances obtained by voice recognition to social networking service (SNS), and download the utterance contents from SNS. Through the robot, elderly people can keep in touch with their family and friend. I propose emotional expression of a robot partner based on SNS contents. I develop a Mobile Ad-hoc Network Simulation and smart device application in order to give a chance for elderly people to join social community and to immediately evacuate them from unsafe places in the case of a disaster. Figure 2.6 displays encountering communication realized by Bluetooth devices for Ad-hoc communication. When two people get in Bluetooth possible communication area, they can share information, chat each other without Wi-Fi and 3G communication environments.

Connection with Place: I can know personal location, moment range, and visited places outdoors from the smart phone GPS log [118]. Kinect sensor and Sun SPOT sensor can provide us with the human's global and specific positions indoor [67, 119]. The robot partner can use the human's motion log to provide the user with the necessary information.

Connection with Objects: Human is related to many objects in the daily life environment. Knowing these relations between objects and human is useful. The communication between the robot and the human can be extended, by using the object related information [130]. For example, when the user takes a cup, the robot will talk about drinking.

Connection with Events: Elderly people should have a chance to join regional or local events. From the web service and social networking service I can get many events information. The robot can recommend information based on personal preference

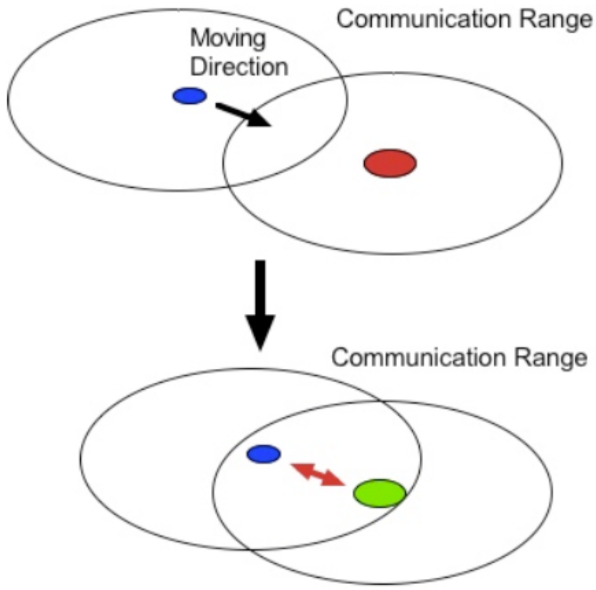



Figure 2.6: Information transfer in the encountering communication

and hobby to join social activities. For example, the robot partner can recommend restaurants, shops, and sightseeing spots to users [107].

## 2.3 Environmental System

I apply the concept of ISS to build the proposed system as shown in Fig. 2.7. This system is divided into an environmental system and a database. The environmental system is composed of a sensor network system, a web system, and a robot system. For measuring the environment and the human condition information, I apply the sensor network system. News or weather reports can be extracted using the web system, and the robot system is utilized for conducting directional interaction with the human while saving the human communication history into the database. The data processing work flow is as follows. First, the sensor network as an information collecting module gets all information required including environmental and human condition information periodically. These information is stored into the database server as a perceptual input for the emotional model. The emotional model processes all information to realize a robot partner which emotionally interacts with human. The output of the emotional model is sent to the robot partner as a signal to be converted into conversation contents, gestural and facial expression. The human reaction as the result of the robot partner's action is used again to update the robot

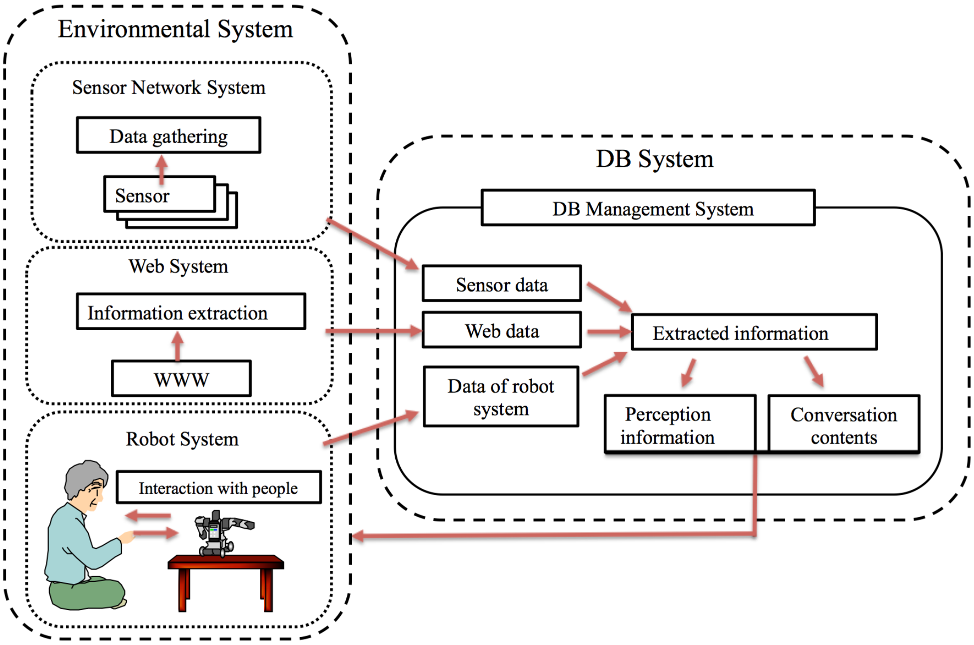



Figure 2.7: The structure of the proposed system. It consists of an environmental system and a DB system.

partner's next action through the sensor network system, database and emotional model.

## 2.4 Robot Partner Systems

### 2.4.1 Robot Partners Using Smart Devices

Recently, various types of smart phone and tablet PC have been developed, and their price is decreasing year by year [82]. Furthermore, the embedded system technology enables to miniaturize such a device and to integrate it with many sensors and other equipment. As a result, we can get a mechatronics device including many sensors, wireless communication systems, GPU and CPU composed of multiple cores with low price. Furthermore, elderly people unfamiliar with information home appliances have started using tablet PC, because touch panels and touch interfaces have been popularized at ticket machines and information services in public areas. Therefore, I started the development project on on-table small sized human-friendly robot partners called iPhonoid and iPadrone based on smart phone or tablet PC to realize information support for elderly people (Fig. 2.8). Since iPhone is equipped with various sensors such as gyro, accelerometer, illumination sensor, touch interface, compass, two cameras, and microphone, the robot itself is enough to be equipped with only cheap range

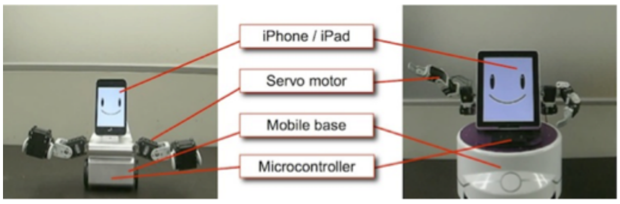
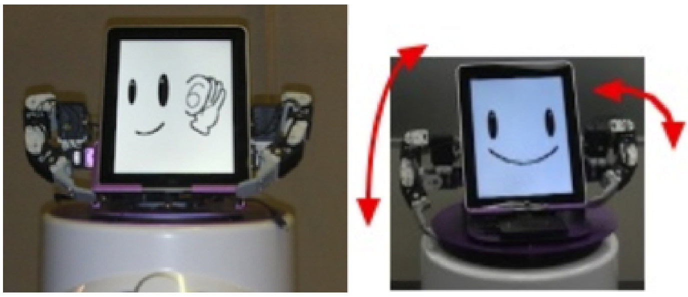



Figure 2.8: Robot partners using a smart phone and a tablet PC



(a) Voice recognition (b) A gesture

Figure 2.9: Robot behaviors for social communication with people

sensors. The mobile base is equipped on the bottom, however the mobile base is not used on the table for safety's sake. In order to control the actuators of a robot partner from the smart phone or tablet PC, wireless LAN and wireless PAN (Bluetooth) can be used in addition to a wired serial communication.

Basically, human detection, object detection, and voice recognition are performed by smart phone or tablet PC. Furthermore, touch interface is used as a direct communication method. The robot partner starts the multi-modal interaction after a smart phone is attached to the robot base. I use touch interface on the smart phone or tablet PC as the nearest interaction with the robot partner. The facial parts are displayed as icons for the touch interface on the display (Fig. 2.9). Since the aim of this study is to realize information support for elderly people, the robot partner provides elderly people with their required information through the touch interface. The eye icon and mouth icon are used for providing the visual information and text information, respectively.

The ear icon is used for direct voice recognition because it is difficult to perform high performance of voice recognition in the daily communication with the robot partner. If the person touches the mouth icon, then the ear icon appears, and the voice recognition starts. The voice recognition is done by Nuance Mobile Developer Program (NMDP). NMDP is a self-service program for developers of iOS and Android application. In this way, the total performance of multimodal communication can be

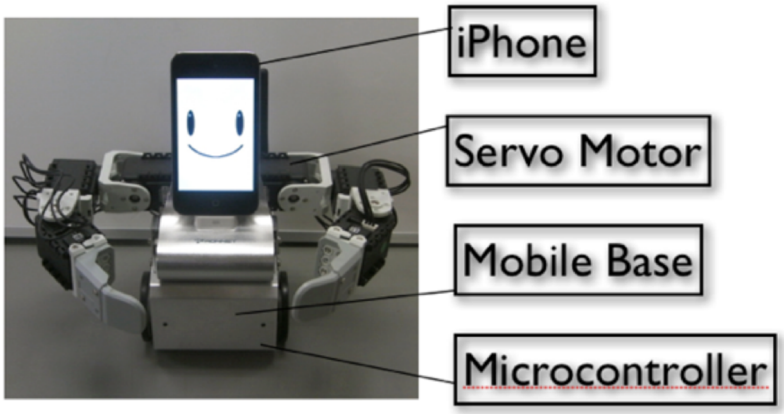



Figure 2.10: iPhonoid

improved by combining several communication modalities of touch interface, voice recognition, and image processing. The conversation system is composed of (A) daily conversation mode, (B) information support mode, and (C) scenario conversation mode [63, 59, 131, 64, 120].

In the previous researches, I have developed and applied various types of robot partners such as MOBiMac, Hubot, Apri Poco, and Miuro to be used as a support system for elderly people [61, 70]. However, since the widespread of smartphone and tablet PC makes their price decreasing along the time, I have been developing a robot partner, which combines smartphone and embedded system into a small, mobile, and economical device. The word "Economical" means that as smartphones are equipped with various sensors like accelerometer, gyro, camera, and microphone, I can decrease the price of the robot partner. In this paper the robot partner will act not only to measure the human condition using touch sensor and voice recognition, but iPhonoid will also process the collected data through sensor network using emotional model to perform a particular action. The architecture of the iPhonoid can be seen in Fig. 2.10.

### 2.4.2 Interaction Modes

I can discuss three different types of robot partners using a smart phone or a tablet PC from the interactive point of view: a physical robot partner, a pocket robot partner, and a virtual robot partner (Fig. 2.11). These modes are not independent, and it can be interacted with the robot partner based on several modes. Interaction modes mean the ways how I interact with the robot partner. I can interact with a physical robot partner by using multi-modal communication like with a human. The interaction is

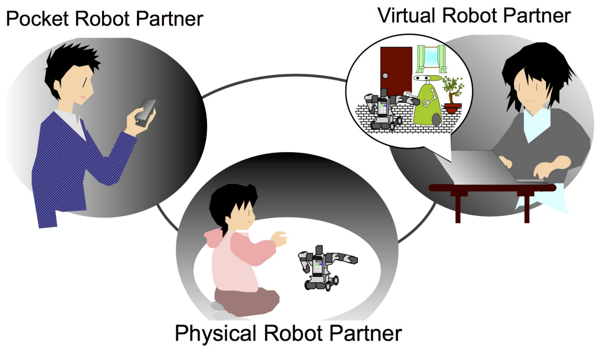



Figure 2.11: Interaction with robot partners from different view points

symmetric. The other one is a virtual robot partner. The virtual robot partner exists in the virtual space in the computer and can be considered as a computer agent, but I can interact with it through the virtual person or robot by immersing him or her in the virtual space. Therefore, the interaction is symmetric. The pocket robot partner has no mobile mechanism, but I can easily bring it everywhere and I can interact with the robot partner by touch and physical interface. Furthermore, the pocket robot partner can estimate the human situation by using internal sensors such as illumination sensor, digital compass, gyro, and accelerometer. The advantage of this device is in the compactness of integrated multi-modal communication interfaces in a single device. Each style of robot partners is different, but the interaction modes depend on each other, and I interact with the robot partner with the same knowledge on personal information, life logs, and interaction rules.

Since the facial expression is used on the display for human interaction (Fig. 2.8, 2.9), the robot partner should estimate the human interaction mode: (a) the physical robot partner mode (attached on the robot base), (b) the pocket robot partner mode (having removed from the robot base), or (c) other mode (on the table, in the bag, or others). In this thesis, 7 interaction modes are defined which will be detailed in Chapter 3.

## 2.5 Database Systems

### 2.5.1 Database Systems

I propose the database structure as illustrated in Fig. 2.12 to realize the informationally structured space. The database structure is divided into eight parts; (A) Human condition, (B) Personal model, (C) Life log, (D) Human behavior, (E) Conversation log, (F) Conversation contents, (G) Web information, and (H) Sensor raw data.

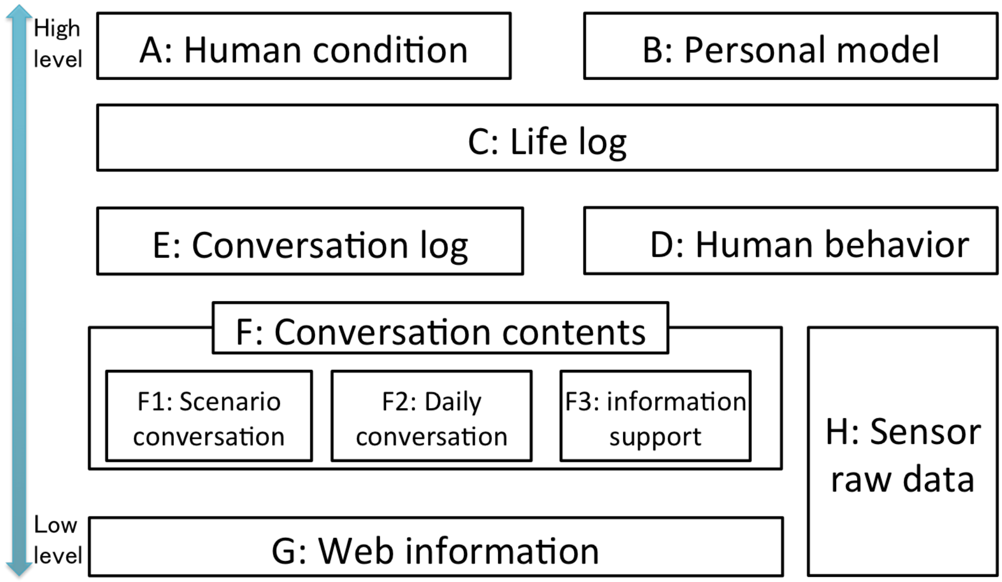



Figure 2.12: Database structure

The sensor raw data (H) is acquired from the measurement by sensor network inside the room and by the smartphone outside the room. Web information (G) is the database that stores the information extracted from the Web to generate sentences in the conversation. Conversation contents (F) provides robot partner with conversation contents to support the communication between the robot partner and the human. The conversation contents is composed of (F1) Scenario conversation, (F2) Daily conversation, and (F3) Information support. Conversation between the robot and the human recorded as Conversation log (E).

Human behavior (D) records human behavior estimation results, which are estimated by active sensing from the robot partner (E) and passive sensing from the sensor network (H). Life log (C) is the database of life log. It is constructed by storing human behavior in time. Human condition (A) and Personal model (B) are statistical analysis result of Life Log. Personal models are the database of personal models, which lead to individual lifestyle and preference extraction. Human states are the database for estimating regular and irregular human state.

### 2.5.2 Web System

As the widespread of tablet-PC and smartphone enforces the development of communication technology, people easily use cloud technology anywhere and anytime. Moreover, using twitter and facebook as social media, people around the world can easily share their opinion, feeling and information by connecting to the Internet. The

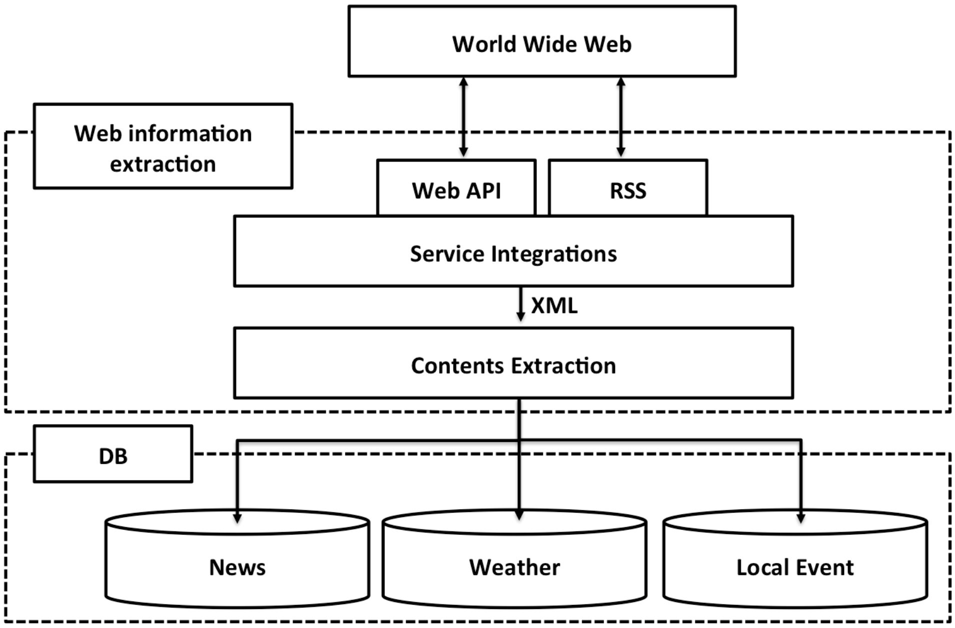



Figure 2.13: Web information extraction system structure

latest news around the world even some gossips about movie stars can be acquired using the RSS (Rich Site Summary) service. Additionally, we can even know which and where the best restaurant around us is, using Google Web API or Yahoo Web API.

However, for the elderly people or people with cognitive decline, information provided by computer or smart devices cannot be acquired, because for them these new technologies are not accustomed yet. Because of this reason, I develop an information support system for these people, which can extract information from Internet. This system is called web system and shown in Fig. 2.13. Web system is composed of web information extraction and database, where the extraction information called meta data can be seen in Fig. 2.14. The web information extraction uses RSS and yahoo web API for acquiring weather and news information in XML file. This extracted data is stored as contents in database to be reused. Table 2.1 shows the example of weather news, while Table 2.2 shows the news example.

## 2.6 Summary

This chapter introduced the basic properties of informationally structured space including information sharing, information representation, information interpretation,

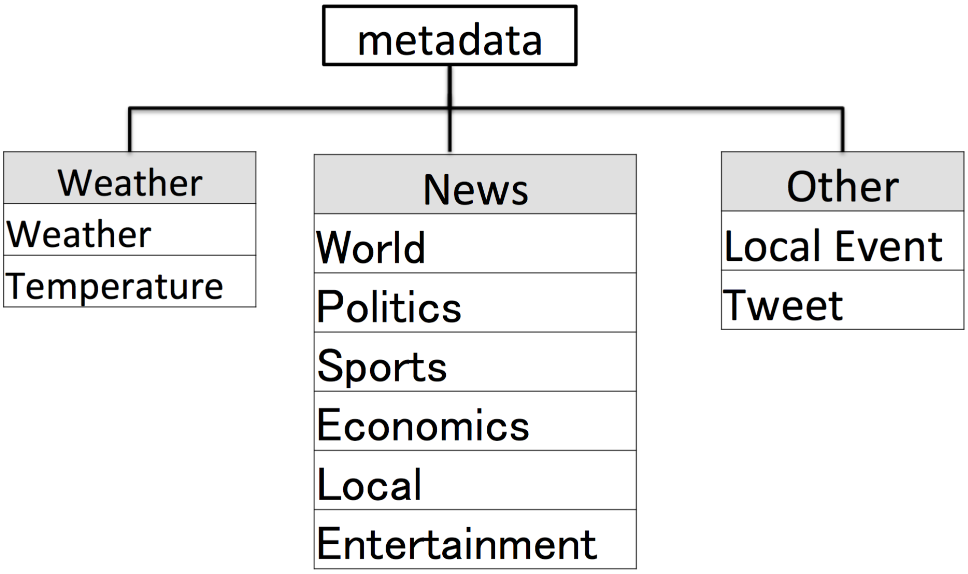



Figure 2.14: Metadata of web information

Table 2.1: Example of weather news database

| Day | Date | Temp (max) | Temp (min) | Weather |
|---|---|---|---|---|
| Wednesday | 140507 | 22 | 16 | Sunny |
| Thursday | 140508 | 24 | 17 | Sunny |
| Friday | 140509 | 25 | 19 | Sunny and Cloudy |

Table 2.2: Example of the latest news database

| No | Topic | Time | Contents |
|---|---|---|---|
| 10001 | World | 140517202514 | Laos government's plane crash cause death of vice prime minister |
| 10002 | World | 140517131908 | China Republic requests payment from Vietnam |
| 10003 | World | 140517153048 | CNN fired editor of 50 plagiarism |

and information operation. I also discussed the methods for connecting various information such as environmental condition, the Internet, and human condition acquired from informationally structured space. At the end of this chapter, I discussed the database system for informationally structured space built as the result of interaction between robot partner and environmental system for estimating human behavior.

# Chapter 3

# Informationally Structured Space for Distributed Sensing System

## 3.1 Sensing Layer in Informationally Structured Space

Various types of concepts and technologies on ubiquitous computing, sensor networks, ambient intelligence, disappearing computing, intelligent spaces, and other fields have been proposed and developed to realize information gathering, life support, safe and secure society [56, 104, 97, 15, 87, 54]. One of the most important issues in the concepts and technologies is the structuralization of information. The structuralization of information means to give qualitative meaning to data and quantitative information in order to improve the accessibility and usability of information. We can obtain huge size of data through sensor networks, however useful, meaningful and valuable information should be extracted from such data.

Nevertheless, we have to design how to use and update ISS in a more useful way for sensor network devices for human estimation and monitoring. Therefore I propose a methodology for human behavior estimation by wireless sensor networks. The measurement system can manage sensor network devices based on ISS according to sensor states and behaviors. I propose a method to monitoring sensor state and human behavior state.

In this chapter the sensors of a smart phone are used for estimating human transport and interaction modes. As the computational power of a smart phone is not so high compared to a standard PC, the computational cost should be reduced as much as possible. Computational intelligence techniques are able to find good compromise between computational cost and solution accuracy. Fuzzy spiking neural network is proposed for estimating human human transport and interaction modes. Additionally, evolution strategy is used for optimizing the parameters of the fuzzy spiking neural network. The performance of estimation is analyzed by several experimental results.

This chapter presents informationally structured space for distributed sensing system. First, I explain the problem of outdoor and indoor measurement, global and local measurement in sensing layer from the viewpoint of environmental condition and from the target of the measurement. Next, I propose human behavior measurement by applying spiking neural network, and structuring and updating the informationally structured space based on the relationship between location and human behavior.

## 3.2 Indoor Human Activity Measurement

### 3.2.1 Indoor Human Activity Measurement System

I conducted the measurement by fusing global and local sensor system. The local sensor system includes acceleration sensor (SensorTag, SunSPOT) and pressure sensor installed on the furniture. However, it is difficult to measure spatial movement of human or furniture by these local sensor systems. On the other hand, the global sensor system which includes Laser Range Finder (LRF) and 3D distant image sensor (Microsoft Kinect) can measure human spatial movement. In conducting global sensor measurement, a furniture becomes an obstacle resulting in dead angle. This problem can be handled using local sensor system to measure the furniture's utilization. By this combination of local and global sensor system, daily human activities can be measured even in the real dynamic world. As a reference, we did not use camera, since camera can cause privacy problem and elderly people also refused to use camera for monitoring.

Sensor characteristics such as measurement interval and data type are registered beforehand as sensor property in the informationally structured space database. During pre-processing time, we acquired the characteristics of sensor data and standardized the data. The system can handle the different property of sensor data even when the sensor device is the same but the maker is different. We used spiking neuron for the unification of various sensor nodes firing methods, after that the standardization is conducted. In other words, the properties of each sensor can influence the pre-processing calculation and threshold from measurement data to spiking neuron input. Only after that we can use spiking neuron's firing information to do estimation. Therefore the estimated human activity is independent on the sensor property.

Table 3.1: Specification of Microsoft Kinect

| | |
|---|---|
| Size | $282 \times 72 \times 72$ [mm] |
| Horizontal field of view | 57 [deg] |
| Vertical field of view | 43 [deg] |
| Physical tilt range | $\pm 27$ [deg] |
| Measuring range | $1.2 - 3.5$ [m] |
| Resolution | $320 \times 240$, $640 \times 480$ [pixel] |
| Frame rate | 30 [fps] |

Table 3.2: Specification of LRF (laser range finder)

| | |
|---|---|
| Size | $50 \times 50 \times 83$ [mm] |
| Measuring area | 60 to 4095 [mm]<br>240 [deg] |
| Accuracy | $\pm 10$ [mm] |
| Scanning time | 100 [ms] |
| Weight | Approx. 160 [g] |
| Interface | USB, RS-232C |

In my previous work, sensor network based on ISS was applied to realize human localization [67]. However, it is difficult to realize it in the real environment because of uncertainty. I propose a general approach for human localization and behavior estimation on ISS. I define relation between sensors and behaviors, relation between behaviors, lifetime, and the behavior's required time on ISS in advance (Fig. 3.3, Fig. 3.4). A given relation structure for sensors and behaviors can improve the measurement system's reliability and generality.

The sensor networks are composed of global measurement system and local measurement system. I apply 6 kinds of sensor device (Fig. 3.3), global measurement sensors: Kinect sensor, LRF (laser range finder) and local measurement sensor: bed sensor, pressure sensor, SensorTag, SunSPOT. The details of the sensor specifications are presented in Tables 3.1, 3.2, 3.3, 3.4, 3.5, 3.6.

### 3.2.2 Indoor Human Activity Measurement Method

Many approaches have been proposed for human behavior modeling so far, for example Hidden Markov Model, Bayesian methods, Support Vector Machines and neural networks are widely applied [17, 27, 3]. However, they require supervised training data, they cannot handle the unknown states well. In my previous work, spiking neural network was applied to localize human, object and sensor device according to local and global sensor specification [67]. The important feature of spiking neural networks

Table 3.3: Specification of bed sensor

| | |
|---|---|
| Sensor range | 0 to 1023 |
| Sensing interval (Min) | 50 [ms] |
| Communication standards | IEEE 802.15.4 (ZigBee) |
| Optical oscillosensor | |
| Size | $86 \times 59 \times 24$ [mm] |
| Weight | 150 [g] |
| Pneumatic sensor | |
| Body of rubber tube size | $7 \times 5$ [mm] |
| Body of sheet size | 0.75 (thickness) [mm] |

Table 3.4: Specification of pressure sensor

| | |
|---|---|
| Sensor range | 0 to 1023 |
| Sensing interval (Min) | 50 [ms] |
| Communication standard | IEEE 802.15.4 (ZigBee) |
| Body of rubber tube size | $7 \times 5$ [mm] |
| Body of sheet size | 0.75 (thickness) [mm] |

Table 3.5: Specification of SensorTag

| | |
|---|---|
| Size | $71.2 \times 36 \times 15.5$ [mm] |
| Battery | Coin cell<br>battery (CR2032) |
| Angular resolution | 0.36 [deg] |
| Sensing interval(Min) | 100 [ms] |
| Communication standard | IEEE 802.15.1 (BLE) |
| Attached sensor | IR temperature sensor<br>Humidity sensor<br>Pressure sensor<br>Accelerometer<br>Gyroscope<br>Magnetometer |
| 3-axis accelerometer range | $\pm 2G$, $\pm 4G$ or $\pm 8G$ |

Table 3.6: Specification of SunSPOT

| | |
|---|---|
| Size | $41 \times 23 \times 70$ [mm] |
| Weight | 54 [g] |
| 3-axis accelerometer range | 2G/3G |
| Light sensor range | $0 - 750$ [raw reading from 1x] |
| Battery | 720 [mAh] lithium-ion battery |
| OS | Squawk VM |
| Communication standard | IEEE 802.15.4 (2.4GHz) |


Figure 3.1: Experimental room

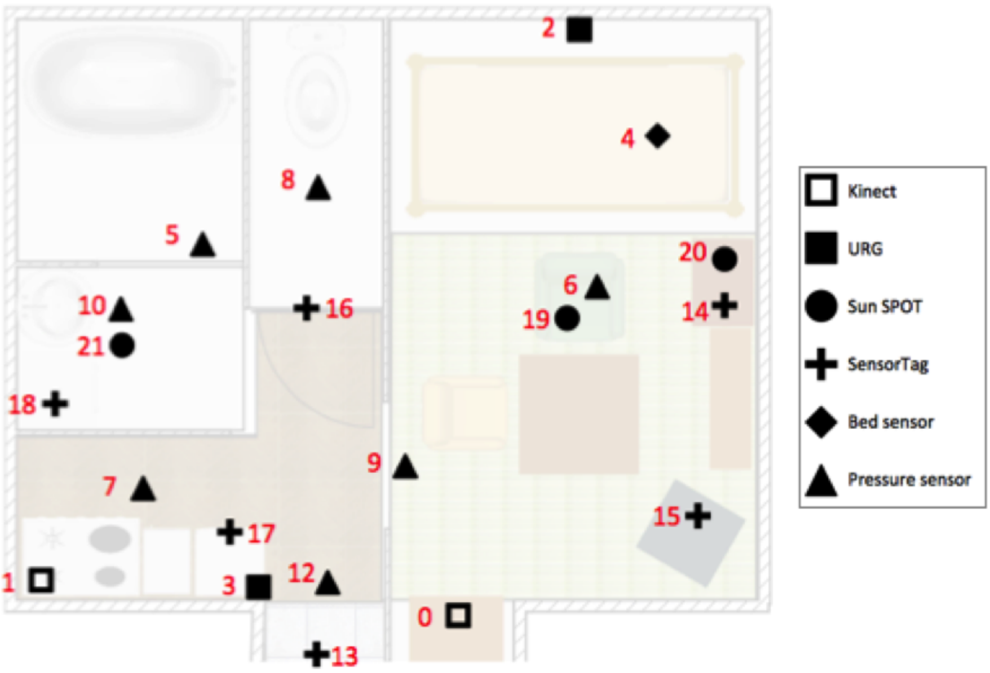



Figure 3.2: Sensor position

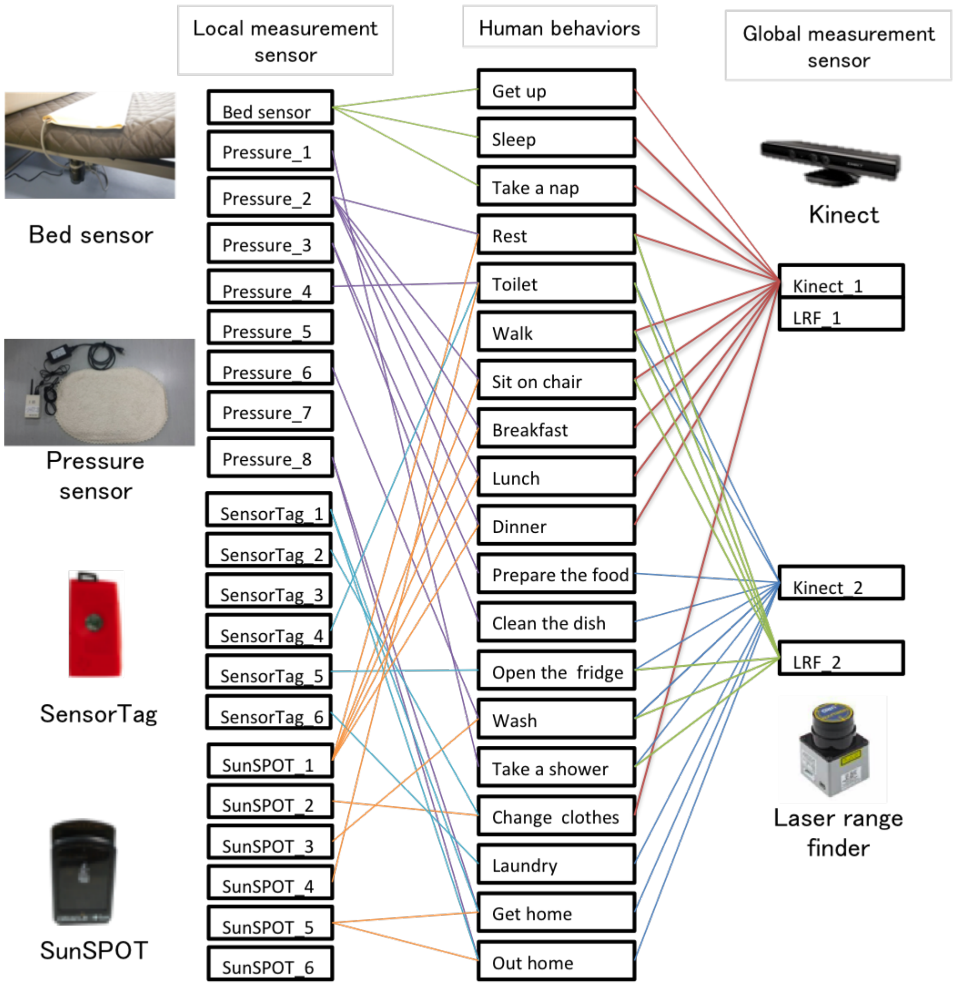



Figure 3.3: Relation between sensors and behaviors

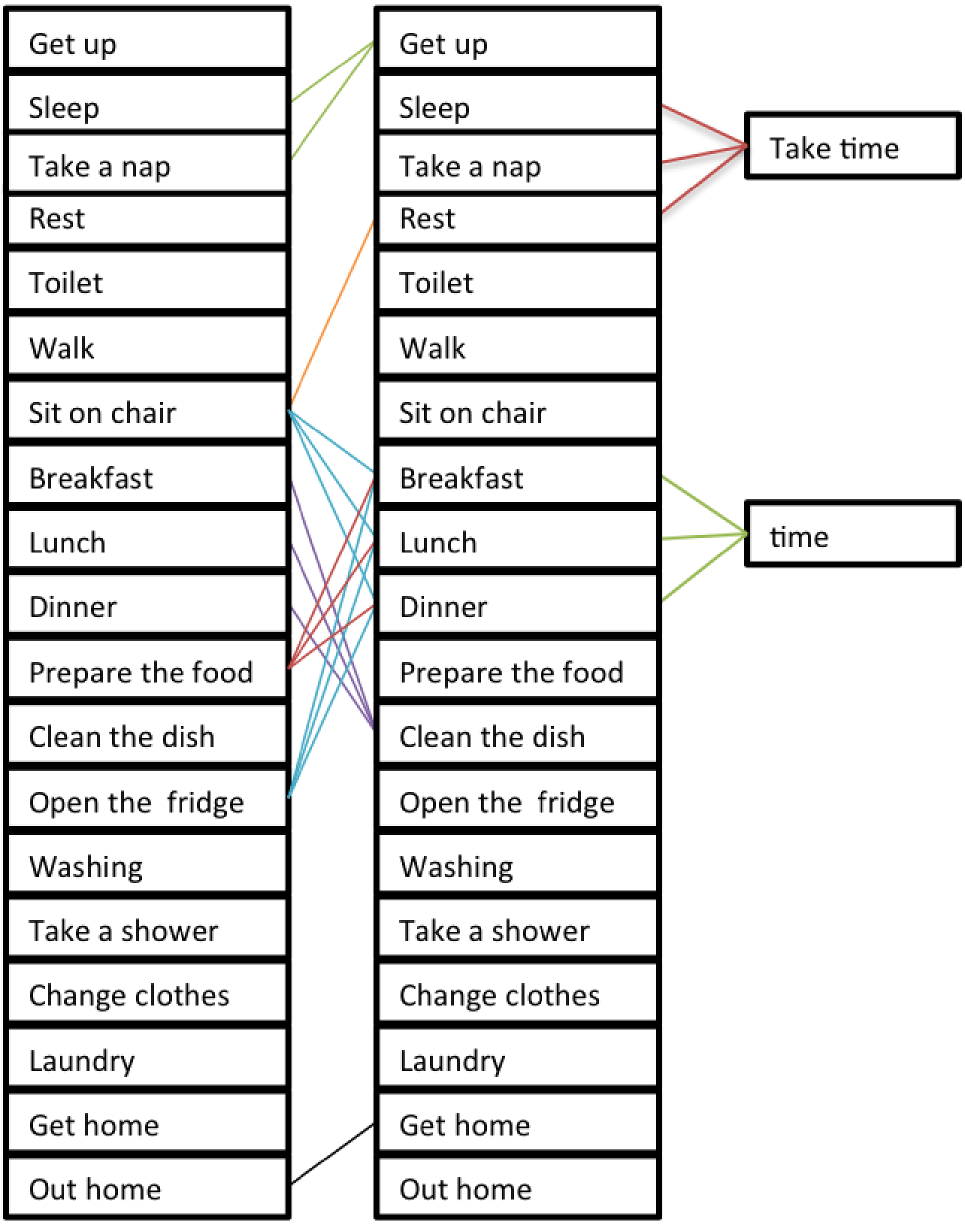



Figure 3.4: Relation between behaviors

is the capability of temporal coding and resistance to noise [79]. In order to reduce the computational cost, a simple spike response model is used. The internal state $h_i(t)$ of the $i$-th spiking neuron at the discrete time $t$ is $h_i(t) = \tanh(h_i^{syn}(t)+h_i^{ext}(t)+h_i^{ref}(t))$, where $h_i^{syn}(t)$ includes the pulse outputs from the other neurons, $h_i^{ref}(t)$ is used for representing the refractoriness of the neuron, $h_i^{ext}(t)$ is the input to the $i$-th neuron from the external environment.

I apply the SNN for human behavior estimation based on measured data of the sensor network. Basically, each furniture or equipment is attached with a sensor. If its measured value had a large change, then the difference from the base value is used as the input to a spiking neuron as follows:

$$h_i^{ext}(t) = \sum_{j=1}^{N_s} r_{j,i} \cdot w_{j,i}^t(t) \cdot S_j(t) \tag{3.1}$$

where $S_j(t)$ is the sensory input, $N_s$ is the number of sensors, $r_{j,i}$ is the relationship between sensors and behaviors, and $w_{j,i}^t(t)$ is the weight on temporal behavior calculated by

$$w_{j,i}^t(t) = v_{j,i} \cdot \exp\left(-\frac{(t-a_{j,i})^2}{b_{j,i}}\right) \tag{3.2}$$

where $a_{j,i}$, $b_{j,i}$ and $v_{j,i}$ are the central value, the width, and the height of Gaussian function.

$h_i^{syn}(t)$ contains the output pulses from other neurons calculated by

$$h_i^{syn}(t) = \gamma^{syn} \cdot h_i(t-1) + \sum_{j=1, j\neq i}^{N} w_{j,i} \cdot h_j^{psp}(t-1), \tag{3.3}$$

where $w_{j,i}$ is a weight coefficient from the $j$-th to the $i$-th neuron; $h_j^{PSP}(t)$ is the presynaptic action potential (PSP) approximately transmitted from the $j$-th neuron at the discrete time $t$; $N$ is the number of neurons. When the internal state of the $i$-th neuron is larger than the predefined threshold $\theta$, a pulse $p_i(t)$ is outputted with value 1, otherwise $p_i(t)$ is 0. In case when the $i$-th neuron is firing, its refractoriness $h_i^{ref}(t)$ is discounted by a discount rate $\gamma^{ref}$ and reduced by $R$, otherwise it is only discounted by $\gamma^{ref}$.

The spiking neurons are interconnected, and the presynaptic spike output is transmitted to the connected neuron according to the PSP with the weight connection. The $i$-th neuron PSP ($h_i^{psp}$) is excitatory if the weight parameter $w_{i,j}$ is positive. When the $i$-th neuron is firing, $h_i^{psp}(t)$ is 1, otherwise $h_i^{psp}(t)$ is discounted by a discount rate

Table 3.7: Specification of iPhone's sensors

| | Accelerometer | Gyro | Attitude | GPS |
|---|---|---|---|---|
| Acquired data | x, y, z | x, y, z | pitch, roll, yaw | latitude, longitude |
| Range of data | $\pm 2.3G$ | x, y, z | $\pm 90$, $\pm 180$, $\pm 180$, | $\pm 90$, $\pm 180$, |
| Sampling time | 100ms | 100ms | 100ms | 100ms |
| Recording time | 500ms | 500ms | 500ms | 500ms |

$\gamma^{psp}(0 < \gamma^{psp} < 1)$. If the condition $0 < h_j^{PSP}(t-1) < h_j^{PSP}(t)$ is satisfied, the weight parameter is trained based on the temporal Hebbian learning rule as follows [42]:

$$w_{j,i} \leftarrow tanh(\gamma^{wgt} \cdot w_{j,i} + \xi^{wgt} \cdot h_j^{psp}(t-1) \cdot h_i^{psp}(t) \tag{3.4}$$

where $\gamma^{wgt}$ is a discount rate and $\xi^{wgt}$ is the learning rate.

## 3.3 Outdoor Human Activity Measurement

### 3.3.1 Sensory Inputs from Smart Devices

In outdoor measurement, different with indoor measurement, smart watch, smart band, and smart phone are used as sensor devices. However, one issue is the trade-off between energy consumption and measurement accuracy. In outdoor measurement, we also integrate two types of measurement; the global measurement (GPS) and local measurement (Accelerometer). This research estimates the smartphone state by consider the trade-off between energy consumption and measurement accuracy. Thus, I combine the global and local measurement to estimate human transport modes.

Several sensory data measured by a smart phone can be used. As depicted in Fig. 4, since iOS 4.0 there has been a Core Motion framework to deal with obtaining sensory data. The acceleration of human movement is calculated by using a highpass filter for the measured data by the accelerometer. The angular velocity is calculated by using a low-pass filter for the measured data by the gyro sensor. The iPhone's attitude is calculated by the measured data of accelerometer, gyroscope,and magnetometer. The specification of the measured data is presented in Table 1. The data are updated in every 100 ms.

The acceleration can be calculated as Eq. (3.5):

$$a(t) = \sqrt{a_x(t)^2 + a_y(t)^2 + a_z(t)^2} \tag{3.5}$$

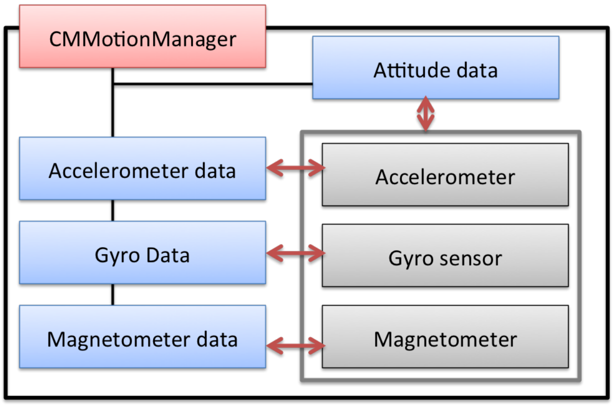



Figure 3.5: iOS core motion framework for obtaining sensory data

where $a_x(t)$, $a_y(t)$, and $a_z(t)$ are the components of the acceleration in the unit directions at time $t$.

The angular velocity is computed as Eq. (3.6):

$$\omega(t) = \sqrt{\omega_x(t)^2 + \omega_y(t)^2 + \omega_z(t)^2} \tag{3.6}$$

where $\omega_x(t)$, $\omega_y(t)$, and $\omega_z(t)$ are the angular velocities at time t in the X-axis, Y-axis, and Z-axis, respectively.

The iPhone's attitude is (Eq. (3.7)):

$$\theta_x(t) = \left|\frac{\theta_x(t)}{90^\circ}\right|, \ \theta_y(t) = \left|\frac{\theta_y(t)}{180^\circ}\right|, \ \theta_z(t) = \left|\frac{\theta_z(t)}{180^\circ}\right| \tag{3.7}$$

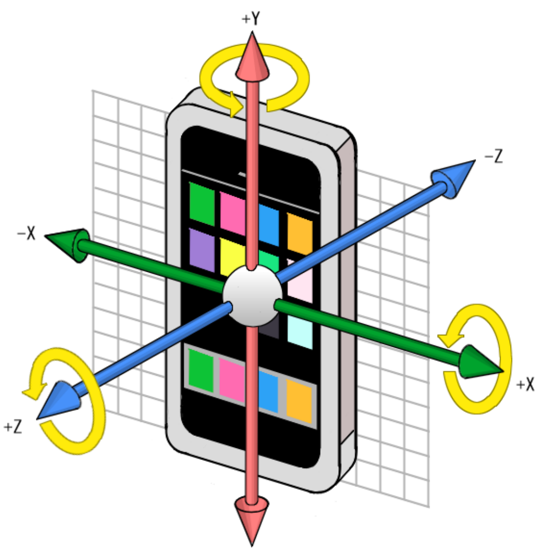
where $\theta_x(t)$, $\theta_y(t)$, and $\theta_z(t)$ are the pitch, roll, and yaw Euler angles at time $t$, respectively.

The movement distance by GPS is:

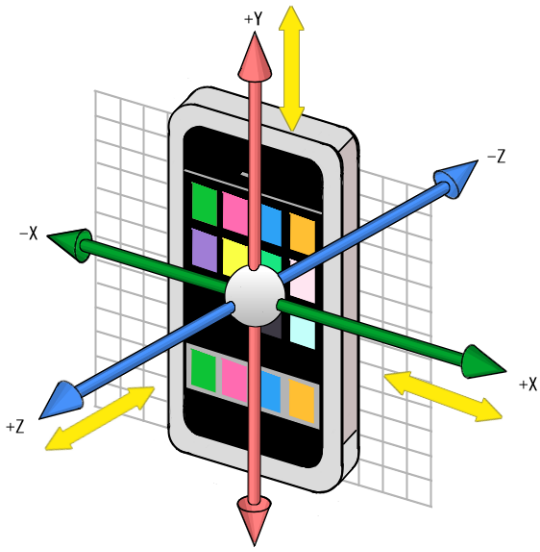
$$s(t) = \sqrt{(x_t - x_{t-1})^2 + (y_t - y_{t-1})^2} \tag{3.8}$$

where $x_t$, and $x_{t-1}$ are the latitude components and $y_t$, and $y_{t-1}$ are the longitude components at time $t$ and $t-1$, respectively. The altitude component of the GPS data is not considered here because of the different scale of that component.

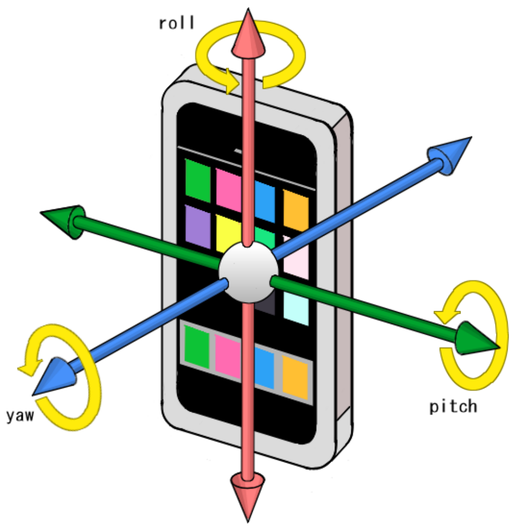
Since the measured data includes noise, some smoothing functions have to be used. Two different kinds of weighted moving averages are applied as presented in Eq. (3.9) and Eq. (3.10), where $d$ is the window length. In Eq. (3.9) the weights increase from the smallest weight at time $(t-d+1)$ to the current data point at time $t$. In Eq. (3.10) the weights increase first, from the smallest weight at time $(t-d/2)$




(a) Angular velocity data of iPhone




(b) Acceleration data of iPhone




(c) iPhone's attitude data

Figure 3.6: iOS core motion framework for obtaining sensory data

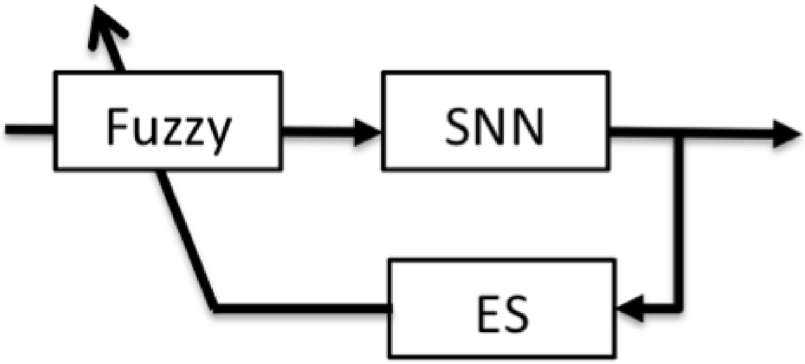


Figure 3.7: Fuzzy spiking neural network

to the current point at time t, after that they decrease till time $(t + d/2 - 1)$. In Eq. (3.9) and Eq. (3.10) $j$ indicates the input.

$$\bar{x}_j(t) = \frac{1}{\sum_{k=0}^{d-1} \exp\left(-\frac{k}{d}\right)} \sum_{k=0}^{d-1} \exp\left(-\frac{k}{d}\right) x_j(t-k) \tag{3.9}$$

$$\bar{x}_j(t) = \frac{1}{\sum_{k=0}^{d-1} \exp\left(-\frac{|k-0.5\cdot d|}{d}\right)} \sum_{k=0}^{d-1} \exp\left(-\frac{|\ k - 0.5 \cdot d\ |}{d}\right) x_j(t + k - 0.5 \cdot d) \tag{3.10}$$

### 3.3.2 Outdoor Human Activity Measurement Method

#### 3.3.2.1 Fuzzy Spiking Neural Network

I estimate the human interaction modes by fuzzy spiking neurons. One important feature of spiking neurons is the capability of temporal coding. In fact, various types of spiking neural networks (SNNs) have been applied for memorizing spatial and temporal context [5, 35, 36]. A simple spike response model is used in order to reduce the computational cost. In the model the SSN has fuzzy inputs, it is a fuzzy spiking neural network (FSNN) [119, 118, 14]. Figure 3.7 illustrates the FSNN model. Evolution strategy is used to adapt the parameters of the fuzzy membership functions applied as inputs to the spiking neural network. Figure 3.8 depicts the detailed structure of the FSNN model for estimating human interaction modes. The inputs of the FSNN are the sensory data, the outputs are the interaction modes. Figure 3.9 illustrates the detailed structure of the FSNN model for estimating human transport modes. In this case, the inputs are similar as in the other case, however, only 3 sensory data are used. The outputs are the transport modes.

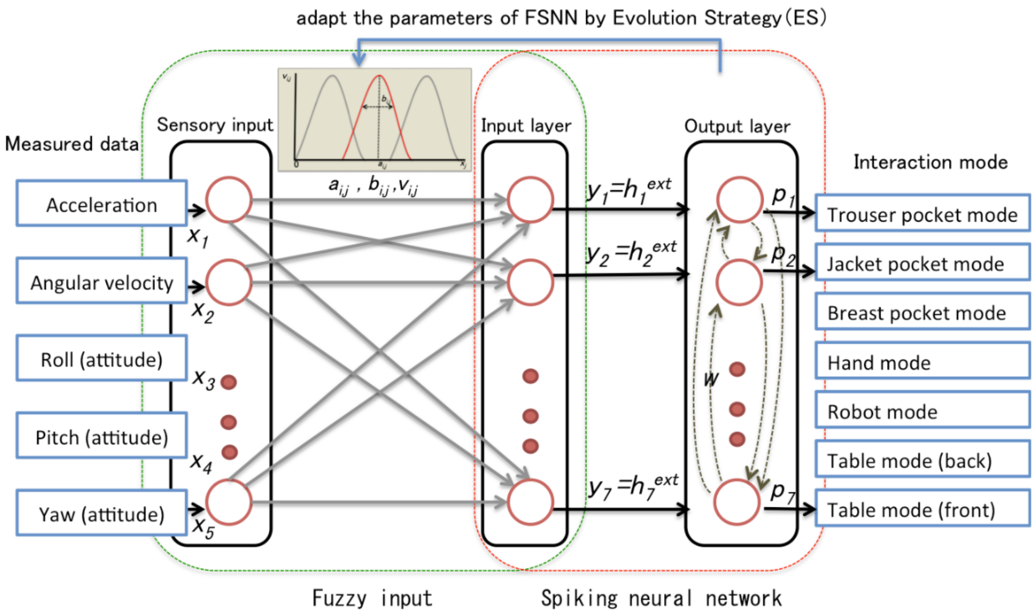



Figure 3.8: Detailed structure of fuzzy spiking neural network for interaction modes

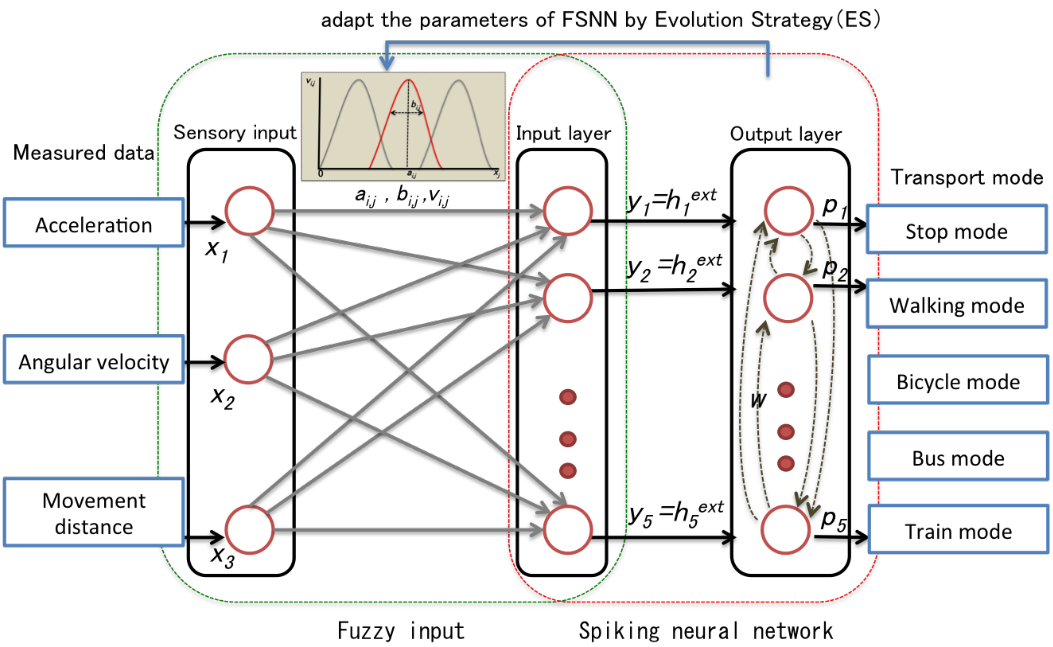



Figure 3.9: Detailed structure of fuzzy spiking neural network for transport modes

On the sensory input fuzzy inference is performed by:

$$\mu_{A_{i,j}}(x_j) = \exp\left(-\frac{(x_j - a_{i,j})^2}{b_{i,j}}\right) \tag{3.11}$$

$$y_i = \prod_{j=1}^{m} v_{i,j} \cdot \mu_{A_{i,j}}(x_j) \tag{3.12}$$

where $a_{i,j}$ and $b_{i,j}$ are the central value and the width of the membership function $A_{i,j}$ and $v_{i,j}$ is the contribution of the $j$-th input to the estimation of the $i$-th human interaction mode. The result of fuzzy inference, $y_i$, will be the input of the spiking neurons.

The membrane potential, or internal state $h_i(t)$ of the $i$-th spiking neuron at the discrete time $t$ is given by:

$$h_i(t) = \tanh(h_i^{syn}(t) + h_i^{ext}(t) + h_i^{ref}(t)) \tag{3.13}$$

where $h_i^{syn}(t)$ includes the pulse outputs from the other neurons, $h_i^{ref}(t)$ is used for representing the refractoriness of the neuron, $h_i^{ext}(t)$ is the input to the $i$-th neuron from the external environment. The hyperbolic tangent function is used to avoid the bursting of neuronal fires.

The external input, $h_i^{ext}(t)$ is calculated based on the fuzzy inference in Eq. (3.11) and Eq. (3.12), and it is equal to $y_i$ as illustrated in Figs. 3.8 and 3.9, thus:

$$h_i^{ext}(t) = \prod_{j=1}^{m} v_{i,j} \cdot \exp\left(-\frac{(x_j - a_{i,j})^2}{b_{i,j}}\right) \tag{3.14}$$

Furthermore, $h_i^{syn}(t)$ indicates the output pulses from other neurons presented by dashed arrows in Figs. 3.8 and 3.9 in the output layer:

$$h_i^{syn}(t) = \sum_{j=1, j\neq i}^{N} w_{j,i} \cdot h_j^{psp}(t-1), \tag{3.15}$$

where $w_{j,i}$ is a weight coefficient from the $j$-th to the $i$-th neuron; $h_i^{PSP}(t)$ is the presynaptic action potential (PSP) approximately transmitted from the $j$-th neuron at the discrete time $t$; $N$ is the number of neurons. When the internal state of the $i$-th neuron is larger than the predefined threshold, a pulse is outputted as follows:

$$p_i(t) = \begin{cases} 1 & \text{if } h_i(t) \geq q^{pul} \\ 0 & otherwise \end{cases} \tag{3.16}$$

where $q^{pul}$ is a threshold for firing. The outputs of FSNN are the $p_i$ values as presented in Figs. 3.8 and 3.9. Thus, the output is the interaction mode $i$ which for $p_i = 1$. If there are more than one neuron with output pulse 1 (i.e.,$\exists i, j$ , $i \neq j$ $p_i = p_j = 1$), then the output will be that one which fired in the previous step, $t-1$. If none fired, then the output neuron will be selected randomly.

Furthermore, $R$ is subtracted from the refractoriness value as follows:

$$h_i^{ref}(t) = \begin{cases} \gamma^{ref} \cdot h_i^{ref}(t-1) - R & \text{if } p_i(t-1) = 1 \\ \gamma^{ref} \cdot h_i^{ref}(t-1) & otherwise \end{cases} \tag{3.17}$$

where $\gamma^{ref}$ is a discount rate and $R > 0$. The spiking neurons are interconnected, and the presynaptic spike output is transmitted to the connected neuron according to the PSP with the weight connection. The PSP is calculated as follows:

$$h_i^{psp}(t) = \begin{cases} 1 & \text{if } p_i(t) = 1 \\ \gamma^{psp} \cdot h_i^{psp}(t-1) & otherwise \end{cases} \tag{3.18}$$

where $\gamma^{PSP}$ is the discount rate $(0 < \gamma^{PSP} < 1.0)$. Therefore, the postsynaptic action potential is excitatory if the weight parameter, $w_{j,i}$ is positive, and inhibitory if $w_{j,i}$ is negative. I set $w_{j,i} = -0.2$ in order to suppress the firing chance of other neurons when a given neuron fires.

In the equations describing the three components of the internal state simple functions are used instead of the differential equations proposed in the original model of spiking neural network [36]. By the proposed simple spike response model the computational complexity can be kept at low level.

#### 3.3.2.2 Evolution Strategy for Optimizing Parameters of FSNN

I apply $(\mu + \lambda)$-Evolution Strategy (ES) for the improvement of the parameters of fuzzy spiking neural network in the fuzzy rules. In $(\mu + \lambda)$-ES $\mu$ and $\lambda$ indicate the number of parents and the number of offspring produced in a single generation, respectively [105]. I use $(\mu + 1)$-ES to enhance the local hill-climbing search as a continuous model of generations, which eliminates and generates one individual in a generation. The $(\mu + 1)$-ES can be considered as a steady-state genetic algorithm (SSGA) [115]. As it can be seen in Eqs. (3.11), (3.12), and (3.14) and in Figs. 3.8 and 3.9, a candidate solution will contain the parameters of the fuzzy membership functions which play role in the input layer of the spiking neural network. These parameters are the central value ($a_{i,j}$), the width ($b_{i,j}$), and the contribution value ($v_{i,j}$):

$$g_k = [g_{k,1}, g_{k,2}, g_{k,3}, \cdots g_{k,l}] = [a_{k,1,1}, b_{k,1,1}, v_{k,1,1}, \cdots v_{k,n,m}] \tag{3.19}$$

where $n$ is the number of human interaction modes; $m$ is the number of inputs; $l = n \cdot m$ is the chromosome length of the $k$-th candidate solution. The fitness value of the $k$-th candidate solution is calculated by the following equation:

$$f_k = \sum_{i=1}^{n} f_{k,i} \tag{3.20}$$

where $f_{k,i}$ is the number of correct estimation rates of the $i$-th human interaction mode. The FSNN's each output is compared in the time sequence with the corresponding desired output. If the FSNN's output is the same as the desired output, then it is counted as a matching. The number of matchings for the $i$-th interaction mode is $f_{k,i}$. Thus, the evaluation of the individual and consequently the learning process is performed in supervised manner.

In $(\mu + \lambda)$-ES, only one existing solution is replaced with the candidate solution generated by crossover and mutation. Elitist crossover and adaptive mutation are used. Elitist crossover randomly selects one individual, and generates one individual by combining genetic information between the selected individual and the best individual in order to obtain feasible solutions from the previous estimation result rapidly. The newly generated individual replaces the worst individual in the population after applying adaptive mutation on the newly generated individual. In the genetic operators I use the local evaluation values of the human interaction mode estimation. The inheritance probability of the genes corresponding to the $i$-th rule of the best individual is calculated by:

$$p_i = \frac{1}{2} \cdot (1 + f_{best,i} - f_{k,i}) \tag{3.21}$$

where $f_{best,i}$ and $f_{k,i}$ are the part of the fitness value related to the $i$-th rule ($i$-th genes) of the best and the randomly selected $k$-th individuals, respectively. By Eq. (3.21), the selection probability of the $i$-th genes can be biased from 0.5 to the direction of the better individuals $i$-th genes among the best individual and the kth individual. Thus, the newly generated individual can inherit the $i$-th genes ($i$-th rule) from that individual which the better $i$-th genes has. After the crossover operation, an adaptive mutation is performed on the generated individual:

$$g_{k.h} \leftarrow g_{k,h} + \alpha_h \cdot (1 - t/T) \cdot N(0, 1) \tag{3.22}$$

where $N(0,1)$ indicates a normal random value; $\alpha_h$ is a parameter of the mutation operator ($h$ stands for identifying the three subgroups in the individual related to $a$, $b$, and $v$); $t$ is the current generation; and $T$ is the maximum number of generations.

## 3.4 Experimental Results

### 3.4.1 Experimental Results for Estimation of Human Transport Modes

This section shows comparison results and analyzes the performance of the proposed method. In the spiking neural network there are 3 inputs in the input layer: acceleration, angular velocity, and movement distance. In the output layer there are 5 outputs related to the following five transport modes: (1) stop, (2) walking, (3) taking a bicycle (4) taking a bus, (5) taking a train. The weights of the neural network are randomly initialized with uniform random number between 0 and 1: $0 < w_{j,i} < 1$. The parameters of the neural network are as follows: the temporal discount rate for refractoriness is 0.684, the temporal discount rate for EPSP is 0.72, the threshold for firing is 1.0, and $R$ is 1.0. Two training datasets and one test dataset are used in the experiments.

Fig. 3.10a illustrates the experimental example of the measured user motion outdoors. The gray line is the high-pass filtered data measured by the accelerometer. The blue line depicts the angular velocity calculated by the low-pass filtered data measured by the Gyro sensor. The green line is the moving distance calculated by the GPS. The second part of Fig. 3.10a shows the target output (blue line) and the estimated output by the FSNN (red line). The number of spiking neurons is 5. These neurons are used for measuring the five transport modes.

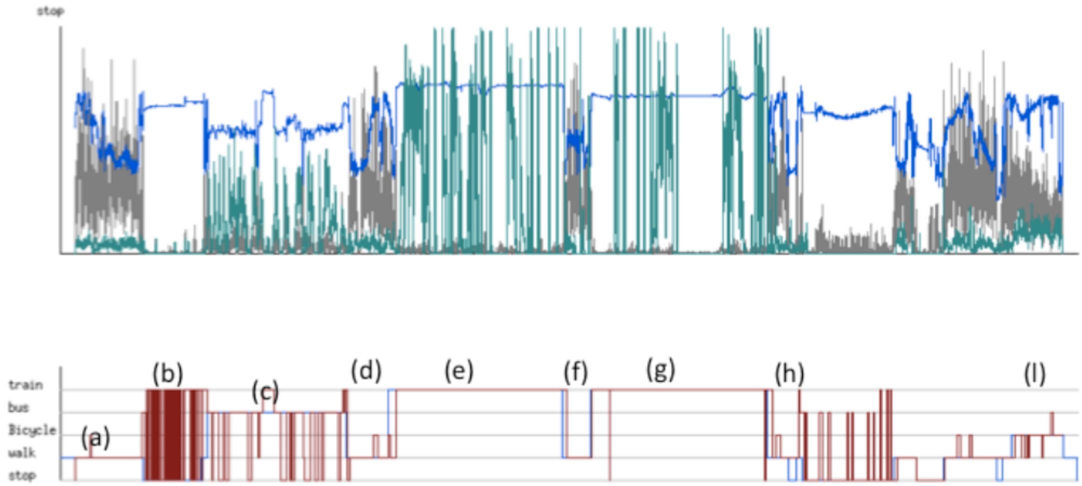
In the first experiment the sensors raw data are used as input to the FSNN (Fig. 3.10a, 3.10b and 3.10c). Fig. 3.10a shows experimental results by using the raw data of the first training dataset. In this case the person walks to the bus stop (a), and waits for the bus (b). Thereafter he/she takes a bus (c) and gets off the bus to wait for a train (d). After taking the train (e), the person gets off the train to wait for another train (f), and takes a train again (g). The person gets off the train, walks and stops several times (h), and finally he/she takes a bicycle and goes home (l). The number of fitting data is 10956 from 13788 training data, the running time is 38 ms.

Fig. 3.10b illustrates experimental results by using the raw data of the second training dataset. In this case the person goes out, walks toward the train station (a) and takes a train (b). After that the person gets off the train, walks toward the bus stop (c) next he/she takes a bus (d), and finally gets off the bus and walks (e). The number of fitting data is 2406 from 5135 training data, the running time is 18 ms. The total number of fitting data based on the two training datasets is 13362 from 18923 data, the running time is 56 ms, and the fitting rate is 70.6%. Fig. 3.10c depicts test data. The number of fitting data is 5658 from 11251 test data. The running time is 33 ms, and the fitting rate is 50.2%. The reason for low fitting estimation rate is the noise in the raw data.

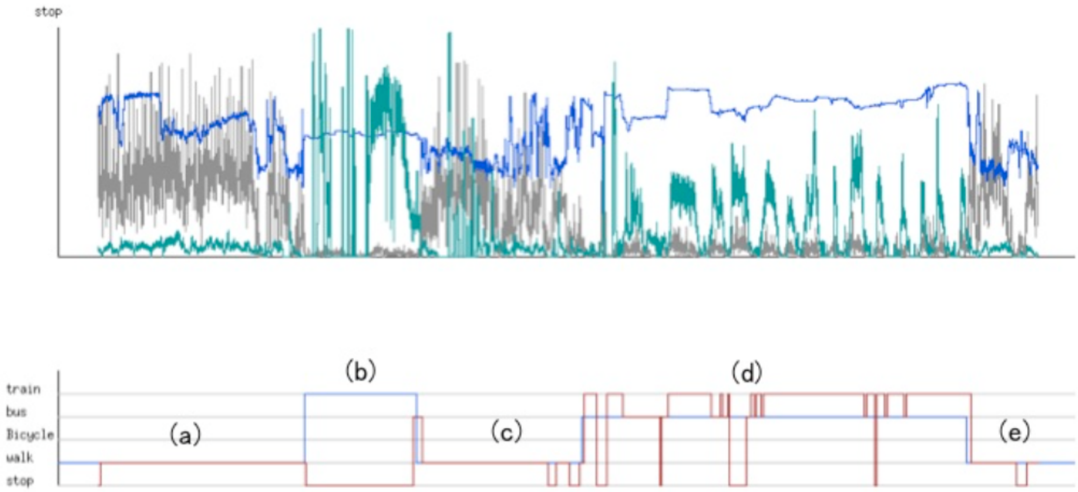
In order to reduce the noise, some smoothing functions have to be used as mentioned in Section 3.3.1. Two different kinds of weighted moving averages are applied. In the second experiment I present results by using the smoothing function described in Eq. 3.9. Figure 3.11a depicts the result for the first training dataset. The number of fitting data is 10944 from 13788 training data, the running time is 40 ms. Fig. 3.11b shows the result for the second training dataset. The number of fitting data is 2688 from 5135 data, and the running time is 13 ms. The total number of fitting data is 13632 from 18923 training data. The running time is 53 ms, and the fitting rate is 72%. Fig. 3.11c displays test data. The number of fitting data is 6792 from 11251. The running time is 31 ms, and the fitting rate is 60.3%.

In the third experiment, I present experimental result by the other smoothing function defined by Eq. 3.10. Fig. 3.12a, 3.12b and Fig. 3.12c illustrate the results for the first and second training dataset and the test dataset, respectively. In case of the first dataset the number of fitting data is 11493 from 13788, and the running time is 40 ms. For the second dataset the number of fitting data is 3081 from 5135, and the running time is 16 ms. The total number of fitting training data is 14574 from 18923. The running time is 56 and the fitting rate is 77%. For the test set the number of fitting data is 8292 from 11251, the running time is 31 ms, and the fitting rate is 73.77%.

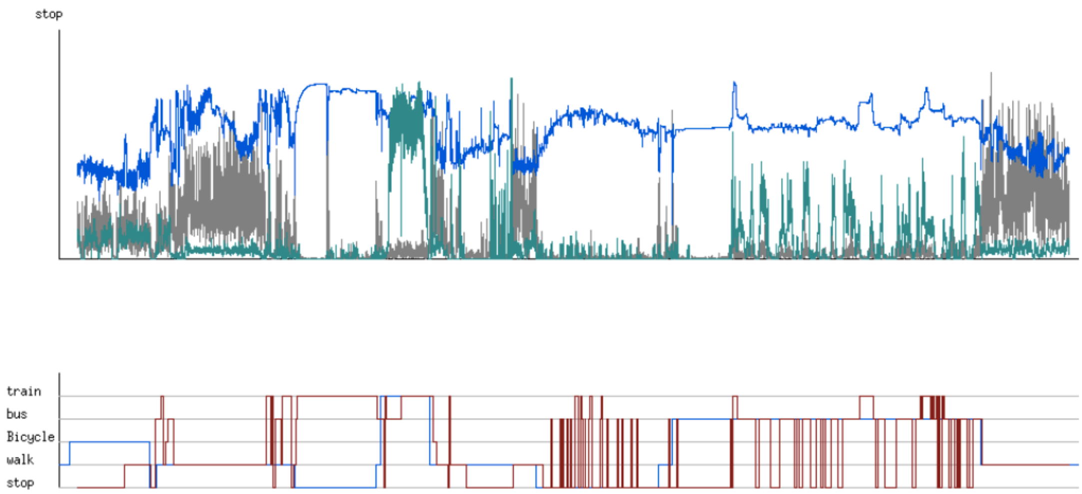
In the fourth experiment I use evolution strategy for optimizing the parameters of the FSNN. For each three previous cases ten simulations were performed because of the stochastic nature of evolution strategy. Fig. 3.13a, 3.13b and 3.13c illustrate the best results by ES applied for the raw data for the first and second training dataset and the test dataset, respectively. The total number of fitting data is 15936 from 18923. The application of ES has an additional computational cost. The running time for the total training phase is 182149 ms. The fitting rate is 84.2%. In case of the test dataset the number of fitting data is 9407 from 11251 test data. The running time is 191762, the fitting rate is 83.6%.




(a) Training dataset 1

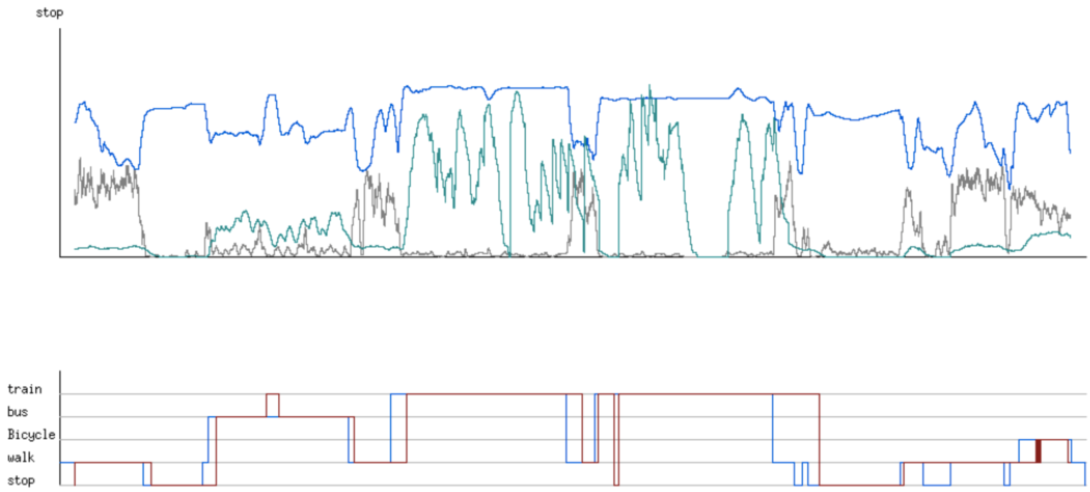



(b) Training dataset 2

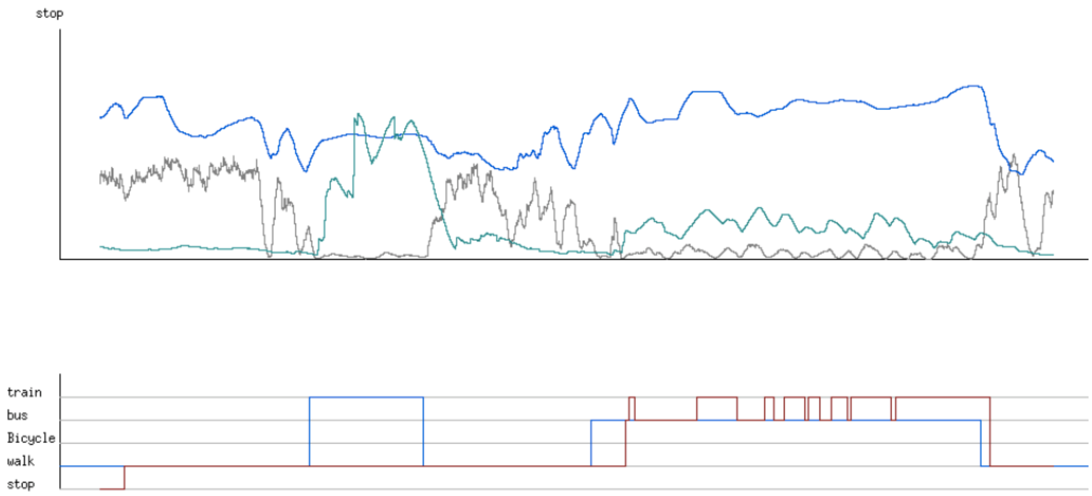



(c) Test data

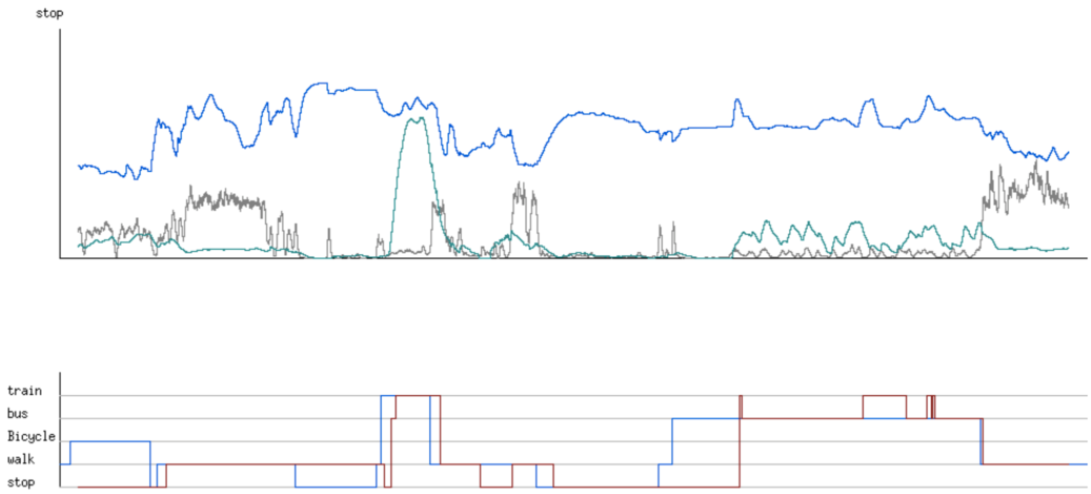
Figure 3.10: Experimental results by using the raw data




(a) Training dataset 1

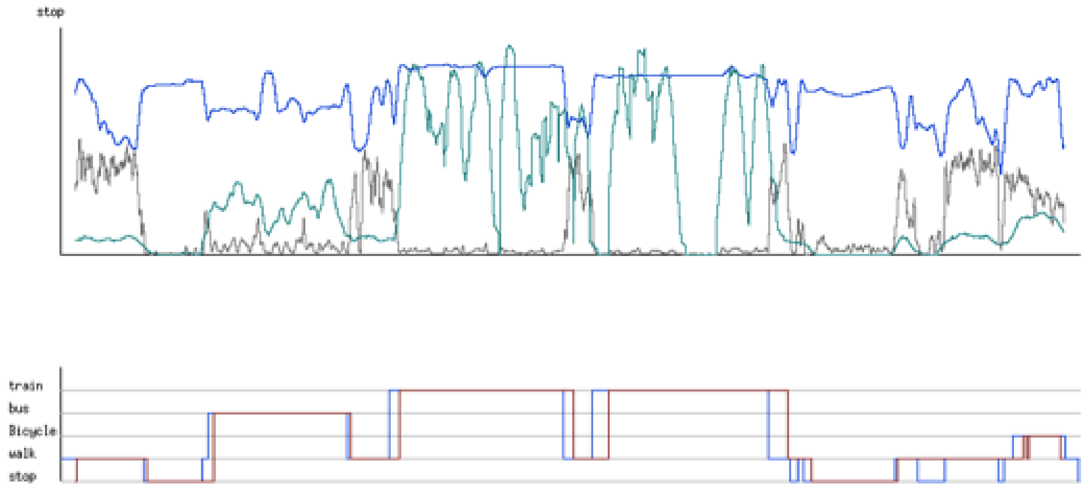



(b) Training dataset 2

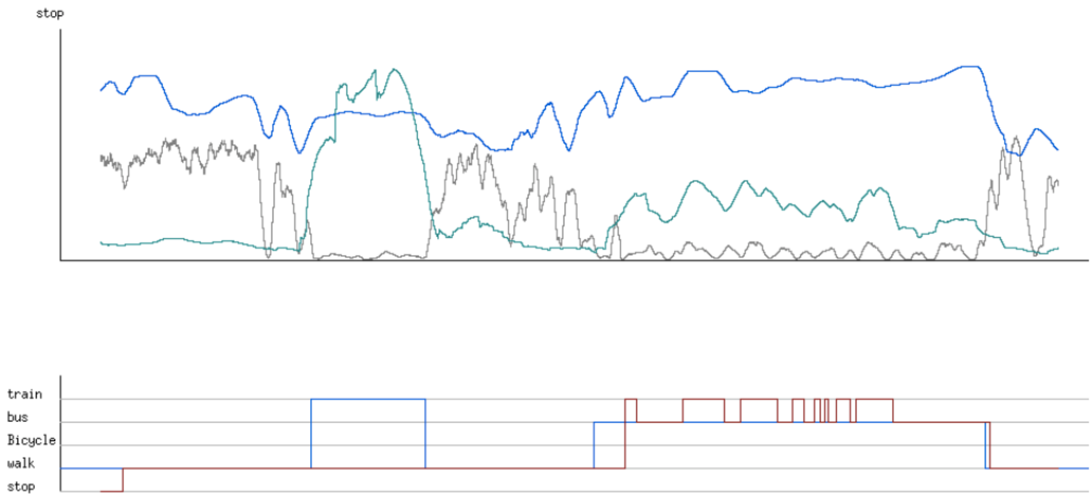



(c) Test data

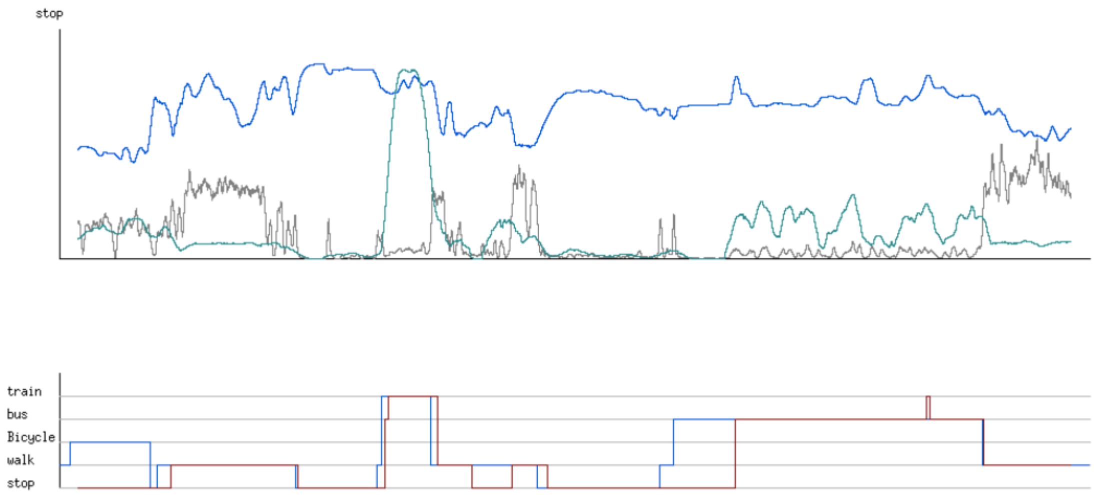
Figure 3.11: Experimental results by using the smoothing function in Eq. 3.9




(a) Training dataset 1

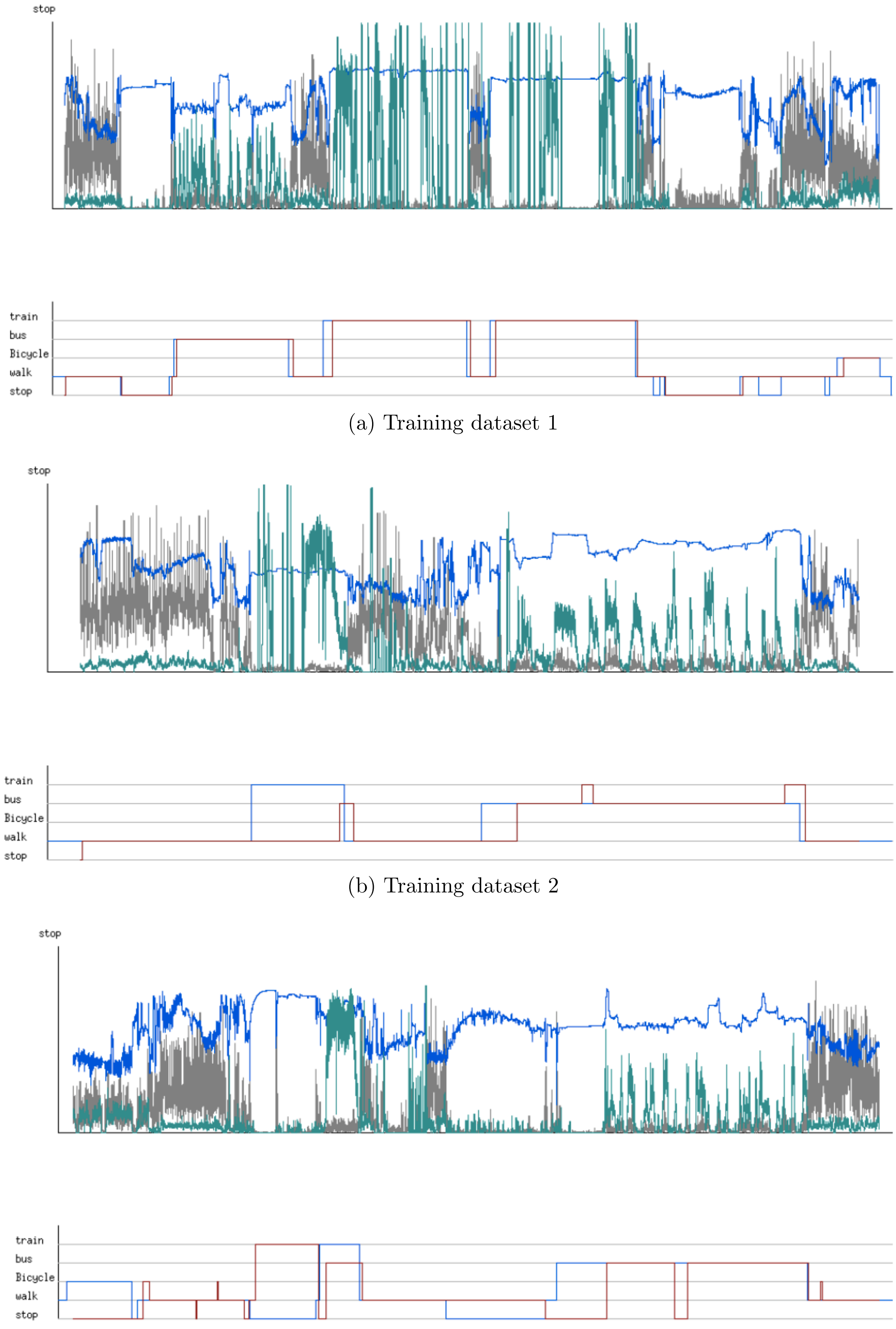



(b) Training dataset 2




(c) Test data

Figure 3.12: Experimental results by using the smoothing function in Eq. 3.10




(a) Training dataset 1

(b) Training dataset 2

(c) Test data

Figure 3.13: Experimental results by using Evolution Strategy for parameter optimization (raw data)

Table 3.8: Summary of experimental results

| | Experiment | Number of data | Number of fitting data | Fitting rate(%) | Running time(ms) |
|---|---|---|---|---|---|
| Raw data | traning | 18923 | 13362 | 70.6% | 56 |
| | test | 11251 | 5658 | 50.2% | 33 |
| Smoothing function Eq. 3.9 | traning | 18923 | 13632 | 72.0% | 53 |
| | test | 11251 | 6792 | 60.3% | 31 |
| Smoothing function Eq. 3.10 | traning | 18923 | 14574 | 77.0% | 56 |
| | test | 11251 | 8292 | 73.7% | 31 |
| ES for raw data | traning | 18923 | 15936 | 84.2% | 182149 |
| | test | 11251 | 9407 | 83.6% | 191762 |
| ES for Smoothing function Eq. 3.9 | traning | 18923 | 16631 | 87.8% | 186698 |
| | test | 11251 | 9872 | 87.7% | 193606 |
| ES for Smoothing function Eq. 3.10 | traning | 18923 | 16925 | 89.4% | 186362 |
| | test | 11251 | 10054 | 89.3% | 190135 |

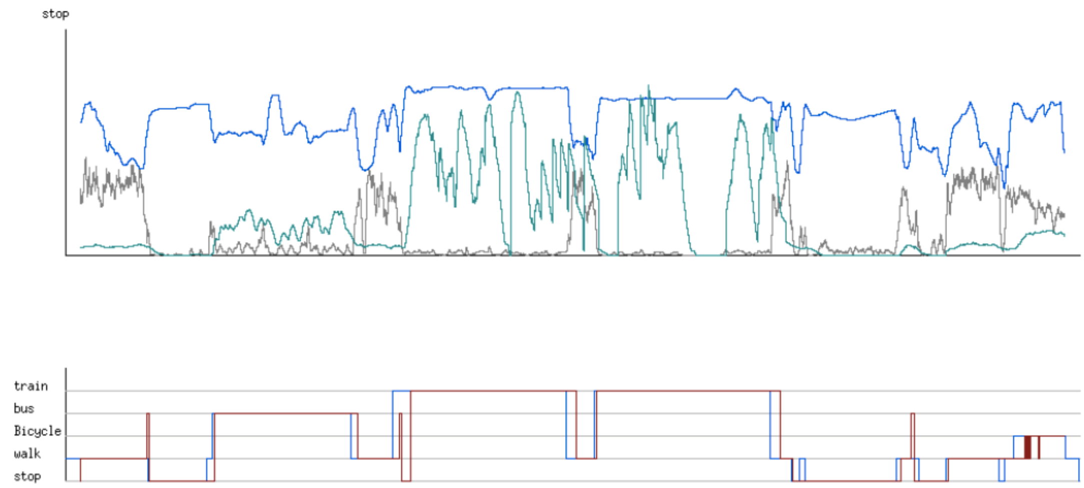
Fig. 3.14a, 3.14b and 3.14c show the best results by ES when the smoothing function in Eq. 3.9 is applied for the first and second training dataset and the test dataset, respectively. The total number of fitting data is 16631 from 18923. The running time for the total training phase is 186698 ms. The fitting rate is 87.8%. In case of the test dataset the number of fitting data is 9872 from 11251 test data. The running time is 193606, the fitting rate is 87.7%.

Fig. 3.15a, 3.15b and 3.15c depict the best results by ES when the smoothing function in smoothing function Eq. 3.10 is applied for the first and second training dataset and the test dataset, respectively. The total number of fitting data is 16925 from 18923. The running time for the total training phase is 186362 ms. The fitting rate is 89.4%. In case of the test dataset the number of fitting data is 10054 from 11251 test data. The running time is 190135, the fitting rate is 89.3%. Fig. 3.16 presents the average of the best fitness values based on the ten simulations in this latter case. The population size is 500, the number of generations is 20000, and the evaluation time step is 1000.

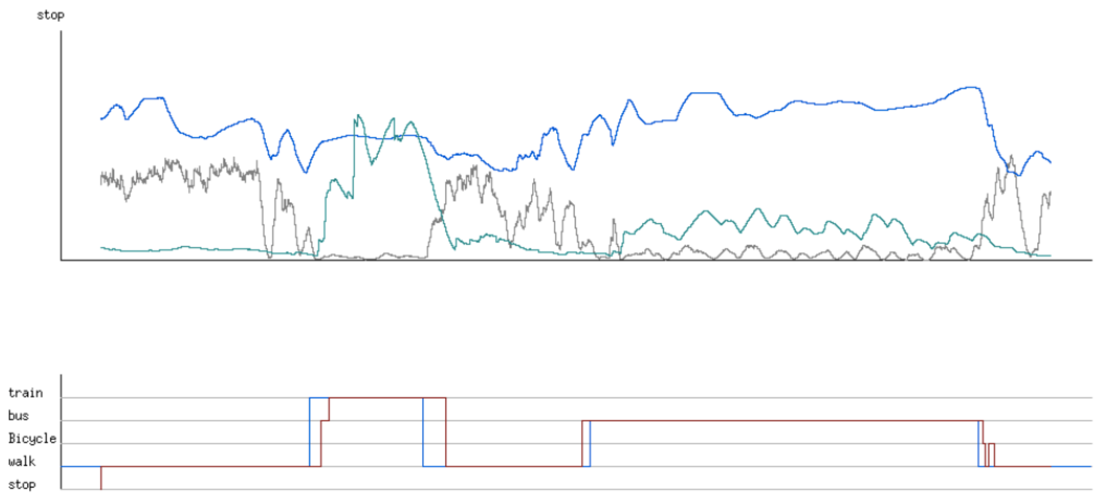
Table 3.8 summarizes the experimental results. The best results were obtained by evolution strategy.

### 3.4.2 Experimental Results for Estimation of Robot Interaction Modes

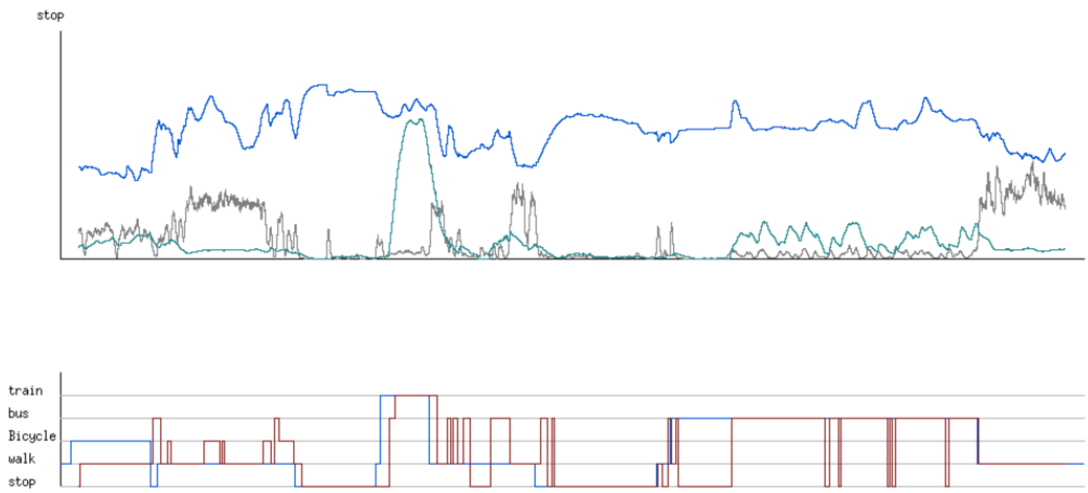
This section shows comparison results and analyzes the performance of the proposed method. In the spiking neural network there are 5 inputs in the input layer: accel-




(a) Training dataset 1

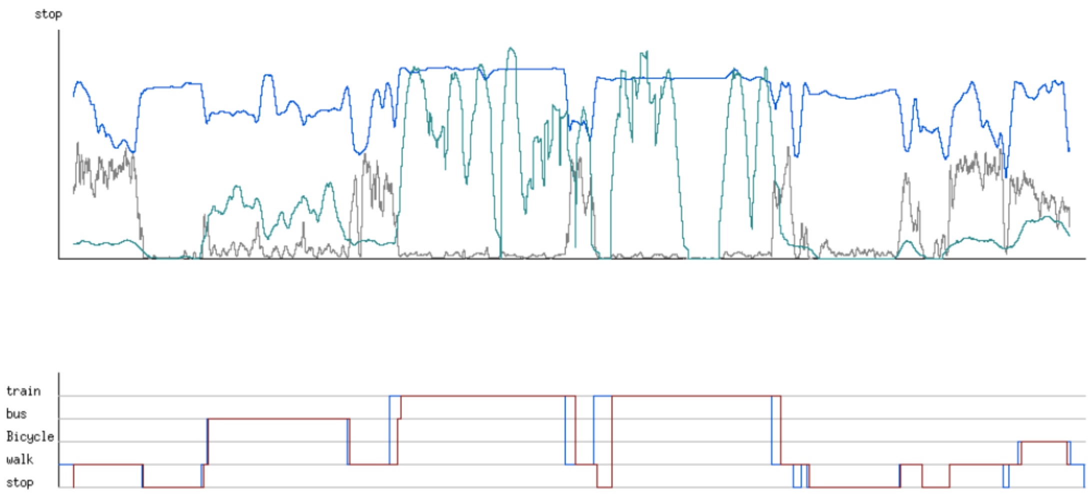



(b) Training dataset 2

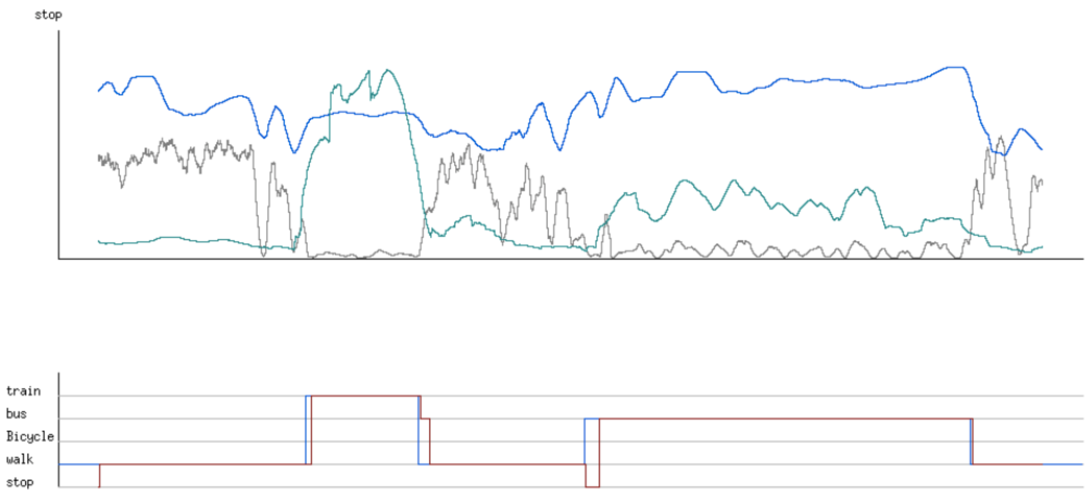



(c) Test data

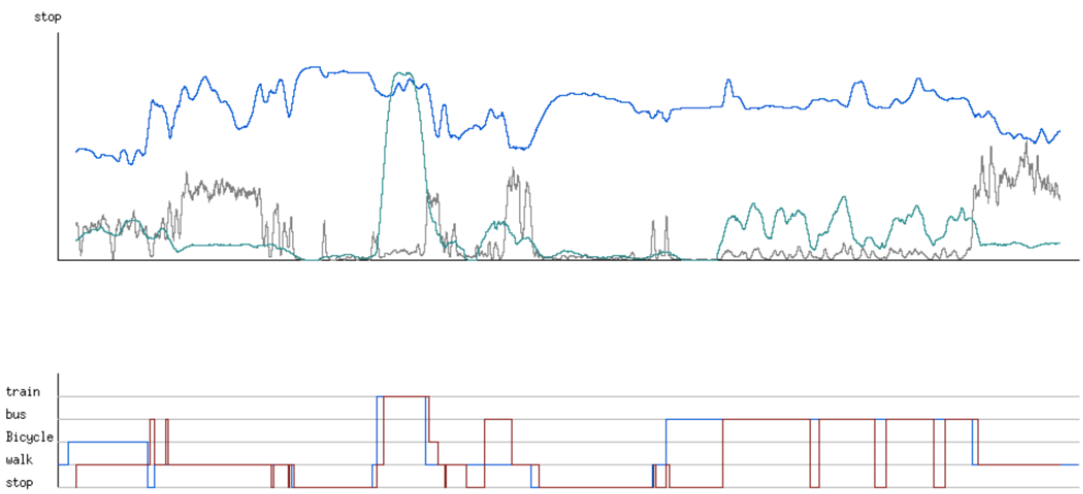
Figure 3.14: Experimental results by using evolution strategy for parameter optimization (smoothing function Eq. 3.9)




(a) Training dataset 1




(b) Training dataset 2




(c) Test data

Figure 3.15: Experimental results by using evolution strategy for parameter optimization (smoothing function Eq. 3.10)

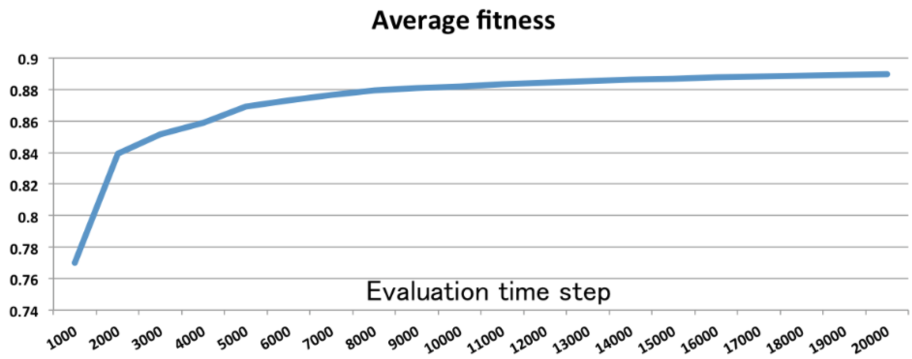



Figure 3.16: Average of the best fitness values based on ten simulations


Figure 3.17: Robot interaction modes

eration, angular velocity, and attitude of pitch, roll, and yaw. In the output layer there are 7 outputs related to the following 7 robot interaction modes (Fig. 3.17): (1) TableMode(front), (2) TableMode(back), (3) RobotMode, (4) HandMode, (5) BreastPocketMode, (6) JacketPocketMode, (7) TrouserPocketMode. The parameters of the neural network are as follows: the temporal discount rate for refractoriness ($\gamma^{ref}$) is 0.88, the temporal discount rate for PSP ($\gamma^{PSP}$) is 0.9, the threshold for firing ($q_{pul}$) is 0.9, and $R$ is 1. Fourteen training datasets and 4 test datasets are used in the experiments. When obtaining the training set, in the case of BreastPocketMode, JacketPocketMode, and TrouserPocketMode the person was walking for about 2 minutes, then standing for about 2 minutes. In the TableModes, RobotMode, and HandMode there was no motion.

Fig. 3.18 illustrates the experimental example of the measured smart phone mode. The cyan line is the high-pass filtered data measured by the accelerometer. The green line depicts the angular velocity calculated by the low-pass filtered data measured by the gyro sensor. The red line is the attitude of pitch data. The blue line is the attitude of roll data. The pink line is the attitude of yaw data. The second part of Fig. 3.18 shows the target output (blue line) and the estimated output by FSNN (red

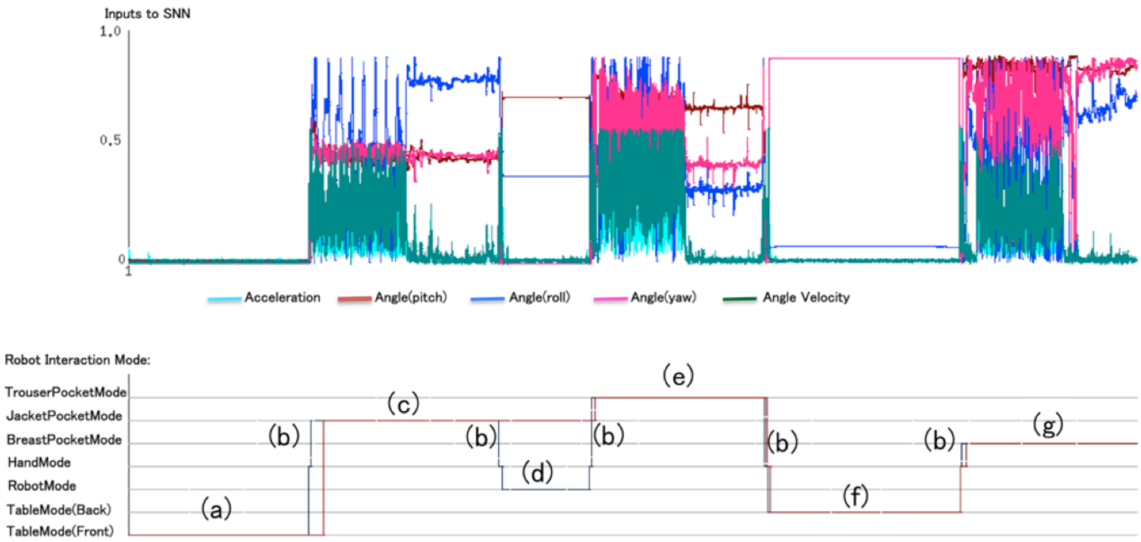



Figure 3.18: Experimental results by using the raw data for training dataset 2

line). The number of spiking neurons is 5. These neurons are used for measuring the 7 robot interaction modes.

In the first experiment the sensor's raw data are used as input to the FSNN. Fig. 3.18 shows experimental results by using the raw data of the second training dataset. In this case the phone is put on the table by the front side (a), and the person takes it in hand (b). Thereafter he/she puts the phone in jacket pocket (c) and takes out the phone putting it on the robot base (b,d). After that the person puts the phone in trouser pocket (b,e). Then he/she takes out the phone putting it on the table by the back side (b,f), and finally he/she takes the phone putting it in breast pocket (b,g). The number of fitting data is 16911 from 19213 training data in the case of second dataset. There are 14 training datasets, the total number of fitting data is 64936 from 72638 training data, and the running time is 4410 ms. The fitting rate is 89.4%. Fig. 3.19 shows experimental results by using the raw data of second test dataset. The number of fitting data is 5966 from 7974 test data. There are 4 test datasets, the total number of fitting data is 31377 from 40096 test data, and the running time is 1688 ms. The fitting rate is 78.3%.

In order to reduce the noise, some smoothing functions have to be used as mentioned in Section 3.3.1. Two different kinds of weighted moving averages are applied. In the second experiment I present results by using smoothing function described in Eq. 3.9. Figure 3.20 depicts the result for second training dataset. The number of fitting data is 18234 from 19213 training data. The total number of fitting data using all training datasets is 63659 from 72638 training data, the running time is 4445

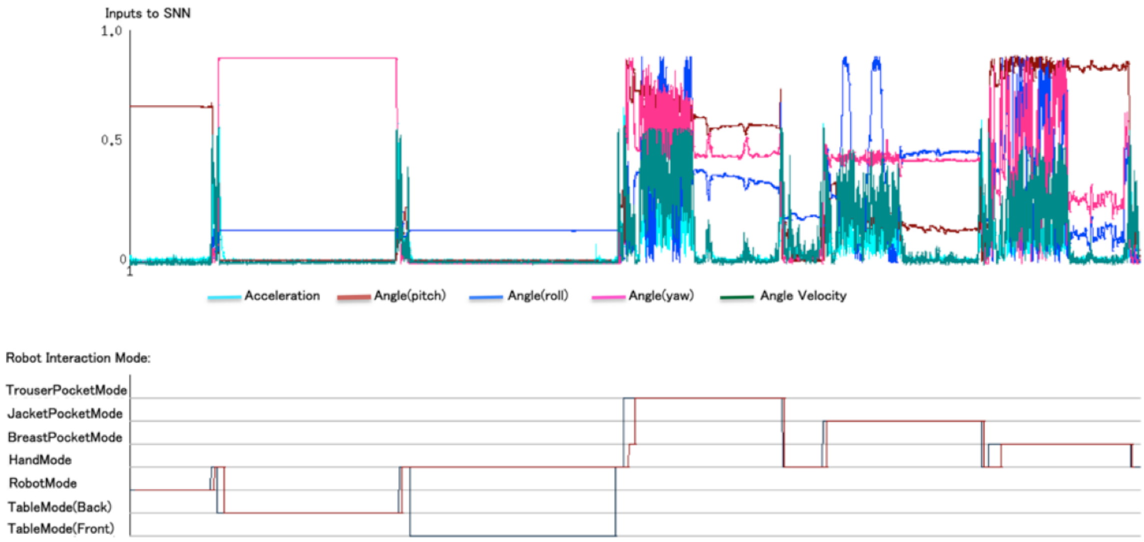



Figure 3.19: Experimental results by using the raw data for test dataset 2

ms, and the fitting rate is 87.6%. Figure 3.21 shows experimental results by using smoothing function in Eq. 3.9 for test dataset 2. The number of fitting data is 6911 from 7974 training data. The total number of fitting data using all test datasets is 34327 from 40096 test data, the running time is 1609ms, the fitting rate is 85.6%.

In the third experiment, I present experimental result by the other smoothing function defined by Eq. 3.10. Figure 3.22 illustrates the results for second training dataset. In the case of second training dataset the number of fitting data is 18507 from 19213, and for all training datasets the total number of fitting data is 69356 from 72638, the running time is 4438 ms, the fitting rate is 95.5%. Fig. 3.23 shows experimental results by using smoothing function in Eq. 3.10 for test dataset 2. The number of fitting data is 7499 from 7974 test data. The total number of fitting data for all test datasets is 36842 from 40096 test data, the running time is 16104 ms, the fitting rate is 91.9%.

In the fourth experiment I use evolution strategy for optimizing the parameters of FSNN. The population size is 100, the number of generations is 6000, and the evaluation time step is 1000, $_a = 0.01$, $_b = 0.005$, $_v = 0.05$. Figure 3.24 illustrates the best results by ES for the raw data of second training dataset. The total number of fitting data using all training datasets is 68933 from 72638. The application of ES has an additional computational cost. The running time is 684259 ms, the fitting rate is 94.9%. Figure 3.25 shows the result for second test dataset after using evolution strategy for optimizing the parameters of FSNN based on training datasets. In the case of test datasets the total number of fitting data is 34842 from 40096 test data.

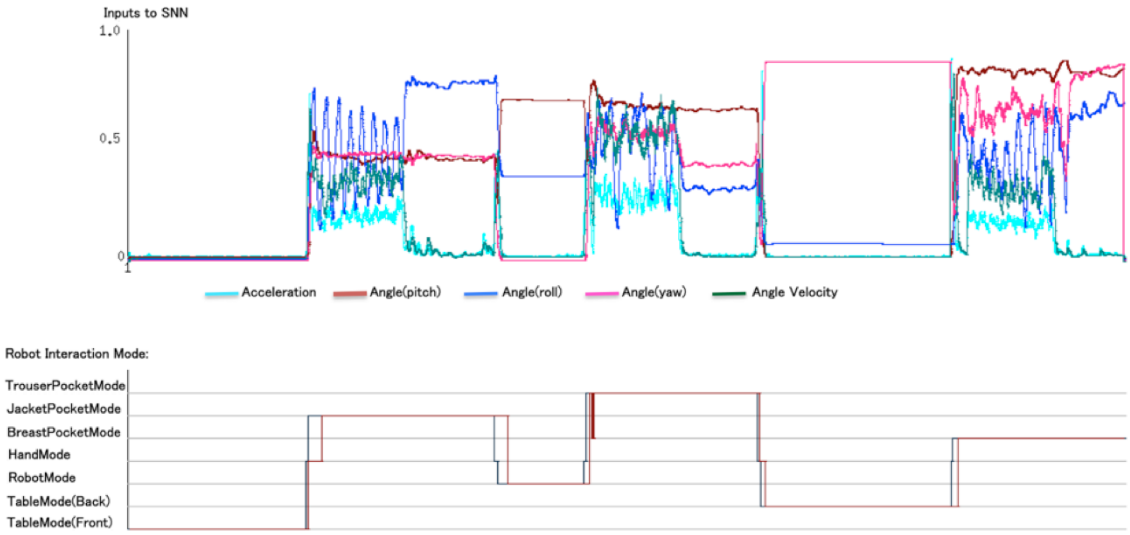



Figure 3.20: Experimental results by using the smoothing function in Eq. 3.9 for training dataset 2

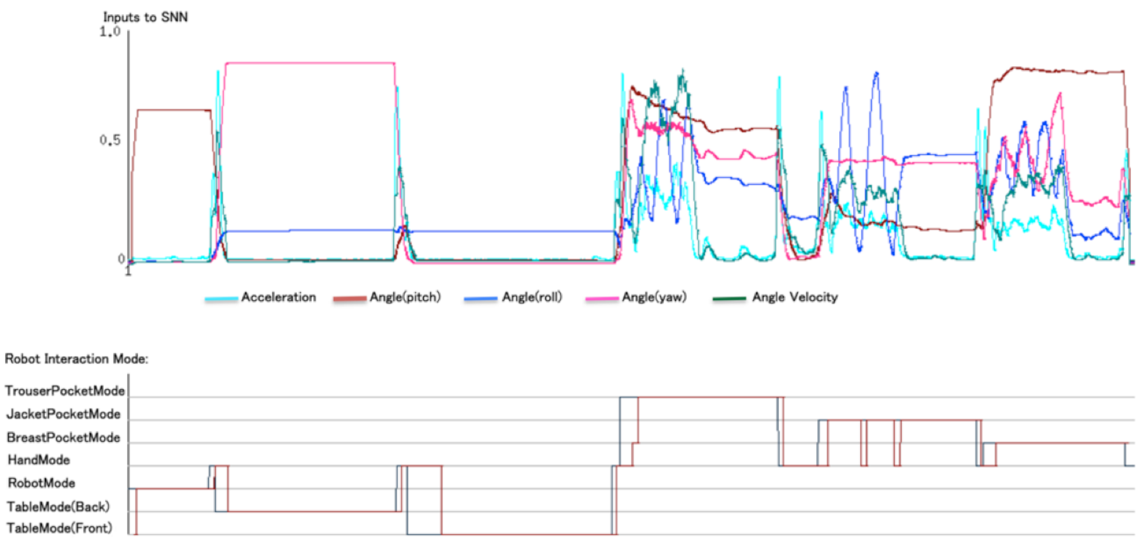



Figure 3.21: Experimental results by using the smoothing function in Eq. 3.9 for test dataset 2

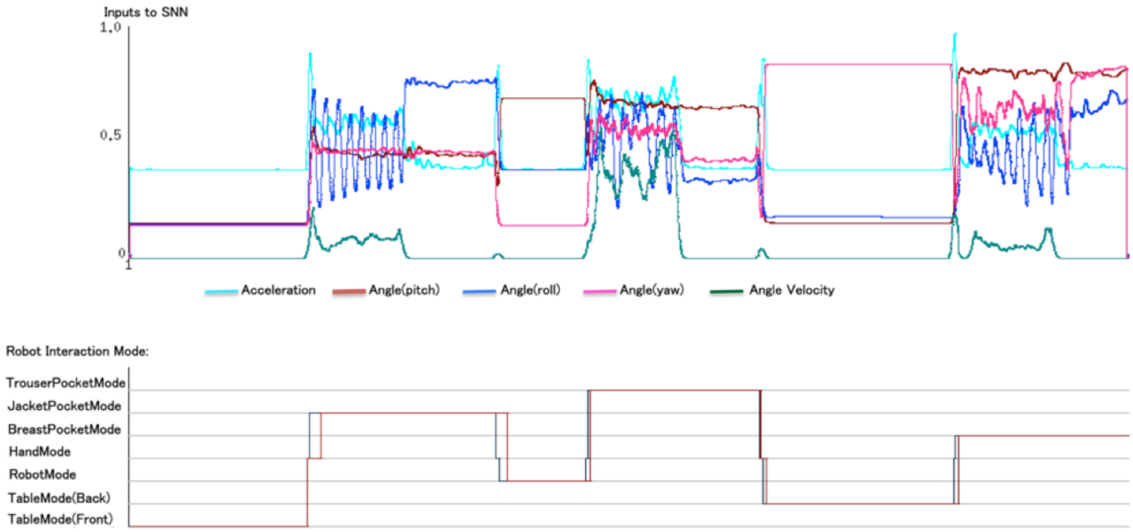



Figure 3.22: Experimental results by using the smoothing function in Eq. 3.10 for training dataset 2

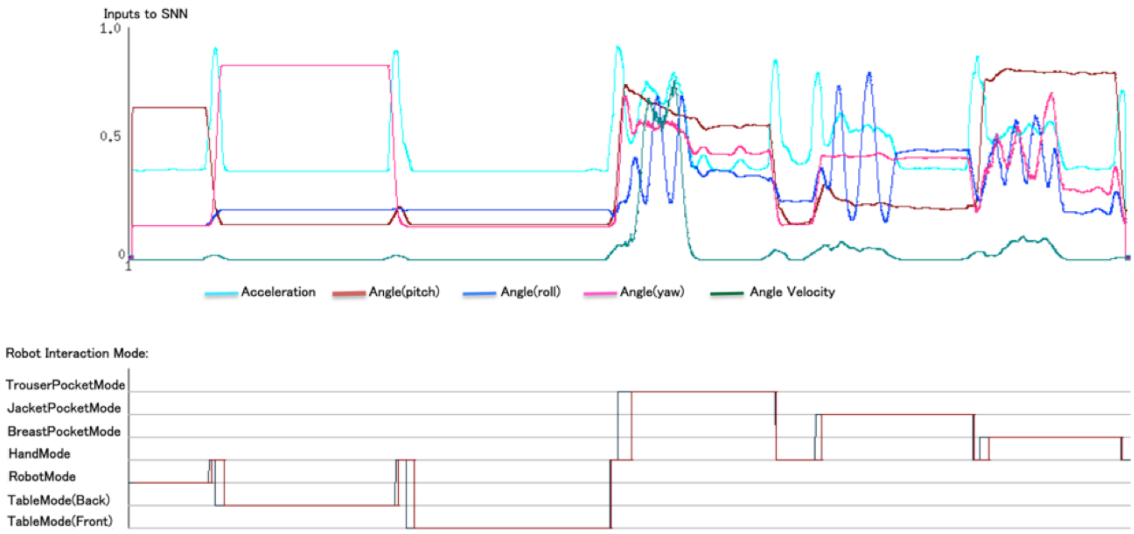



Figure 3.23: Experimental results by using the smoothing function in Eq. 3.10 for test dataset 2


Figure 3.24: Experimental results by using ES for parameter optimization for training dataset 2 (for raw data)

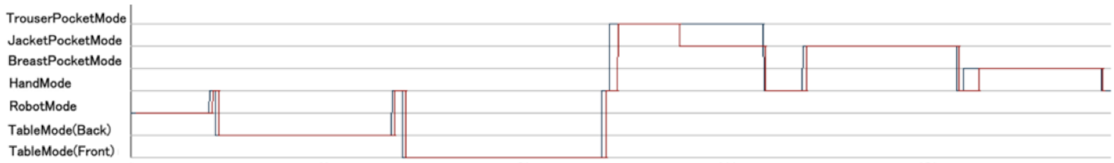



Figure 3.25: Experimental results by using FSNN for test dataset 2 after parameter optimization (for raw data)

The running time is 1810 ms, the fitting rate is 86.9%.

Figure 3.26 depicts the best results by ES for training dataset 2 when using smoothing function in Eq. 3.9. The total number of fitting data using all training datasets is 69441 from 72638. The running time is 683293 ms, the fitting rate is 95.6%. Figure 3.27 shows the result for second test dataset when using smoothing function in Eq. 3.9 after using evolution strategy for optimizing the parameters of FSNN based on training datasets. In the case of test datasets the number of fitting data is 35805 from 40096 test data. The running time is 1743 ms, the fitting rate is 89.3%.

Figure 3.28a presents the best results by ES for training dataset 2 when using smoothing function in Eq. 3.10. The total number of fitting data using all training datasets is 70604 from 72638. The running time is 654505 ms, the fitting rate is 97.2%. Figure 3.28b shows the result for second test dataset when using smoothing function in Eq. 3.10 after using evolution strategy for optimizing the parameters of FSNN based on training datasets. In the case of test datasets the number of fitting data is 37730 from 40096 test data. The running time is 1586 ms, the fitting rate is 94.1%.

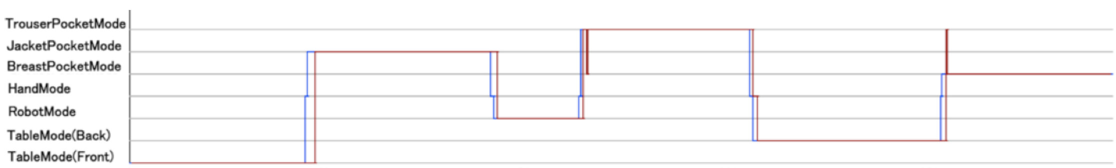
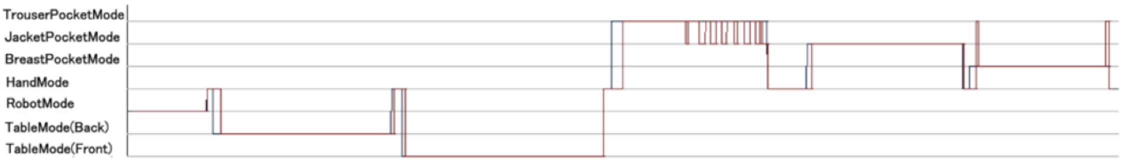



Figure 3.26: Experimental results by using ES for parameter optimization for training data set 2 (using smoothing function in Eq. 3.9)


Figure 3.27: Experimental results by using FSNN for test dataset 2 after parameter optimization (using smoothing function in Eq. 3.9)


(a) Training data set 2




(b) Test dataset 2

Figure 3.28: Experimental results by using ES for parameter optimization (using smoothing function in Eq. 3.10)

Table 3.9 summarizes the experimental results. The best results were obtained by evolution strategy.

### 3.4.3 Experimental Results for Estimation of Indoor Human activity

The experiment in this paper was conducted inside the room as shown in Fig. 3.1. In this experiment I used 25 sensor devices. As depicted in Fig. 3.2, four global measurement sensors (two Kinect and two LRF sensors) were installed in living room and kitchen, while 21 local measurement sensors (one bed sensor, eight air pressure sensors, six sensor tags, and six SunSPOTs) were installed on chair, bed, wardrobe, refrigerator, washstand, washing machine, door, toilet, kitchen, and bathroom.

Figure 3.31 is snapshot of human behaviors in the experiment. The human behavior flow in the experiment is as follows: (a) the person leaves the toilet, (b) uses the washstand, (c) changes clothes, (d) the behavior for walking to kitchen room, (e) the behavior when using the refrigerator, (f) when preparing a food, (g) the behavior when the person has finished preparing the food and walked to the living room to sit on chair, (h) the behavior of taking a breakfast.

Table 3.9: Summary of experimental results

| Method | Experiment | Number of data | Number of fitting data | Fitting rate(%) | Running time(ms) |
|---|---|---|---|---|---|
| Raw data | traning | 72638 | 64936 | 89.4 | 4410 |
| | test | 40096 | 31377 | 89.4 | 1688 |
| Smoothing function | traning | 72638 | 63659 | 87.6 | 4445 |
| Eq. 3.9 | test | 40096 | 34327 | 85.6 | 1609 |
| Smoothing function | traning | 72638 | 69356 | 95.5 | 4438 |
| Eq. 3.10 | test | 40096 | 36842 | 91.9 | 1610 |
| ES for raw data | traning | 72638 | 68933 | 94.9 | 684259 |
| | test | 40096 | 34842 | 86.9 | 1810 |
| ES for Smoothing | traning | 72638 | 69441 | 95.6 | 683293 |
| function Eq. 3.9 | test | 40096 | 35805 | 89.3 | 1743 |
| ES for Smoothing | traning | 72638 | 70604 | 97.2 | 654505 |
| function Eq. 3.10 | test | 40096 | 37730 | 94.1 | 1586 |

Table 3.10: Parameters of the spiking neural network

| $N_S$ | $N$ | $\gamma^{syn}$ | $\gamma^{ref}$ | $\gamma^{psp}$ | $R$ | $\theta$ | $\gamma^{wgt}$ | $\xi^{wgt}$ |
|---|---|---|---|---|---|---|---|---|
| 25 | 19 | 0.9 | 0.95 | 0.92 | 1 | 0.45 | 1.0 | 0.99 |

The parameters of SNN are presented in Table 3.10.

Figure 3.29 illustrates simulation result for sensor state and behavior state. Here, we can understand the sensor is on, off or fired, we can also know the behavior's fired state. This monitoring system can be used to check sensor error.

Figure 3.30 shows simulation result for human behavior on iPad. Blue line and orange line are human tracks. Here, I use Kinect sensor to measure human position. Red circle defines the local measurement sensor position. When the local sensor is fired, I can use human position to localize the sensor position. I can also localize the sensor installed on the furniture and consumer electronics. Through this simulation, we can understand the spatio-temporal pattern of the human behavior.

Figure 3.32 shows the human behavior estimation result. The red line shows the state when sitting on chair, while the green line shows the state of walking, and the blue line shows the state of other behavior. From these results, we can understand correctly the method of estimating human behaviors. I can conclude that there is a high probability that two behaviors fire at the same time, for example, the behavior when sitting on chair and having a breakfast occurred at the same time.

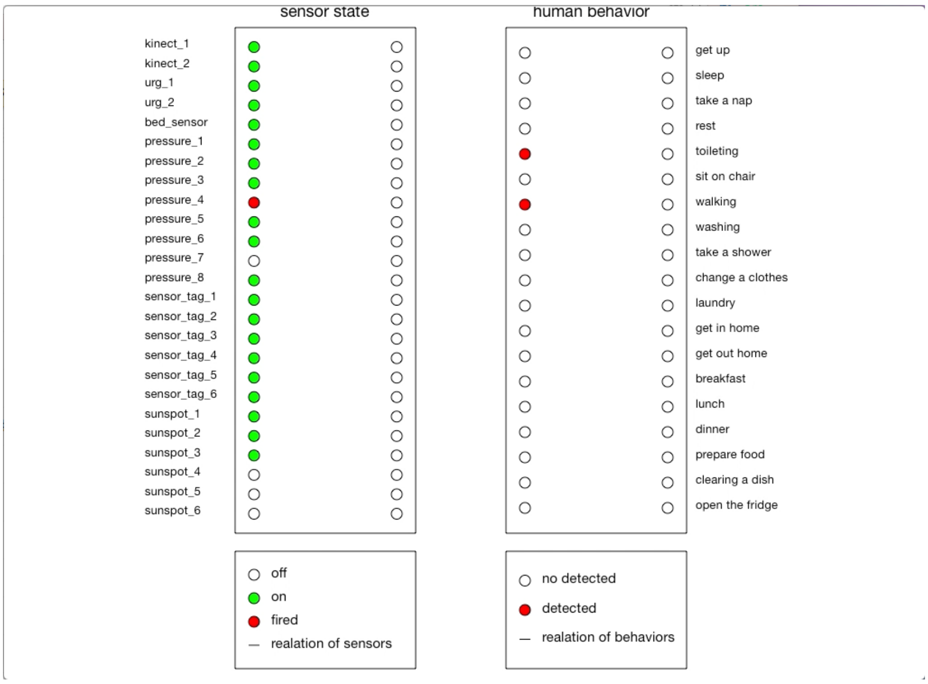



Figure 3.29: Fire states of sensors and behaviors

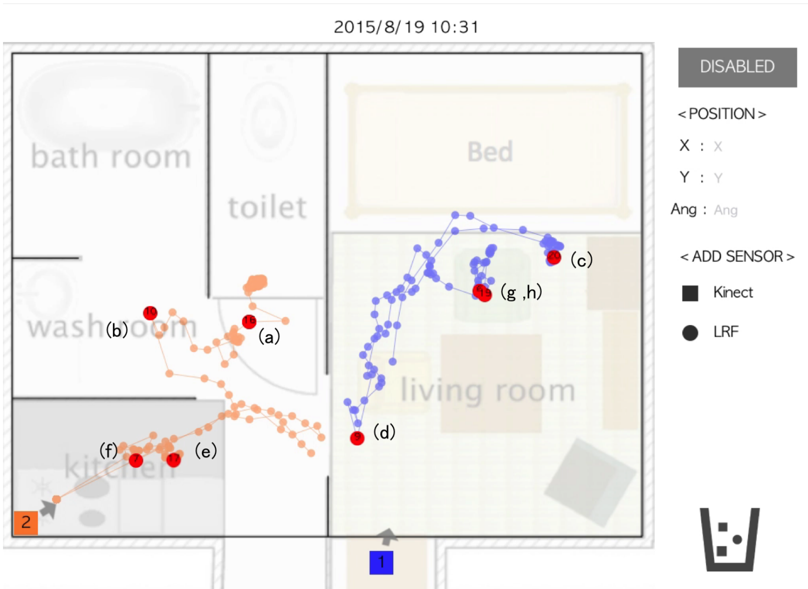



Figure 3.30: Simulation result of human tracking

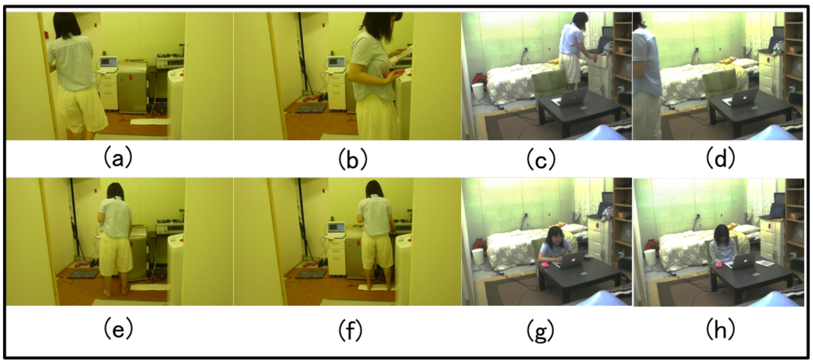



Figure 3.31: Snapshot of human behaviors in the experiment

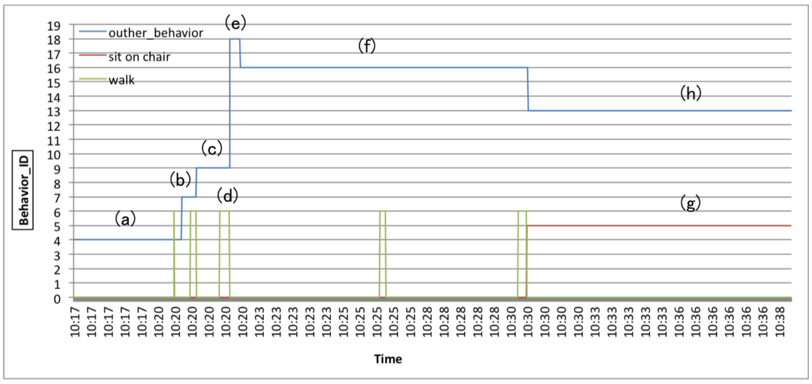



Figure 3.32: Experimental result

## 3.5 Summary

In indoor human activity measurement, I discussed a methodology for human behavior estimation by wireless sensor networks in informationally structured space. I provided several experimental results and discussed the effectiveness of the proposed method. As a future work, in order to realize a flexible measurement of human behavior I will discuss more details of the relations between sensor and behavior and relations between behaviors in informationally structured space.

In outdoor human activity measurement, I analyzed the performance of human transport mode and interaction mode estimation by fuzzy spiking neural network in informationally structured space using smart phone sensors. First, I applied a fuzzy spiking neural network to extract the human activity outdoors and smart phone states. Next, I discussed how to update the base value by preprocessing to generate input values to the spiking neurons. Thereafter, the learning method of fuzzy spiking neural network based on the time series of measured data using evolution strategy was explained. Experimental results showed the effectiveness of the proposed method. However, there is trade-off or balance between the update of the base values and the enhancement of the output mode estimation by weight connection between neurons. The update of the base values can inhibit incorrect fires by external noise, while that might inhibit suitable fires according to small change of the measured data. The weight connection can enhance the suitable fires based on the prediction as a result of the temporal learning by time-series of the measured data, while that might cause incorrect fires by external noise. Therefore, I intend to improve the learning method to realize the suitable balance of enhancement and inhibition as a future work. As other future works, I aim to investigate more datasets. I intend to improve the learning performance according to human life logs.

# Chapter 4

# Informationally Structured Space for Robot Partners

## 4.1 Feature Extraction Layer in Informationally Structured Space

In the proposed method, I implemented relevance theory to build mutual cognitive environment called informationally structured space; Rasmussen's behavior model for conversational system; emotional model and gesture recognition to realize natural communication between human and robot partner. In this section I discuss and compare my proposed method to previous researches. Since mutual cognitive environment has a close relationship with ambient intelligence, I will start to review previous researches on this field. Next, I will discuss emotional model, thereafter gesture recognition and finally conversation system.

In works related to ambient intelligence [84], they build smart environment based on software reference architecture. The smart environment is used to conduct the perception process in a standard office. However, the paper only uses motion detection in order to measure data from sensor. Details such as gestures were not included. On the other hand, [72] proposed a method called mixed context-aware inference, which is a novel sensor-based context-aware system focusing on three inference processes: rule, inference and pattern driven. [28] used various sensors to measure data, which enabled this method to get more accurate result. Moreover, the usage of cloud technology made the process time become shorter. However, this method is difficult to realize concerning the high cost. Furthermore, since user has to wear special clothes to get data from the sensors, it is very cumbersome. [29] proposed fusion-based architecture, detection in activity and location patterns using Hidden Markov Model (HMM).

Although it has a good accuracy in computation, HMM has a disadvantage in high computation cost.

For the emotional model concept, [123] proposed a method, which effects on occupational roles (security vs. health-care), gender (male vs. female), and personality (extrovert vs. introvert) on user acceptance of a social robot. However, they only use stereotype to conduct the evaluation, which makes the result arguable. [24] proposed a method to generate emotion of a robot using expert knowledge by fuzzy logic. The emotion of the robot is determined using 3 types of input data, such as the robot's personality, the ambient environment and the interaction with human. However, since emotional expression is using LED only, the emotional expression done by the robot has a lot of limitations and it is difficult to be evaluated. Emotional model proposed by [57] used episodic memory system, as a result of long term human robot interaction and emotion generation reaction. Although the method of this paper is very interesting, the application is only possible in the virtual environment. [50] used the integration of environment, robot self-states and feedback behavior for generating robot emotion (human data is not included). As a conclusion in the previous research the definition of the robot's emotion is not clear, while in my thesis the change in environment, human state (gesture and distance) effects the robot's emotion, which linked into the conversation system content and robot's facial expression.

For the works related to gesture recognition, [46] proposed a novel approach to real-time and continuous gesture recognition for flexible, natural, and robust human-robot interaction (HRI), and the generation of an ad-hoc HMM. As mentioned before, one disadvantage of HMM is the high computational cost. Gesture recognition method proposed in [128] is based on combination of the CyberGlove and Kinect sensor, which could recognize various gestures. The using of the CyberGlove for measurement device shows that this method cannot be directly realized in daily life now owing to its price. [76] proposed a method in gesture recognition by using on-board monocular camera and specialized gesture detection algorithms. Here, also the dynamic movement primitives (DMP) model is employed. In this method, since the depth information is not acquired, the recognition has many limitations. The gesture recognition in my system used Kinect sensor has lower the cost, although my method is not as accurate as the previous research. In addition, since the robot partner uses iPhone as a mainframe to speed up the computational cost, the growing neural gas algorithm is adopted to extract features of sensing data beforehand and then spiking neural network to recognize the gesture. A deep explanation of gesture recognition can be found in Section 4.2.2.

In the conversation system related research, [89] proposed a dialogue system framework architecture that supports cognitive load prediction and situation-dependent decision making and manipulation of the HCI. This paper also proposed the multimodal fusion and fission which shows the system of how to interact with human and learning. On the other hand, [75] proposed multilingual dialogue systems and seamless deployment to mobile platforms. English and Mandarin systems in various domains (e.g. movie, flight and restaurant) are implemented with the proposed framework. However, since the dialogue system works in online server (connected into Internet), it cannot work in offline state. [77] presented the used of data-driven approach to improve Spoken Dialog System (SDS) performance by automatically finding the most appropriate terms to be used in system prompts. On the other hand, the conversation system of my method can be conducted in the server or in the robot partner. The conversation system in my method is built as the result of the connectivity between the environment and the robot individual intelligence. Basically, the conversation system contents is stored in the ISS server. However, minimum conversation system is also stored in the robot partner used as input-output interface. When the robot starts to communicate with the human, the robot will conduct the learning process based on the conversation contents time, human state, and environment state.

Similar systems to my proposed system were developed previously. For example, [116] explained the recognition of environment by the usage of color property; [106] implemented face recognition technology and [13] employed the gesture recognition technology to perform human recognition. Moreover, [11] used person localization based on the face recognition and skin color detection [117] to develop their robot. Since communication also involves the perception of intention and feeling, human emotion plays an important role in the act of communication, which leads to an action. [10] besides consider human gesture as the information for the robot, the emotional information such as facial expression and voice tones are also utilized to determine the robot action. Similar approaches can be found in [134]. On another variant, according to [14], a robot partner with emotional model can give meaning and value to the perceptual information, which leads to a decision based on internal and external state. That is when a person is in the state of sadness, his action will show the state of sadness. Therefore, it is also important to implement emotion into the robot partner. The comparison of these methods with my method can be found in Section 4.5.5 in the discussion of the experiments.

## 4.2 Robot Behavior Control System

### 4.2.1 Emotional Interaction

I develop emotional models composed by emotion, feeling, and mood measured based on a time scale. This development is done by assuming that emotion change temporally based on the perceptual information on the internal state and the external environment [129, 69]. Emotion is used for intermediating the perceptual system and emotional model, while considering it as an intense short-term mental state based on perceptual information. This leads to the assumption of the emotion changing, which depends on specific perceptual information. As a part of the robot partner, smartphone converts independently the gestures and the environmental information as perceptual information to emotional input based on predefined data. Each feeling is updated as the summation of emotions.

The $i$-th emotional input $u_i^E(t)$ is generated based on the $u_{j,k}^I(t)$ perceptual information as follows:

$$u_i^E(t) = w_{i,j,k}^E \cdot u_{j,k}^I(t), \tag{4.1}$$

where $w_{i,j,k}^E$ is the degree of contribution from the $j$-th gesture and $k$-th environmental data to the $i$-th emotion ($-1 \leq w_{i,j,k}^E \leq 1$).

[129] defined five different feeling models. Now, I apply the model where the state of the $i$-th feeling $u_i^F(t)$ is updated by the emotional input from the viewpoint of bottom-up construction and the top-down constraints from mood values are also considered as displayed in Fig 4.1:

$$u_i^F(t) = \tanh(\kappa u_i^F(t-1) + (1-\kappa)\left[E + F_i\right]), \tag{4.2}$$

where

$$\begin{aligned} E &= \sum_{j=1}^{N^E} u_j^E(t-1) \\ F_i &= \sum_{j=1, j\neq i}^{N^F} w_{i,j}^F \cdot u_j^F(t-1) \\ \kappa &= \frac{\gamma^F}{1 + u_1^M(t-1) - u_2^M(t-1)}, \end{aligned} \tag{4.3}$$

where $\gamma^F$ is the temporal discount rate of feelings ($0 < \gamma^F < 1$), $N^E$ is the number of emotional inputs, $N^F$ is the number of feelings, $w_{i,j}^F$ is the stimulation or suppression

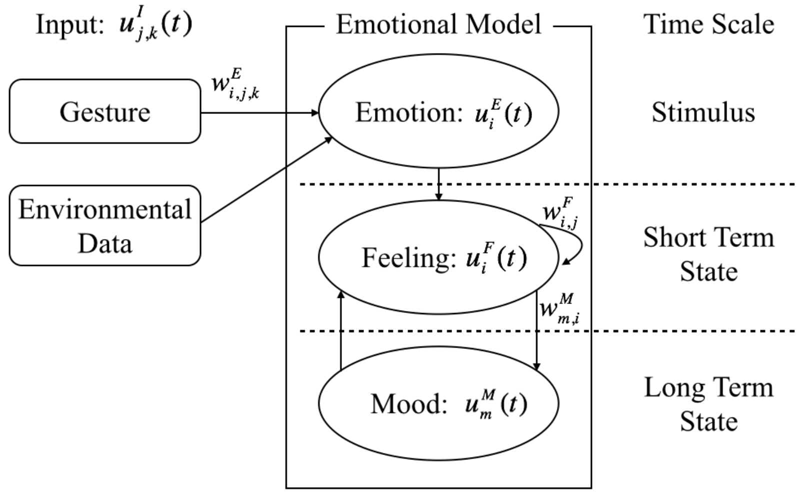



Figure 4.1: The structure of the proposed emotional method

coefficient from the $j$-th feeling to the $i$-th feeling ($-1 \leq w^F_{i,j} \leq 1$), and $u^M_m(t)$ is the value of the $m$-th mood. I use positive mood ($m = 1$) and negative mood ($m = 2$). The hyperbolic tangent is used to regulate the values of feelings.

Mood is defined as the long-term state updated by a change in feelings, and governs changes in feelings. While feeling is defined as a short-term state updated by a change in emotion. The state of the $m$-th mood is updated by the sum of feelings:

$$u^M_m(t) = \tanh\left[\gamma^M u^M_m(t-1) + (1-\gamma^M)\sum_{i=1}^{N^F} w^M_{m,i} u^F_i(t)\right], \tag{4.4}$$

where $\gamma^M$ is the discount rate and $w^M_{m,i}$ is the stimulation or suppression coefficient from the $i$-th feeling to the $m$-th mood ($-1 \leq w^M_{m,i} \leq 1$). The structure of the model is shown in Fig. 4.1. In this figure, it can be seen how the feeling and mood influence each other and the emotion can be considered as an input impulse to the feeling.

### 4.2.2 Gesture Interaction

In my thesis, the structured learning (SL) is adopted as similarly in [14, 12]. SL contains two stages, a topology learning phase and a spatio-temporal learning phase. The growing neural gas (GNG) is applied in the first phase for information extraction, and the spiking neural network (SNN) is applied in the second stage to recognize the gesture.

#### 4.2.2.1 Growing Neural Gas for Information Extraction

Unsupervised learning is performed by using data without any teaching signals [60, 83, 31, 32]. Self-Organizing Map (SOM) [60], Neural Gas (NG) [83], Growing Cell Structures (GCS) [31], and Growing Neural Gas [32] are some well known unsupervised learning methods that use the competitive learning approach. In SOM, the number of nodes and the topological structure of the network are designed beforehand [60]. In NG, the number of nodes is also constant, however its topological structure is updated according to the distribution of sample data [83]. On the other hand, GCS and GNG can dynamically change the topological structure based on the adjacent relation (edge) referring to the ignition frequency of the adjacent node according to the error index. GCS does not delete nodes and edges and it must consist of $k$-dimensional simplices whereby $k$ is a positive integer chosen in advance. On the other hand, GNG can delete nodes and edges based on the concept of ages [32]. The initial configuration of each network is a $k$-dimensional simplex, if $k = 1$ then it is a line, if $k = 2$ then it is a triangle, and if $k = 3$ then it is a tetrahedron [31]. GCS has been applied to construct 3D surface models by triangulation based on 2-dimensional simplex. However, because the GCS does not delete nodes and edges, the number of nodes and edges is over increasing. Another disadvantage of GCS is that it cannot divide the sample data into several segments. GNG can overcome these drawbacks. When applying GNG, the distance criterion is used for extracting human motions. The GNG algorithm is described in Algorithm 1.

The following notations are used in the learning algorithm of GNG [32, 33]: $\mathbf{r}_i$ is the 3-dimensional vector of a node (reference vector, $\mathbf{r}_i \in \mathbb{R}^3$); $\mathbf{v}$ is the 3-dimensional input data, calculated from the Kinect data, describes the relative position from shoulder where shoulder position is set at $(0, 0, 0)$, $A$ is a set of node indices, $N_i$ is a set of node indices connected to the $i$-th node, and $a_{i,j}$ is the age of the edge between the $i$-th and the $j$-th node.

#### 4.2.2.2 Spiking Neural Network for Gesture Recognition

In the second stage of the proposed method, the spiking neural network is applied to recognize the human gestures. However, in order to reduce the computational cost, a modified spike response model is applied [12, 118, 68]. I use two-layered SNNs which are composed of an input layer and an output layer. Each gesture is recognized by one SNN. The number of spiking neurons in each SNN is $N$, which is the same as the number of reference vectors. The proposed model is depicted in Fig. 4.2.

**Algorithm 2** GNG ALGORITHM

**Step 1:** Generate two units at random position, $\mathbf{r}_1$, $\mathbf{r}_2$ in $\mathbb{R}^3$. Initialize the connection set.
**Step 2:** Generate an input data $\mathbf{v}$ randomly according to $p(\mathbf{v})$ which is the probability density function of data $\mathbf{v}$.
**Step 3:** Select the nearest unit (winner), $s_1$ by Eq. (4.5) and the second-nearest unit, $s_2$ by Eq. (4.6).

$$s_1 = \arg\min_{i \in A} ||\mathbf{v} - \mathbf{r}_i|| \tag{4.5}$$

$$s_2 = \arg\min_{i \in A \backslash \{s_1\}} ||\mathbf{v} - \mathbf{r}_i|| \tag{4.6}$$

**Step 4:** If a connection between $s_1$ and $s_2$ does not exist already, create the connection. Set the age of the connection between $s_1$ and $s_2$ to zero, $a_{s_1,s_2} = 0$.
**Step 5:** Add the squared distance between the input data and the winner to a local error variable (which is initialized as 0): $E_{s_1} \leftarrow E_{s_1} + ||\mathbf{v} - \mathbf{r}_{s_1}||^2$.
**Step 6:** By using the total distance to the input data, update the reference vectors of the winner node, $s_1$ (see Eq. (4.5)) using Eq. (4.7) and its direct topological neighbors using Eq. (4.8) by the learning rate $\eta_1$ and $\eta_2$, respectively.

$$\mathbf{r}_{s_1} \leftarrow \mathbf{r}_{s_1} + \eta_1 \cdot (\mathbf{v} - \mathbf{r}_{s_1}) \tag{4.7}$$

$$\mathbf{r}_j \leftarrow \mathbf{r}_j + \eta_2 \cdot (\mathbf{v} - \mathbf{r}_j) \quad \text{if } j \in N_{s_1} \tag{4.8}$$

**Step 7:** Increment the age of all edges emanating from $s_1$: $a_{s_1,j} \leftarrow a_{s_1,j} + 1 \quad \text{if } j \in N_{s_1}$
**Step 8:** Remove edges with an age larger than $a_{max}$. If this results in units having no more emanating edges, remove those units as well.
**Step 9:** If the number of input signals generated so far is an integer multiple of a parameter $\lambda$, insert a new unit using the following steps:

a. Select the unit $q$ with the maximum accumulated error according to Step 5.

b. Add a new unit $r$ to the network and interpolate its reference vector from $q$ and $f$ using Eq. (4.9), where $f$ is that neighbor of $q$ which the largest error has according to Step 5.

$$\mathbf{r}_r = 0.5 \cdot (\mathbf{r}_q + \mathbf{r}_f) \tag{4.9}$$

c. Insert edges connecting the new unit $r$ with units $q$ and $f$, and remove the original edge between $q$ and $f$.

d. Decrease the error variables of $q$ and $f$ by a fraction $\alpha$:

$$E_q \leftarrow E_q - \alpha \cdot E_q \tag{4.10}$$

$$E_f \leftarrow E_f - \alpha \cdot E_f \tag{4.11}$$

e. Interpolate the error variable of $r$ from $q$ and $f$:

$$E_r = 0.1 \cdot (E_q + E_f) \tag{4.12}$$

**Step 9:** Decrease the error variables of all units:

$$E_i \leftarrow E_i - \beta \cdot E_i \quad (\forall i \in A) \tag{4.13}$$

**Step 10:** Continue with Step 2 if a stopping criterion is not yet fulfilled. The net size or some performance measure can be used as a stopping criterion.

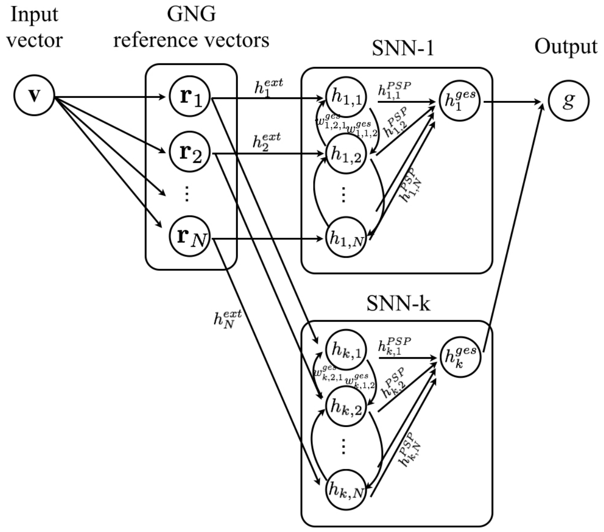



Figure 4.2: The proposed spiking neural network for gesture recognition.

The internal state $h_{k,i}(t)$ of a spiking neuron $i$ in the input layer for the $k$-th gesture is calculated as follows:

$$h_{k,i}(t) = \gamma^{syn} \cdot h_{k,i}(t-1) + h_{k,i}^{syn}(t) + h_{k,i}^{ref}(t) + h_i^{ext}(t)), \tag{4.14}$$

where $\gamma^{syn}$ is a temporal discount rate, $h_{k,i}^{syn}(t)$ includes the pulse outputs from the other neurons, $h_{k,i}^{ref}(t)$ is used for representing the refractoriness of the neuron, $h_i^{ext}(t)$ is the input to the $i$-th neuron from the environment.

The input to the $i$-th neuron from the external environment is calculated by the difference between the reference vector and the input vector:

$$h_i^{ext}(t) = \exp\left(-\gamma^{env} \cdot (\mathbf{v} - \mathbf{r}_i)^2\right), \tag{4.15}$$

where $\gamma^{env}$ is a coefficient, $\mathbf{v}$ is the input vector.

The pulse outputs from the other neurons, $h_{k,i}^{syn}(t)$ is calculated by:

$$h_{k,i}^{syn}(t) = \begin{cases} \tanh\left(\sum_{j=1,j\neq i}^{N} w_{k,j,i}^{ges} \cdot h_{k,j}^{PSP}(t-1)\right) & \text{if } h_i^{ext}(t) \geq \theta^{syn}, \\ 0 & \text{otherwise,} \end{cases} \tag{4.16}$$

where $w_{k,j,i}^{ges}$ is the weight from the $j$-th neuron to the $i$-th neuron in the $k$-th SNN ($k$-th gesture), $\theta^{syn}$ is a threshold, $h_{k,j}^{PSP}(t)$ is the PostSynaptic Potential (PSP) approximately transmitted from the $j$-th neuron in the $k$-th SNN at the discrete time $t$. The hyperbolic tangent is used to avoid the repeated firing by several neurons without an efficient input (without reaching the $\theta^{syn}$ threshold).

When the internal state of the $i$-th neuron reaches a predefined threshold level, a pulse is outputted as follows:

$$p_{k,i}(t) = \begin{cases} 1 & \text{if } h_{k,i}(t) \geq \theta^{pul}, \\ 0 & \text{otherwise}, \end{cases} \tag{4.17}$$

where $\theta^{pul}$ is a threshold for firing. In case of firing, $R$ is subtracted from the $h_{k,i}^{ref}(t)$ value of neuron $i$ as follows:

$$h_{k,i}^{ref}(t) = \begin{cases} \gamma^{ref} \cdot h_{k,i}^{ref}(t-1) - R & \text{if } p_{k,i}(t-1) = 1, \\ \gamma^{ref} \cdot h_{k,i}^{ref}(t-1) & \text{otherwise}, \end{cases} \tag{4.18}$$

where $\gamma^{ref}$ is a discount rate of $h_{k,i}^{ref}$ and $R > 0$.

The presynaptic spike output is transmitted to the connected neuron through the weight connection. The PSP is calculated as follows:

$$h_{k,i}^{PSP}(t) = \begin{cases} 1 & \text{if } p_{k,i}(t) = 1, \\ \gamma^{PSP} \cdot h_{k,i}^{PSP}(t-1) & \text{otherwise}, \end{cases} \tag{4.19}$$

where $\gamma^{PSP}$ is a discount rate of $h_{k,i}^{PSP}$ and $0 < \gamma^{PSP} < 1$. The PSP is excitatory if the weight parameter, $w_{k,j,i}^{ges}$ is positive. If the condition $h_{k,j}^{PSP}(t) < h_{k,i}^{PSP}(t)$ is satisfied, the weight parameter is trained based on the temporal Hebbian learning rule [42]:

$$w_{k,j,i}^{ges} \leftarrow \tanh\left(\gamma^{wgt} \cdot w_{k,j,i}^{ges} + \xi^{wgt} \cdot h_{k,j}^{PSP}(t) \cdot h_{k,i}^{PSP}(t)\right), \tag{4.20}$$

where $\gamma^{wgt}$ is a discount rate of the weights and $\xi^{wgt}$ is a learning rate.

The evaluation value for the $k$-th gesture in the output layer is calculated by:

$$h_k^{ges}(t) = \gamma^{ges} \cdot h_k^{ges}(t-1) + \sum_{i=1}^{N} h_{k,i}^{PSP}(t), \tag{4.21}$$

where $\gamma^{ges}$ is a discount rate. The gesture recognition at the discrete time $t$ is done by:

$$g(t) = \arg\max_k h_k^{ges}(t-1). \tag{4.22}$$

Finally, the overall recognition result is calculated as the most frequently selected gesture over time. The information flow is illustrated in Fig. 4.2.

## 4.3 Conversation System

The conversation system has been developed for many years with various architectures and standardizations. In the interaction between a human and a robot partner, when the robot partner leads the conversation, the robot partner has a good performance if the human's expectation can be fulfilled. However, if the human's expectation cannot be fulfilled, the interaction between them will be broken. On the other hand, if the human leads the conversation, interaction building between the human and the robot partner is difficult owing to the current technology. Therefore, in order to realize natural communication, it is supposed that first the robot partner leads the conversation, ideally in the middle of interaction the human also takes place to lead the conversation interactively. This process can be performed, as long as the robot partner conducts sequential transitional behavior, while in arbitrary timing it performs some action reflectively to human interruption behavior.

According to [96], human behavior based on information processing is composed of skill level, rule level, and knowledge level. The skill level is a daily common and repetitive behavior, which does not need memory and knowledge referring process while performing it. In other word, the skill level can be defined as an unconscious, reflective, and short time behavior. Meanwhile, the rule level is defined as a behavior based on customs and rules. This kind of behavior needs human memory and knowledge referencing process in order to perform the behavior correctly. Comparing to skill level, the rule level needs more time to conduct. The knowledge level is behavior performance for unknown or unfamiliar situations. In order to perform this kind of behavior, sufficient knowledge is needed. Otherwise, during the performance, the human can have new knowledge while doing some optimization to get the best result.

In the wider application, Rasmussen's behavior model has been applied to human and voice interface. I also believe that Rasmussen's behavior model can be applied to conversation system as well. I propose a human conversation system based not only on Rasmussen's behavior model, but also on existed conversation architecture. As shown in Fig. 4.3, the conversation system is divided into three parts, such as skill, rule, and knowledge based conversation system. In the skill based conversation system, daily conversation with reflective, repeated, and short conversation is performed. The rule based conversation system takes place as information support conversation. Here, using the word understanding model, the latest news or weather condition can be requested. In the other situation, information support conversation can be conducted based on time and on the human's condition. The knowledge based conversation is performed based on the understanding of conversation keyword. Through this keyword, conversation is conducted by picking suitable dialogue topic scenario.

From conversation architecture, the conversation system can be structured into detailed parts as shown in Fig. 4.4. The conversation structure is divided into four

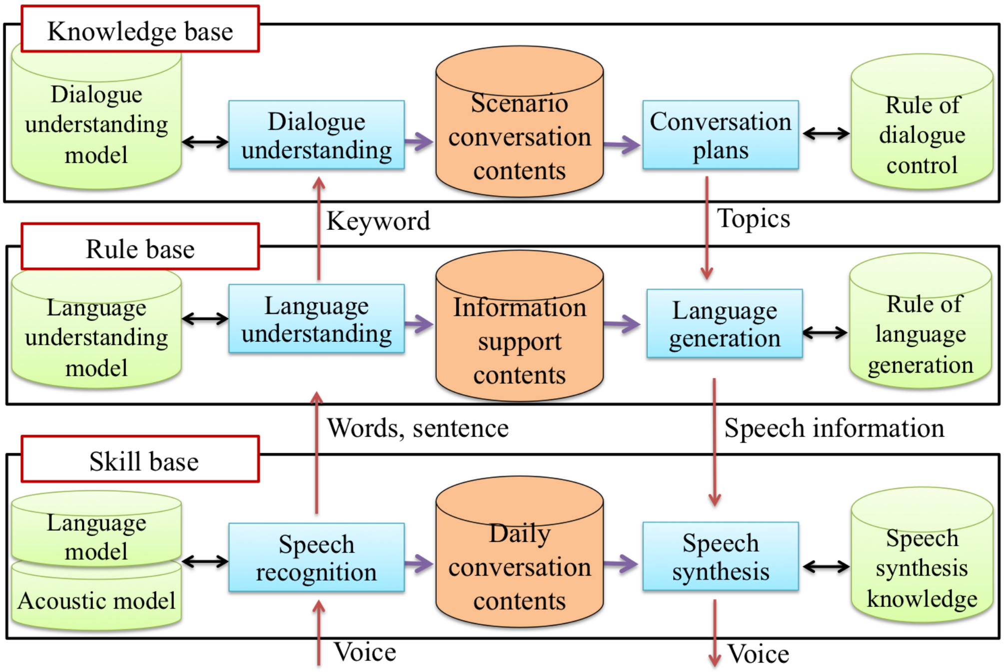



Figure 4.3: Conversation contents architecture

layers, including mode, node1, node2 and contents. The first layer "mode", pointed the conversation mode. While the second and third layer show subcategory of the conversation mode. The last layer mentions the conversation contents. Using the conversation structure, the robot partner can perform the learning process to select suitable conversation contents based on the people's state and time. Figures 4.5 and 4.6 show the daily conversation contents and information conversation contents used in the experiment process. The parameters are defined in the left side of conversation contents column as classifiers in selecting conversation contents.

In the experiment part, as shown in Fig. 4.7, the details of some parameters are defined such as human state, human behavior, human gesture, robot emotion and robot action. For the utterance selection module, I propose a method that composed of two stages: (1) utterance group selection and (2) word or sentence selection. Basically, an utterance group is composed of different words or sentences having the same meaning, e.g., "hello"=hi, hello, ya. First, the utterance group is selected according to the flow of the context in the conversation control module and perceptual information. Next, one word or sentence is stochastically selected from the group according to the state of feelings. The state of feelings corresponds to the utterance group is calculated using spiking neural network.

When the spiking neuron corresponding to the $i$-th utterance group fires, the

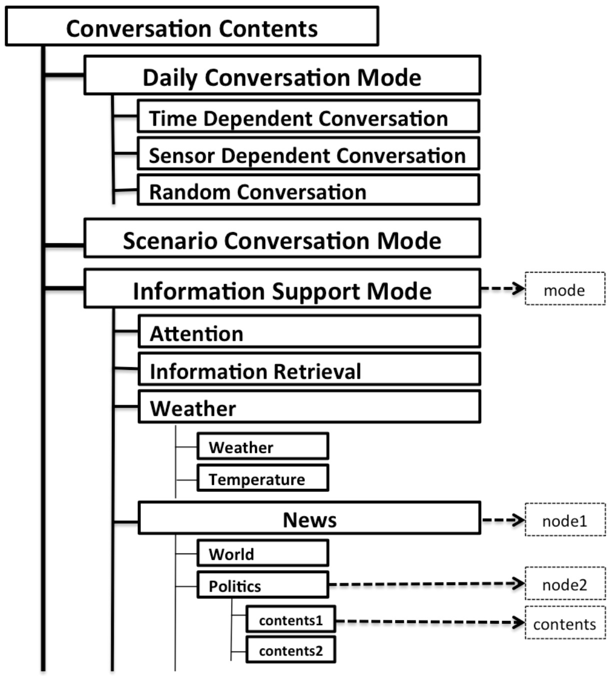



Figure 4.4: Conversation system structure

| contents_id | mode | node1 | node2 | time | h_state | h_behaivor | h_gesture | r_action | speech | contents |
|---|---|---|---|---|---|---|---|---|---|---|
| 11001001000 | 1 | 100 | 100 | 420 | 1 | null | 0 | 1 | hi,hello,morning | Good Morning |
| 11001011000 | 1 | 100 | 101 | 465 | 1 | null | 0 | 0 | null | It's time to take breakfast |
| 11011011002 | 1 | 101 | 101 | 450 | 1 | 13 | 0 | 0 | null | Don't you feel hungry already |
| 11011011003 | 1 | 101 | 101 | 450 | 1 | 13 | 0 | 0 | null | It's good to take breakfast before 9 o'clock |

Figure 4.5: Daily conversation contents

| contents_id | mode | node1 | node2 | time | h_state | h_behaivor | h_gestuer | r_action | speech | contents |
|---|---|---|---|---|---|---|---|---|---|---|
| 21001001000 | 2 | 100 | 100 | 420 | 1 | 1 | 0 | 5 | weather | Today's weather is sunny |
| 21001011000 | 2 | 100 | 101 | 420 | 1 | 1 | 0 | 5 | weather,temperature | Today's maximum temperature is 28 degree |
| 21011001000 | 2 | 101 | 100 | 420 | 1 | 1 | 0 | 5 | news,world | Laos government's plane crash cause death of vice prime minister |
| 21011011000 | 2 | 101 | 101 | 420 | 1 | 1 | 0 | 5 | news,politics | Don't forget to bring towel |
| 21021001001 | 2 | 102 | 100 | 435 | 1 | 14 | 0 | 5 | | The floor is slippery, please watch your step |
| 21021011000 | 2 | 102 | 101 | 495 | 1 | 13 | 0 | 5 | | Don't forget to have a glass of water |

Figure 4.6: Information support contents

| No | human state |
|---|---|
| 0 | nobody in the sensing range |
| 1 | human detection |
| 2 | more people detection |

(a) Human state

| No | robot emotion |
|---|---|
| 0 | happy |
| 1 | sad |
| 2 | fearful |
| 4 | angry |

(b) Robot emotion

| No | human behavior |
|---|---|
| 0 | sleep |
| 1 | get up |
| 11 | walk |
| 12 | launder |
| 13 | open a fridge |
| 14 | get in toilet |
| 15 | sit down |

(c) Human behavior

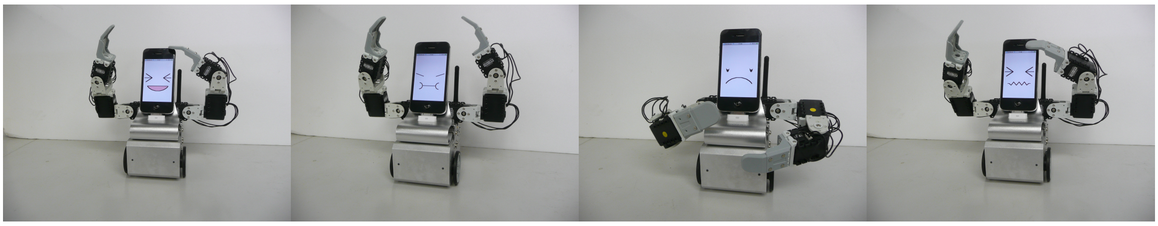
| No | human gesture |
|---|---|
| 0 | no gesture recognition |
| 1 | hello |
| 2 | come here |
| 3 | swing |
| 4 | stop |
| 5 | bye bye |

(d) Human gesture

| No | robot action |
|---|---|
| 0 | no action |
| 1 | hello |
| 2 | come here |
| 3 | swing |
| 4 | bye bye |
| 5 | information support |

(e) Robot action

Figure 4.7: Conversation contents parameters



(a) Pleasure (b) Anger (c) Sadness (d) Fear

Figure 4.8: Facial and gestural expressions

selection strength ($s^G_{i,k}$ ) of the $k$-th words in the $i$-th utterance group related to the $j$-th feeling is calculated by

$$s^G_{i,k} = u^F_j(t) \cdot \exp(-(u^F_j - F^G_{j,i,k})^2). \tag{4.23}$$

After that, the selection probability ($p^G_{i,j}$) is calculated using Boltzmann selection scheme as follows;

$$p^G_{i,j} = \frac{\exp(s^G_{i,k}/T^G)}{\sum_{g=1}^{N^G_i} \exp(s^G_{i,g}/T^G)} \tag{4.24}$$

where $T^G$ is a positive value called the temperature, $N^G_i$ is the number of candidate words in the $i$-th utterance group. According to the equation, when the temperature is high, the robot partner will randomly select utterance words from the $i$-th utterance group. As the temperature decreases, the robot partner deterministically selects the utterance words with high selection strength. At the same time, the robot partner selects hand gesture corresponding to the selected utterance (Fig. 4.8).

## 4.4 Active measurement by robot partner

In Chapter 3, the experimental results showed that the proposed method was able to measure the human behavior in both indoor and outdoor environments flexibly. However, it is difficult to estimate complex human activity by the proposed methods. I propose a method of complementarily using activity information measured by the distributed sensing system and activity information estimated by the robot partner in the feature extraction layer. The measurement components consist of device control, human activity estimation, and environment state estimation (Fig. 4.10). Here, I propose to conduct the human activity estimation as a sub-component. Figure. 4.11 shows the active action measurement algorithm for the fusion of distributed sensor

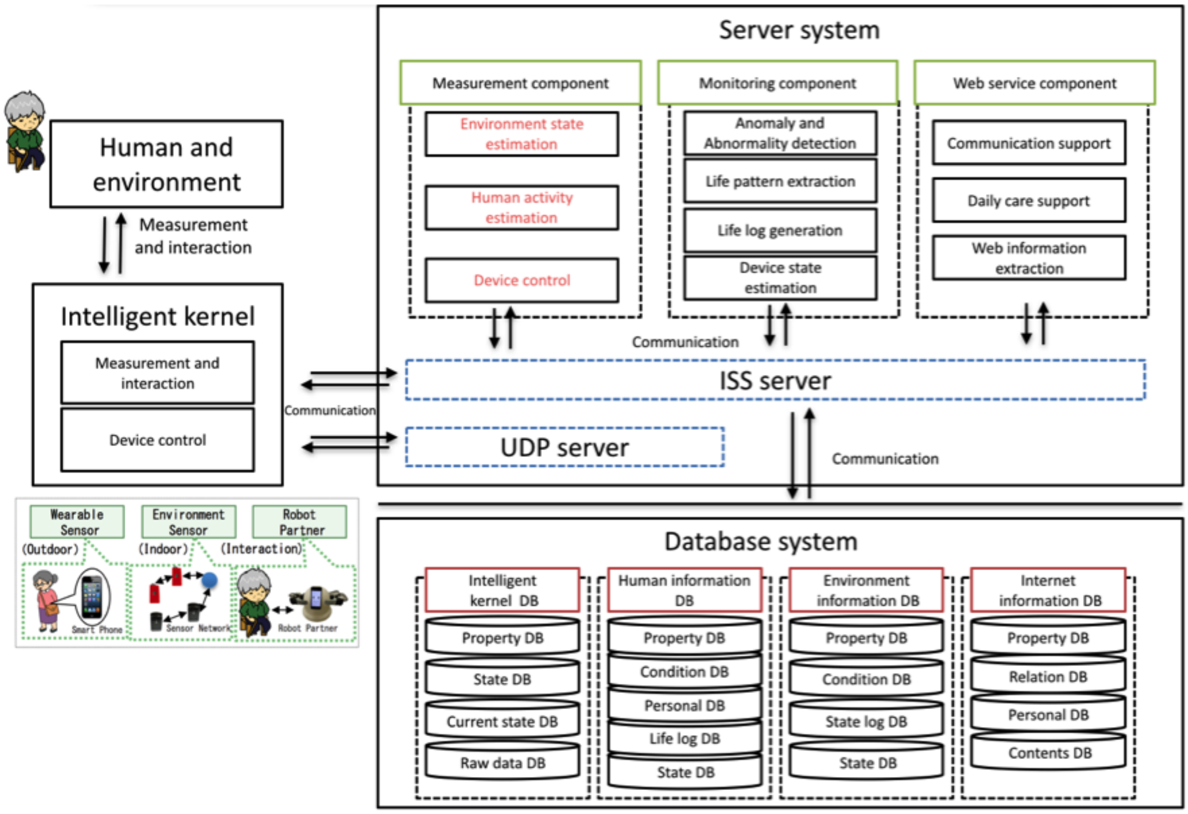



Figure 4.9: System structure of ISS

system and robot partner system. In this algorithm there are some conditions that should be fulfilled such as unknown and sit on chair activity, then the robot partner will start to do active measurement. An example of scenario conversation (human sits on the chair) is show in Table 4.1. This example shows the conversation between robot partner and human, where the human actions were previously registered and the questions are made according to these. After the robot partner checked the human answer, the action is registered into the database.

Table 4.1: Conversation contents example for active measurement

| | |
|---|---|
| Robot partner: | (Question) What are you doing? |
| Human: | Reading book |
| Robot partner: | (Checking) Are you reading book? |
| Human: | Yes |
| Robot partner: | (Finished checking, DB registration) Understood. |

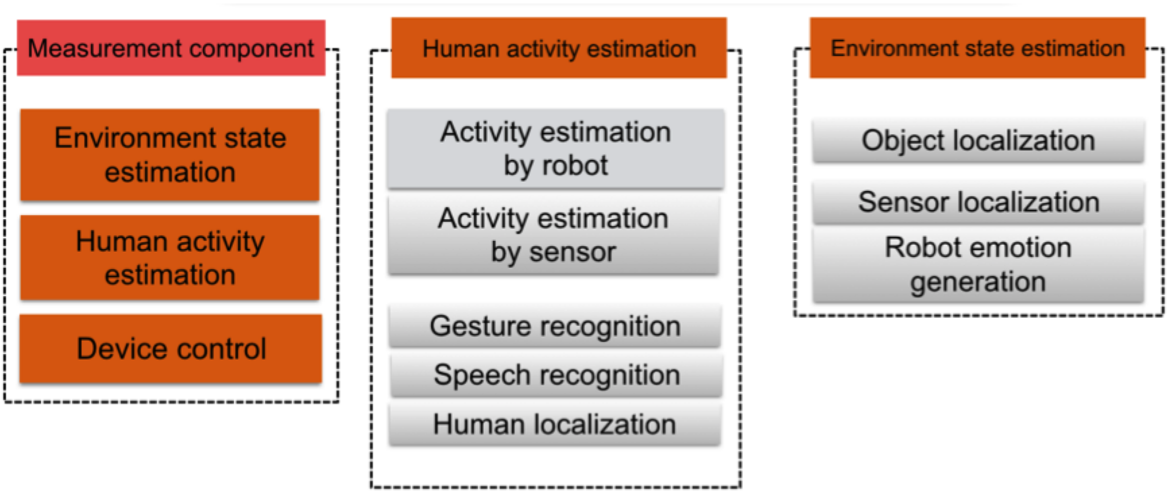



Figure 4.10: Details of measurement component

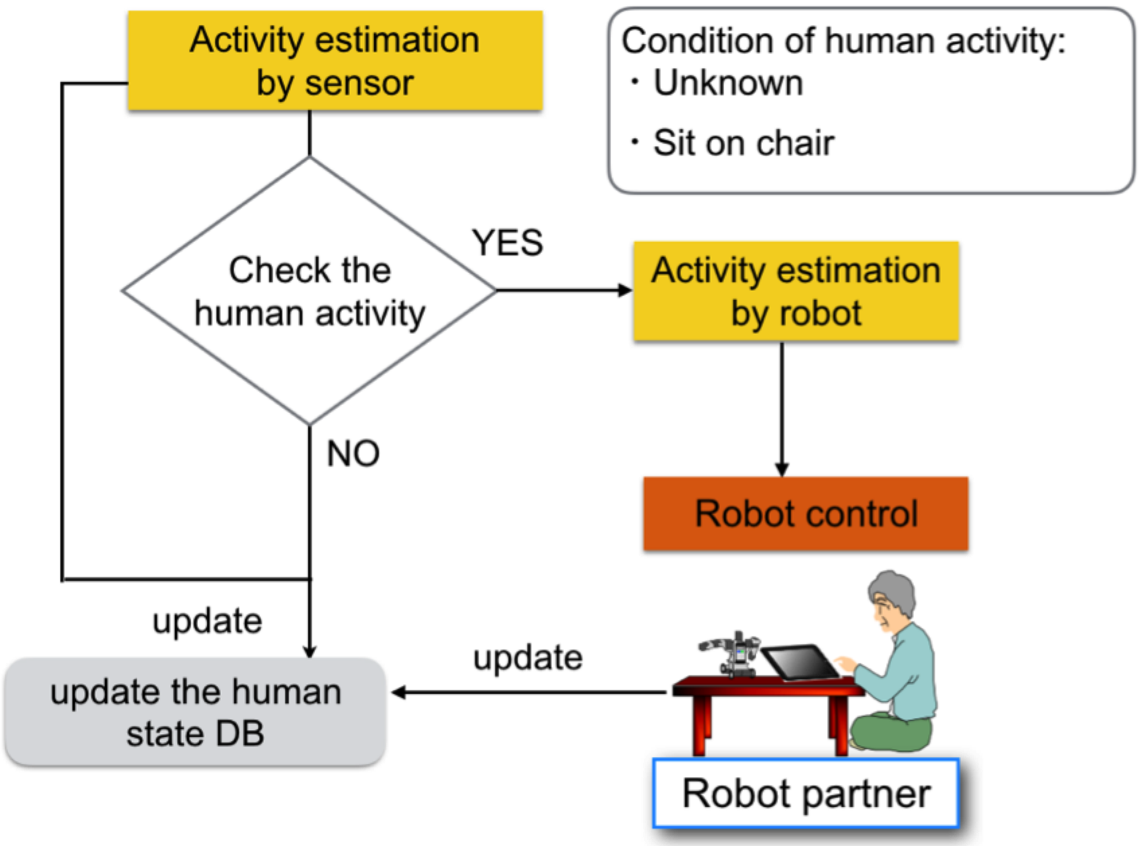



Figure 4.11: Active measurement algorithm

Table 4.2: Contribution parameters from perception input to feeling

| | Pleasure | Sadness | Fear | Anger |
|---|---|---|---|---|
| People Detected | 0.1 | -0.1 | -0.2 | -0.2 |
| Distance to Human | 0.3 | 0 | 0.1 | 0 |
| Human Activity | 0.2 | 0 | 0.1 | 0 |
| Brightness | 0.4 | -0.2 | -0.1 | -0.1 |
| Darkness | -0.2 | 0.2 | 0.4 | 0.1 |
| No People Detected | -0.1 | 0.2 | 0.03 | 0.01 |

## 4.5 Experimental Results

I divided the experiments into four case studies. In the first case study, I want to investigate and validate the proposed emotional model through the computation of environmental conditions and human behaviors (movement and distance) effects to robot partner. For the second case study, I conducted the experiment much further about the robot partner's emotions in conjunction with human gesture recognition. In the third case study, I investigate the feasibility of my integrated system, starting from data collection through sensors, data processing, and conversation system. Finally, in the fourth case study I use a robot partner called Palro.

### 4.5.1 Case Study 1

In this experiment, I considered the effects of environmental conditions during the communication with the robot partner. Table 4.2 shows the contribution parameters from perception input to feeling. These parameters are acquired as the optimum result of the trial and error parameter settings. Figure 4.12 displays the communication process between the robot partner and the person in the experiment. In the experiment, as the environment brightness is high, the robot partner is in the state of happiness (a,c,e) as depicted in Fig. 4.13. In addition, when the person started to make some action (moving action), the happiness value increases faster than before, especially when the person walked near the desk (d,f). Meanwhile, the fear value increases when the room changed to dark (b). From here, it can be noticed that linkage between the emotion model and environmental changes. This is a very crucial aspect as the robot partner need to be very sensitive to the environmental changes, when communicating with the human.

Figure 4.12: Snapshots of the first experiment

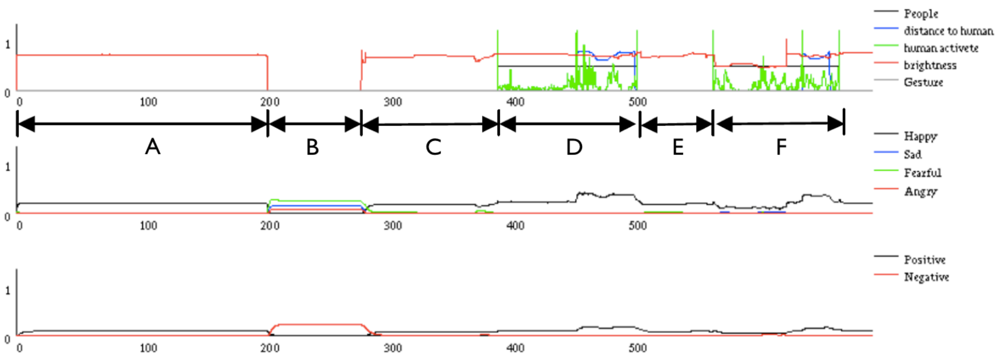



Figure 4.13: The change of states in the first experiment

### 4.5.2 Case Study 2

In the second experiment, gesture recognition is considered in the communication with the robot partner. Through gesture recognition, the robot partner is expected to understand the human behavior using non-verbal communication. On contrary with the first experiment, in this experiment I added some contribution parameters for gesture as shown in Table 4.3. Figure 4.14 illustrates the environmental conditions including the changing of the room brightness and the human gesture recognition process. Figure 4.15 depicts the experiment results through a graph representation. In this graph, the fear value increased when the room was dark (a). The happiness value increases and the fear value decreases when the person comes into the room and turns on the light (b). When the person entered into the view, the happiness value increased (c), however when the robot partner detected the [stop] gesture, the anger value increased, on the contrary the happiness value started to decrease (d).

### 4.5.3 Case Study 3

In the case studies 1 and 2, based on the changes of the environmental condition and human gesture in informationally structured space, I realized the robot partner's

Table 4.3: Contribution parameters from perception input to feeling

| | Pleasure | Sadness | Fear | Anger |
|---|---|---|---|---|
| People Detected | 0.1 | -0.1 | -0.2 | -0.2 |
| Distance to Human | 0.3 | 0 | 0.1 | 0 |
| Human Activity | 0.2 | 0 | 0.1 | 0 |
| Brightness | 0.4 | -0.2 | -0.1 | -0.1 |
| Darkness | -0.2 | 0.2 | 0.4 | 0.1 |
| No People Detected | -0.1 | 0.2 | 0.03 | 0.01 |
| Gesture (Bye Bye) | -0.1 | 0.5 | 0.1 | 0 |
| Gesture (Hello) | 0.4 | 0 | -0.1 | -0.1 |
| Gesture (Stop) | -0.3 | 0.05 | 0.03 | 0.5 |
| Gesture (Swing) | 0.4 | 0 | -0.1 | -0.1 |

Figure 4.14: Snapshots of the second experiment

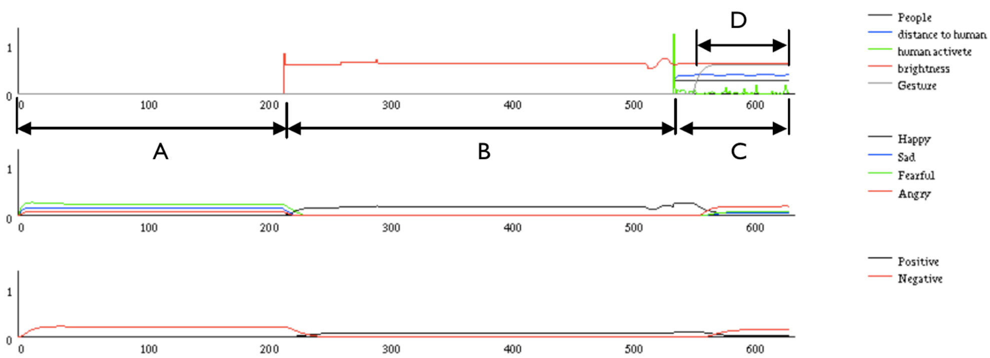



Figure 4.15: The change of states in the second experiment


Figure 4.16: Experimental room

emotion building process. Through the input data for the emotional model, the robot partner can express its emotions based on the current condition. In this experiment, as an information support system I realized multi-modal communication between the robot partner and the human based on environmental conditions. As shown in Fig. 4.16 I built an environmental model of elderly people's house for the experiment. In this room, I installed SunSPOT in chair, toilet, and refrigerator. In addition, I also installed wireless optical oscillo-sensor in the bed and Microsoft Kinect above the bookshelf.

The experiment is conducted to investigate the information support by the robot partner based on human behavior. Starting from getting up from the bed, sitting on the chair, going to the toilet and preparing breakfast. The snapshot of the experiment can be seen in Fig. 4.17. The human behavior record which is saved in the human state database is expressed in Fig. 4.18, where it can be seen that at (a) shows the human when getting up from the bed, (b) shows human is sitting on a chair, (c) shows human speaking with robot, and (e) shows entering the toilet. The robot partner uses this information as a reference in conducting communication with the human. Figure 4.19 shows the sample of human state database. The information in this figure is used to recognize the human's state. Besides human state, human behavior, human gesture, and robot emotion as the main attributes, and the other attributes such as season, year, month, day, week, and time (minutes) are used to recognize personal life time and build the personal model of the human.

(a) Getting up from bed (b) Sitting on chair (c) Speaking with robot partner (d) Entering toilet

Figure 4.17: Snapshots of the third experiment

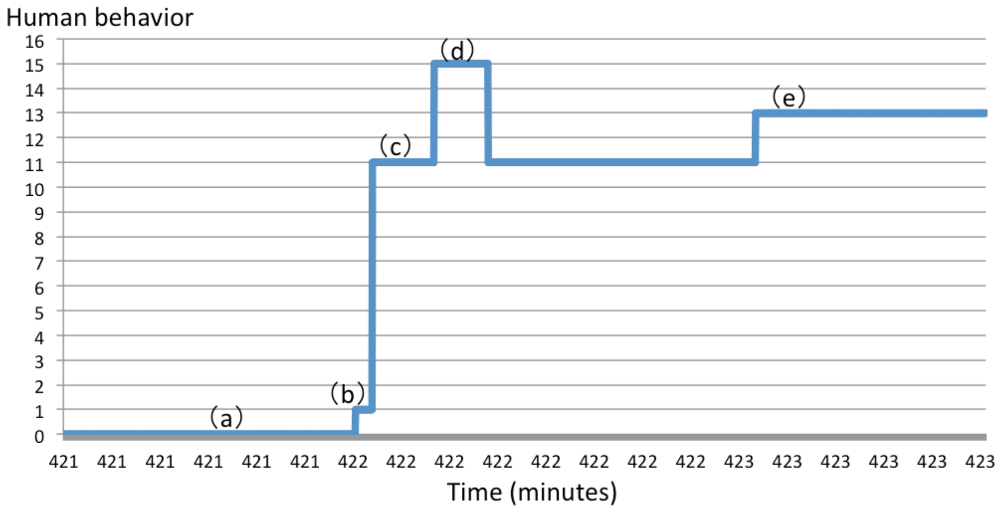



Figure 4.18: Graph of human state log

| id | timestamp | h_state | h_behavior | h_gesture | r_emotion | season | year | month | week | day | time |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 8282 | 2014/6/21 7:01 | 1 | 0 | 0 | 0 | 2 | 2014 | 6 | 6 | 21 | 421 |
| 8283 | 2014/6/21 7:01 | 1 | 0 | 0 | 0 | 2 | 2014 | 6 | 6 | 21 | 421 |

Figure 4.19: Human state log

| timestamp | season | year | month | day | week | time | h_state | h_behavior | h_gestrue | r_motion | r_action | mode | node1 | node2 | contents_id |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2014/6/21 7:02 | 2 | 2014 | 6 | 21 | 6 | 421 | 1 | 1 | 0 | 1 | 1 | 1 | 101 | 100 | 11011001007 |
| 2014/6/21 7:02 | 2 | 2014 | 6 | 21 | 6 | 421 | 1 | 15 | 0 | 1 | 0 | 1 | 101 | 101 | 11011011013 |
| 2014/6/21 7:02 | 2 | 2014 | 6 | 21 | 6 | 420 | 1 | 15 | 0 | 1 | 5 | 2 | 100 | 100 | 21001001000 |
| 2014/6/21 7:03 | 2 | 2014 | 6 | 21 | 6 | 422 | 1 | 13 | 0 | 1 | 0 | 1 | 100 | 101 | 11001011007 |

Figure 4.20: Conversation log in database

| | Conversation | Human Behavior |
|---|---|---|
| robot partner: | hi! | get up from the bed |
| human: | morning! | get up from the bed |
| robot partner: | where are you going today? | sit on the carpet |
| human: | I haven't decided yet | sit on the carpet |
| human: | how about today's weather | sit on the carpet while asking weather information |
| robot partner: | today is cloudy | sit on the carpet while asking weather information |
| robot partner: | let's have breakfast together | open the refrigerator |

Figure 4.21: Conversation log

Figure 4.20 shows data saving in the conversation log database during the experiment. In order to conduct learning process in the conversation process, human and environmental state connected with time become the main attribute. For the time attribute, I use season, year, month, day, week, and time (minutes). Meanwhile, human state, human behavior, human gesture, robot motion, and robot action are used as the attributes to decide utterances. For classifying the conversation contents, I define mode, node1, node2, and content id for the attributes. The conversation contents of the conversation log database is depicted in Fig. 4.21.

### 4.5.4 Case Study 4

In this experiment, I use a robot partner developed by Fujisoft called Palro as depicted in Fig. 4.22. In this experiment, the human activity is presented in Fig. 4.23. The sub-figures show human conditions as follows, (a) walk and get inside living room, (b) sit on chair, (c) reading book, and (d) talk with robot. Figure 4.24 shows the distribution of active estimation by robot partner and passive estimation by distributed measurement system. After the human entered the living room, he sat down on the chair, and after a while active conversation was conducted through the robot partner to check the activity. By using the distributed measurement system, I compare the predicted activity with the real activity checked by the robot partner.

Figure 4.22: Robot partner: Palro

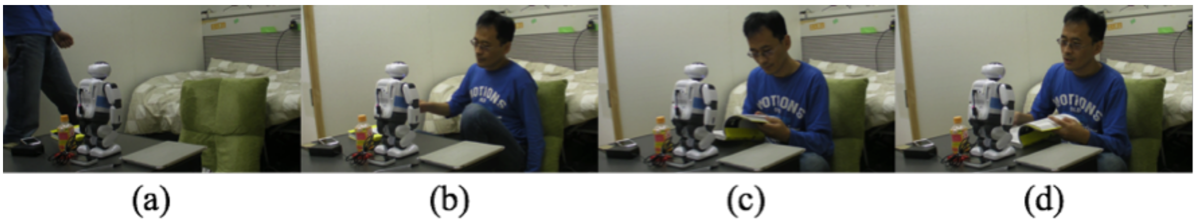



Figure 4.23: Snapshots of the fourth experiment

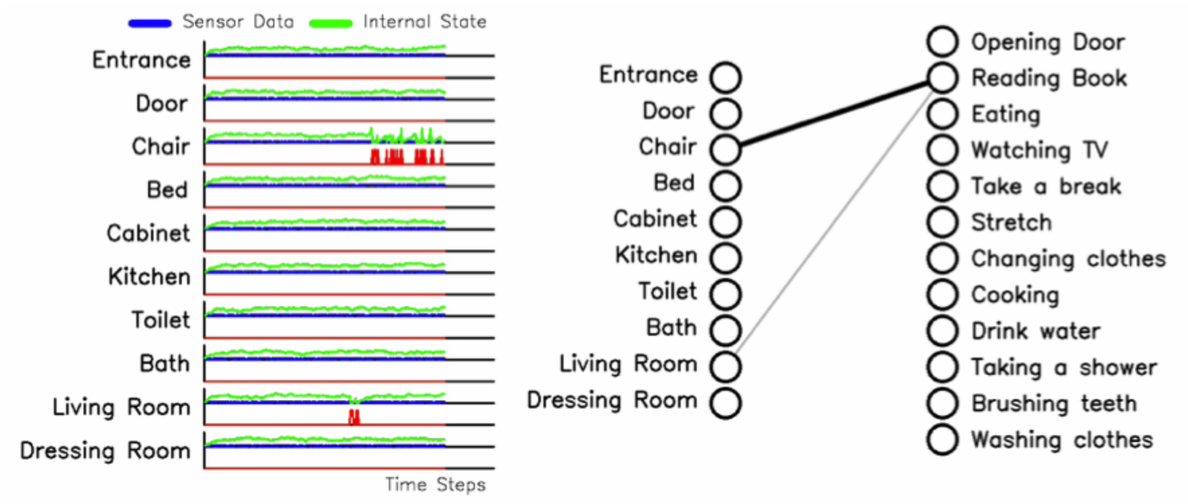



Figure 4.24: The change of states in the fourth experiment

### 4.5.5 Discussions

Through the experimental results, I can conclude that, in the case study 1, feeling is affected dominantly by the brightness of the room. As the perception parameter value on brightness is high, intuitively I could guess, that the brightness as well as the darkness of the room will give important effect on the robot partner's feelings. On the other hand, although human movement and distance also give some effect to the feeling, but still couldn't replace the dominance of room brightness effect.

In the case study 2, basically the robot partner's feelings are affected by the room's brightness. However, when the robot partner recognized a particular human gesture, the dominance of the room's brightness is replaced by human gesture resulting in the changing of the robot partner's feeling. In the case study 3, the realization of informationally structured space is conducted. Here, I took the experiment for several days to investigate the human life cycle activity. The information acquired from and outputted into environment is saved in the database, and used for recognizing the state and learning in the conversation process. In the result, after recognizing the situation, the robot partner could conduct suitable conversation with the human. In the case study 4, the experimental results through time dependent conversation show that the robot partner can estimate human behavior that is difficult to be measured by the distributed sensing system.

As mentioned in the introduction, there are some research methods that aim similar goals. For example [116] was focusing their research on cognitive environment, while [106] was focusing theirs in human recognition technology. Meanwhile [10] conducted their research which related to human recognition and emotional consideration. In contrary, my proposed method including the cognitive environment, human recognition, and emotional consideration realizes the robot partner's reaction through various input data. For example, when the human gets up from the bed, the robot partner with its environmental cognitive ability for recognizing human condition gives some reactions such as utterance and particular gesture to reinforce the utterance contents. Moreover, with the combination of gesture recognition and emotional model as it can be seen in the experimental result, the communication between the robot partner and the human can be realized in a more natural and interesting way. More details about the comparison of the methods can be seen in Table 4.4.

Table 4.4: Comparison to other methods

| Methods | Cognitive Environment | Human Recognition | Considering Emotions |
|---|---|---|---|
| [116] | √ | – | – |
| [106] | – | √ | – |
| [13] | – | √ | – |
| [11] | – | √ | – |
| [10] | – | √ | √ |
| [14] | √ | √ | – |
| Proposed Method | √ | √ | √ |

## 4.6 Summary

This chapter discussed the actualization of natural communication between a robot partner and a human based on the application of relevance theory in the mutual recognition space. In order to realize this, I built the architecture in informationally structured space as the basis of my research along with the database system, which supports the informationally structured space in transferring data to and from the environment. While discussing informationally structured space, I also explained the elements constructed my system such as sensor network, web system, and robot partner including gesture recognition and emotional model. In further, the conversational architecture which allows the natural communication to be realized was deeply discussed. This includes the discussion of database structure and conversation selection algorithm.

In terms of the theoretical contribution, the proposed system is built based on four theories as follows: (a) Relevance Theory - Realizing natural communication between human and robot. Natural communication can be realized when robot can understand human intention or thought. I implemented the theory of relevance proposed by [110] to build mutual cognitive environment between human and robot partner into my system, which called Informationally Structured Space to handle this problem. According to [110], relevance theory is very useful to discuss the multimodal communication, where each person has his or her own cognitive environment that make their communication restricted. Therefore, usually humans use their utterances or gestures to expand their cognitive environment by extracting person's attention into specific target object, event, or person. When human's cognitive environment became wider, they can share each other intention or thought. The implementation of this theory into the system can be observed in the structure of database. (b) Rasmussen's Behavior Theory - For conducting daily conversation with human, robot partner has to have enough knowledge and contents. However, to get enough knowledge and contents, the conversation system contents and task will become bigger. At this state, flexibility and simplicity of the system become a new issue. To deal with this, I proposed a new conversation system architecture based on Rasmussen's behavior model. Using this model, I divided the conversation into three types based on complexity. (c) Computational Intelligence (Soft Computing) - In the gesture recognition, I implemented structured learning involving growing neural gas for information extraction and spiking neural network to recognize the gesture, and finally (d) Boltzmann Selection - For acquiring non-monotonic or lively conversation between human and robot partner I used Boltzmann selection by controlling the value of temperature to perform word selection when the robot partner communicates with the human.

In terms of practical implementation, the development of sensor network installed in nursing home to support caregivers in conducting monitoring for elderly people has been increased. However, in order to give information support and encourage elderly people for social activities, mutual cognitive environment between the human and the robot must be built in order to realize natural communication. The implementation of relevance theory and Rasmussen's model into this system has been discussed in this chapter. The experimental results explained the capability of the proposed method to be applied in the real world. In addition, since a low cost robot partner using iPhone is introduced as a mainframe in the proposed system, this system can be applied in the real world in a short time as an advanced elderly people monitoring system.

# Chapter 5

# Informationally Structured Space for Monitoring

## 5.1 Monitoring Layer in Informationally Structured Space

For health care monitoring and anomaly detection, mainly there are two kinds of approches so far. One is monitoring the human behavior and life pattern [2, 16, 108, 132, 113, 86], another one is monitoring the human's bio-signal and motion [2, 90, 8]. The anomaly detection can be divided into two categories: short-term and long-term anomaly detection. There are several data mining techniques to detect anomaly: classification based, clustering based, and statistical based approaches. The human behavior and life pattern monitoring can mainly be divided into two categories from the measurement point of view: noncomputer vision-based methods and computer vision-based methods.

This chapter discusses informationally structured space for monitoring layer in order to realize long-term human behavior measurement. First, I propose daily life model estimation method using fuzzy modeling from long-term human behavior information. Moreover, I construct live models with different granularities by day, week and so on. I also propose a method that specifies the difference between the life models. The experimental result shows the changing of the life models. In order to realize flexible adaptation to the changing of the environmental condition and system configuration, informationally structured space can flexibly change the way of signal processing and the network structure in the sensing layer, according to the access state of the devices, in the case of some events such as a device is broken or the battery is low. The experiments show how the informationally structured space can

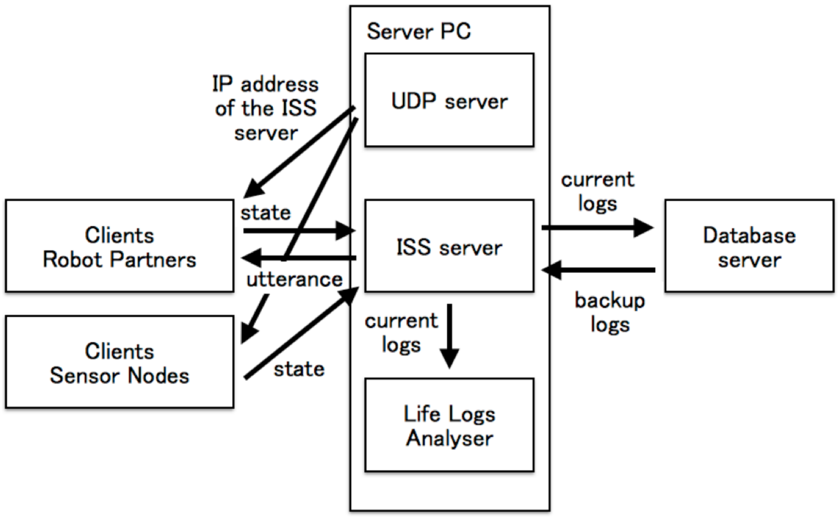



Figure 5.1: Information flow among robot partners, sensor nodes and ISS

flexibly handle such situations. Additionally, I develop a simulator system to imitate the informationally structured space, and I use this simulator system to perform numerical experiments in order to show the effectiveness of the proposed method.

## 5.2 Daily Life Model Estimation

### 5.2.1 Monitoring System Based on Informationally Structured Space

I will discuss how to use ISS from the viewpoint of bottom-up access and top-down access. ISS can be considered as a typical client-server system where a robot partner or sensor network device is a client (physical node) shown in Fig. 5.1. A physical node uploads information to the ISS server and downloads information from the ISS server. This kind of local information processing is considered as the bottom-up access. On the other hand, the ISS can control or manage each physical node from the global point of view. For example, the ISS can set the sampling interval of each sensor network device according to human states and behaviors in the perception level. This kind of global information processing is considered as the top-down access.

An example of ISS can be shown in Fig. 5.3. In this example, I use Sun SPOT and Microsoft Kinect as sensor nodes expressed as SS and Kin, respectively in Fig. 5.3. Furthermore, I use two robot partners named iPhonoid and iPadrone. When a sensor node is activated, sensor node receives the IP address of the ISS server by UDP server, and connects to the ISS server (see Fig. 5.1). At that time, the sensor node is added to the activated sensor list of the ISS, its corresponding state is shown in the activation state (Fig. 5.3a). If each sensor node detects a person, its corresponding human position is highlighted (Fig. 5.3b). In this example, the labels of Bed-3 and BedRoom-9 are highlighted , because SS-3 and Kin-9 detect a person. Furthermore, I can use the sensors that the robot partners equipped with. The sensing range of the camera of a robot partner is divided into right area (R-Right) and left area (R-Left). If a camera detects a person, its corresponding human position is highlighted (Fig. 5.3c). In this example, the labels of Walking-3, Sleeping-5, and Interacting-6 are highlighted, because the person just moved to the bed room from the kitchen area.

Basically, elderly care is done by a time series of human life logs. The anomaly of elderly people can be divided into cognitive anomaly and physical anomaly. Human life logs are extracted by human localization, behavior recognition, and voice recognition (Fig. 5.2). Sensor nodes can specify the human position in the house. Here we assume a one-room apartment, and use the sensor nodes at the entrance door, doorway of the living room, chair in the living room, bed, cabinet, kitchen, toilet, and bath room (Fig. 5.4a).

I use a simple spiking neuron to estimate human position [16]. For sensor and information fusion one can refer to my previous work [121, 67]. The redundant data are pre-processed before the learning stage. If the spike output is done, the monitoring system detects a person at its corresponding position. Human movement transition probability is calculated by the time series of life log data (Fig. 5.4b). The human life pattern can be analyzed by tracing state transition of human behaviors, while the measurement errors can be detected by tracing the simultaneous firing of sensor nodes in the house. The monitoring system is redundant, because several sensors often cover the same area with different resolutions. For example, the camera of a robot partner monitors the same area with the Kinect sensor. If a breakdown or error happens to a sensor node, we can detect different pattern in the simultaneous firing of sensor nodes. Figure 5.4c shows an example of simultaneous firing patterns. If the simultaneous firing occurs in the sensor node, its corresponding connection is highlighted strongly.

### 5.2.2 Estimation Method of Daily Life Model

The daily temporal pattern is extracted from the time series of life log data. Each behavior in the daily temporal pattern is represented by the set of Gaussian membership functions over time,

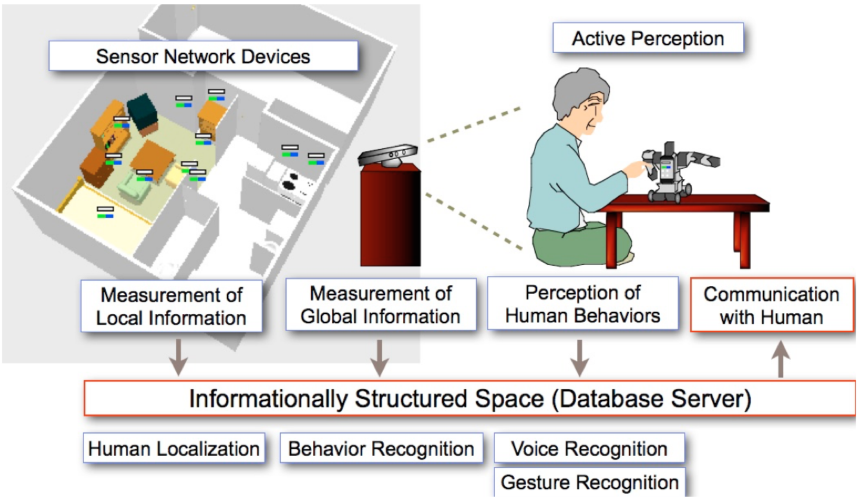



Figure 5.2: Informationally structured space for elderly care in a room

$$\mu_{A_{i,j}}(t) = \exp\left(-\frac{(t-a_{i,j})^2}{b_{i,j}^2}\right) \tag{5.1}$$

where $t$ is the input time; $a_{i,j}$ and $b_{i,j}$ are the central value and width of the $j$th class of the $i$th behavior, respectively. When the $i$-th behavior is observed, if $\mu_{A_{i,j}}(t) \leq \theta$ , then we use the simple update rule,

$$\begin{cases} a_{i,j} \leftarrow (1-\alpha)a_{i,j} + a \cdot (x_i + 0.5 \cdot w_i) \\ b_{i,j} \leftarrow (1-\alpha)b_{i,j} + 0.5 \cdot \alpha \cdot w_i \end{cases} \tag{5.2}$$

where $x_i$ is the starting time and $w_i$ are the consumption time of the $i$th behavior; $\alpha$ and $\theta$ is the learning rate and threshold, respectively. If not, a new cluster is added to the temporal behavior pattern clusters. Each behavior temporal life pattern is calculated by the following equation:

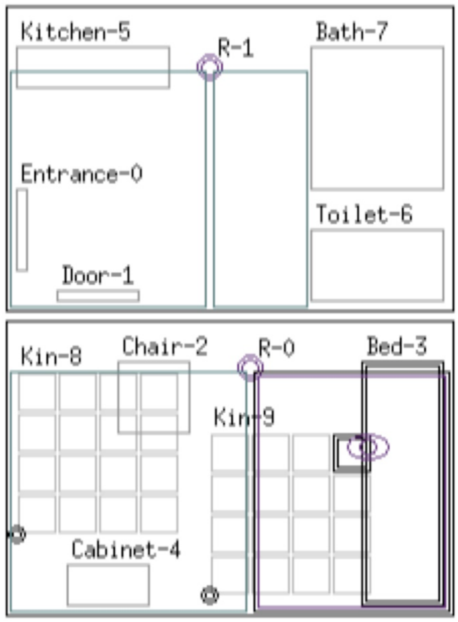
$$\mu_{A_{i,j}}(t) = \frac{T_{i,j}}{\max T_i} \exp\left(-\frac{(t-a_{i,j})^2}{b_{i,j}^2}\right) \tag{5.3}$$

where $Ti,j$ is the selection times of the $j$th cluster of the $i$th behavior; $\max T_i$ is the maximal selection times. Therefore, the height of Gaussian membership function is the relative selection frequency.

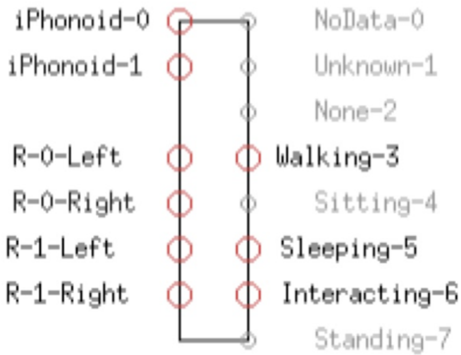
## 5.3 Sensor Failure Detection in Monitoring Layer

Sensor failure can happen because of hardware or software problem. Here, I will propose a method to detect sensor error using monitoring layer (Fig. 5.5). Since


(a) Activation states

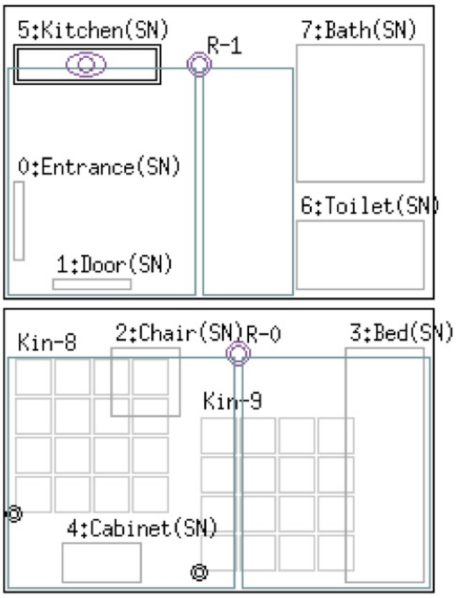



(b) Visualization in informationally structured space

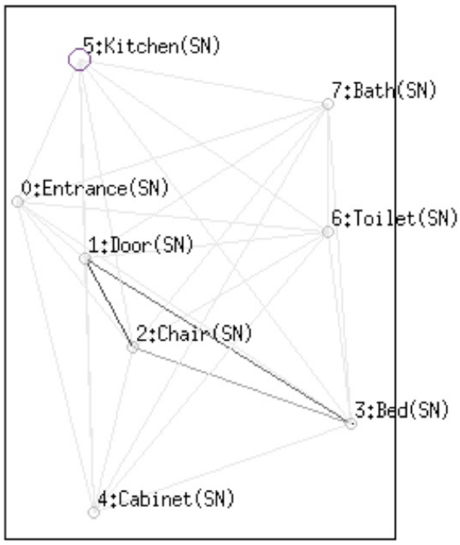



(c) States of robot partners

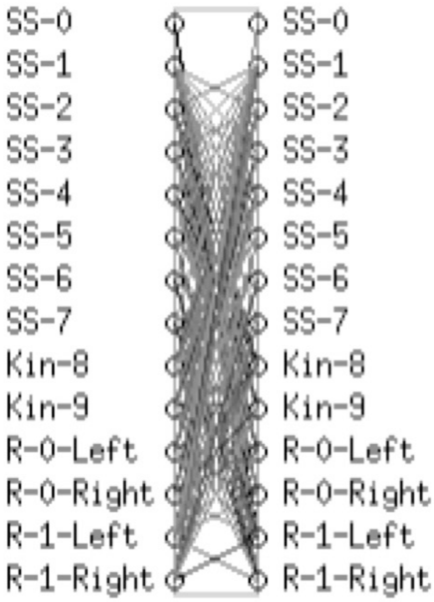
Figure 5.3: Connectivity of sensor nodes with informationally structured space




(a) Monitoring state




(b) Human movement transition




(c) Simultaneous firing of sensor nodes in a house

Figure 5.4: Visualization based on informationally structured space

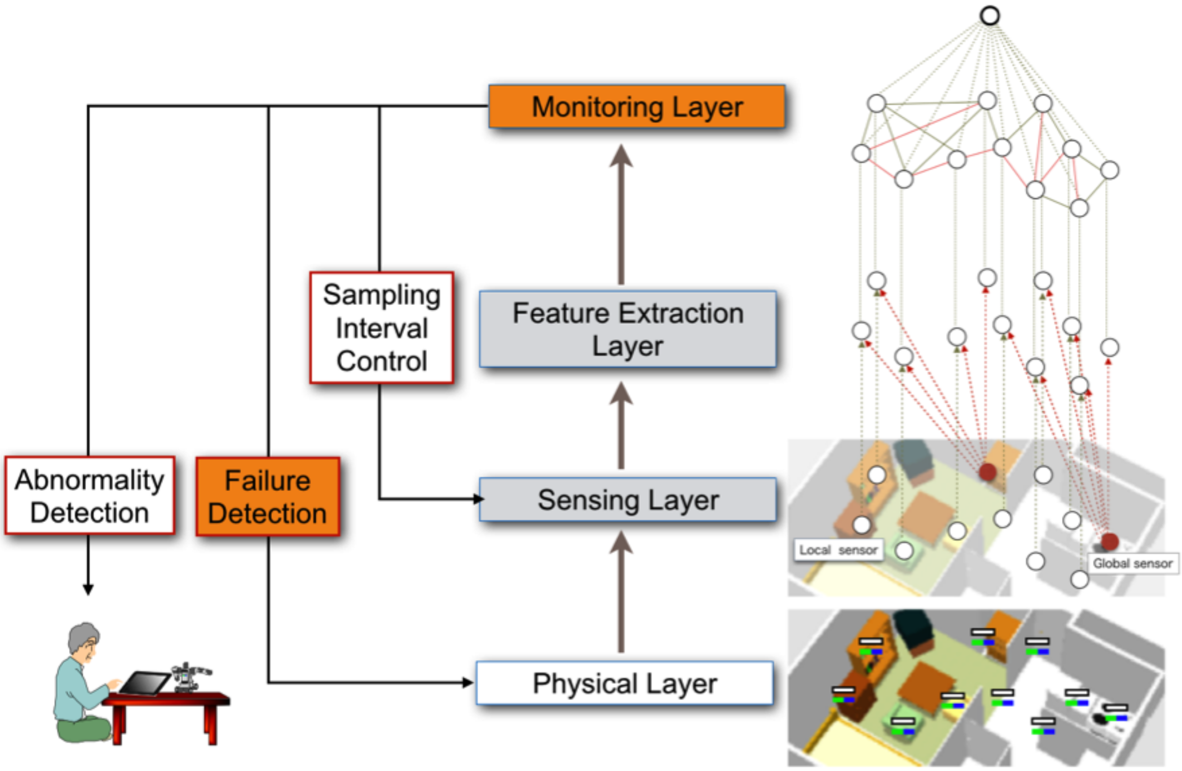



Figure 5.5: Sensor failure detection in monitoring layer

previously registered activity and sensors have correlation, when the time series firing pattern between sensor node changes, the possibility of sensor failure is detected. For example, if the neuron of 3D image distance sensor (KINECT) fires when the human sits on the chair, then the neuron of sensor node attached to chair fires, if there is no sensor error. The sensor error detection algorithm is presented in Fig. 5.6. In informationally structured space, when the changes of human activities are detected, the firing of sensors correlated with these activities are checked. The sensors which are not firing can have a detected error.

## 5.4 Sensor Sampling Interval Control in Monitoring Layer

In a real environment, we need to consider energy consumption and computation cost to operate an activity measurement system. In order to increase the activity precision accuracy, we need to control the sensor sampling adjusted to the activity. For example, when human activities such as sleep, sit on chair, and get out home are investigated, we need to measure only sensors related to these activities. We do not

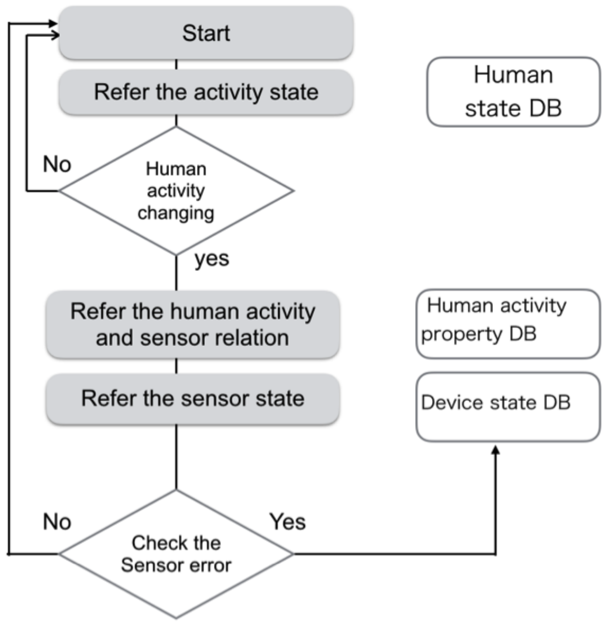



Figure 5.6: Algorithm of sensor failure detection

need to measure others sensors. In order these processes to be operated inside the monitoring layer (Fig. 5.7), I conducted rule based sensor sampling control on human activities (sleep, sit on chair, and get out home). Figure 5.8 shows the real sensor sampling control algorithm. The system can control the sensor sampling interval based on the current human state and the relation between human activities and sensors in ISS.

## 5.5 Estimation of Abnormal State

Previously, abnormality detection sensor systems were proposed, for example by Van Laerhoven et al. [71] and by John Kemp et al. [55]. These approaches were used to measure abnormal activities in the environment but noise could cause these systems to create false warning signal. Therefore, it is important to have human-robot communication before identifying the situation as abnormal situation. In the proposed abnormality detection system, the system will detect two different types of abnormal conditions in the elderly people's daily activities: 1) The system will detect the short-term abnormal activities. For example, if an elderly person get a heart attack or stroke, he or she needs immediate healthcare for stabilizing his or her condition. 2) The system also will detect long-term abnormal activities where the elderly people daily activities pattern changes. For example, the elderly people frequently sleep in late hours. The long-term abnormal activities are important information for early

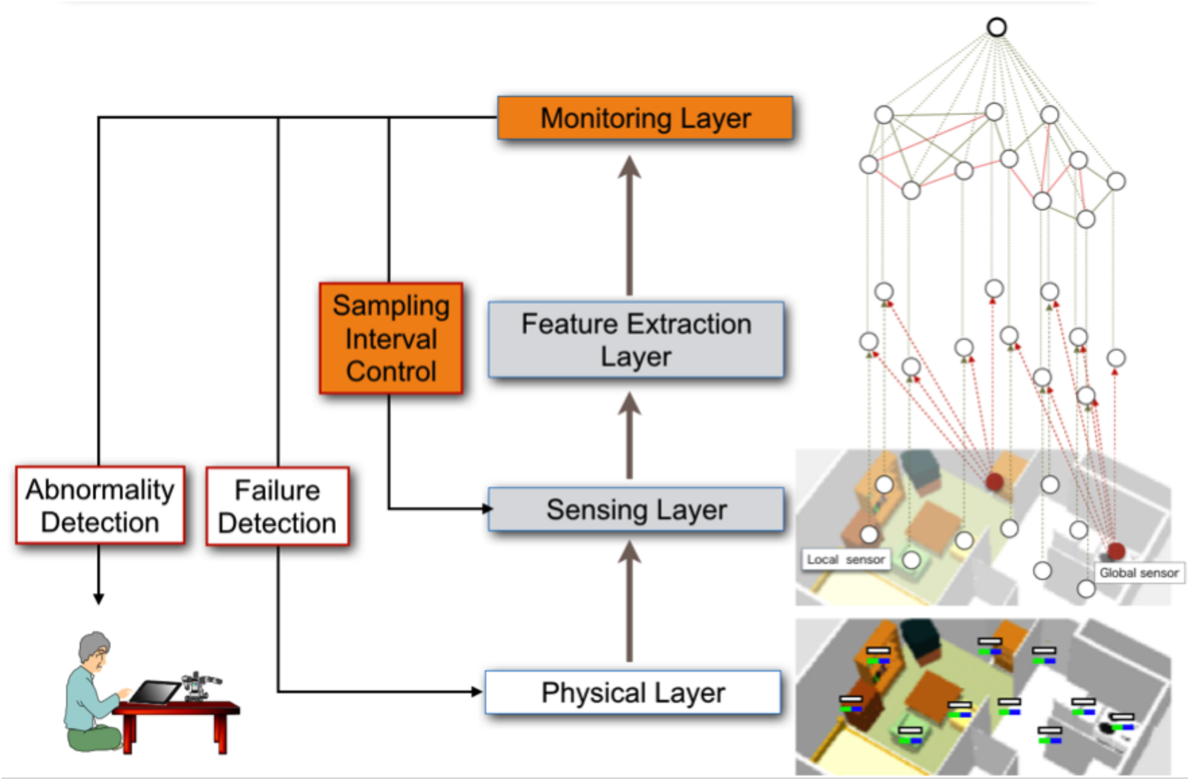



Figure 5.7: Sensor sampling interval control in monitoring layer

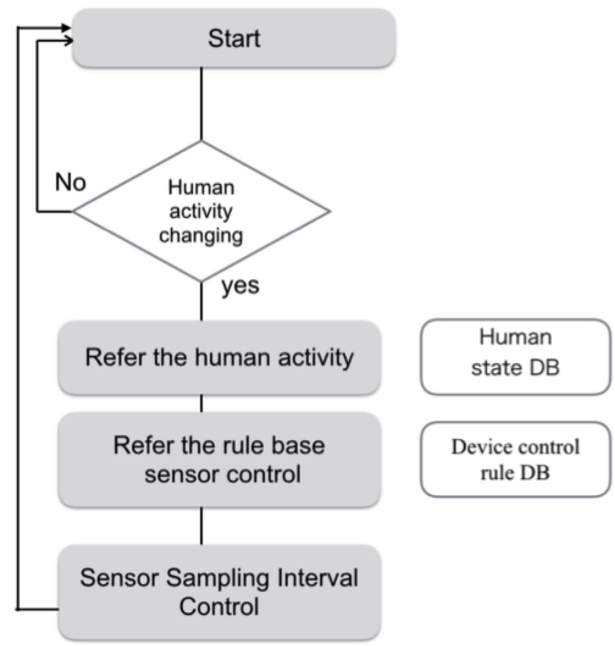



Figure 5.8: Algorithm of sensor sampling interval control

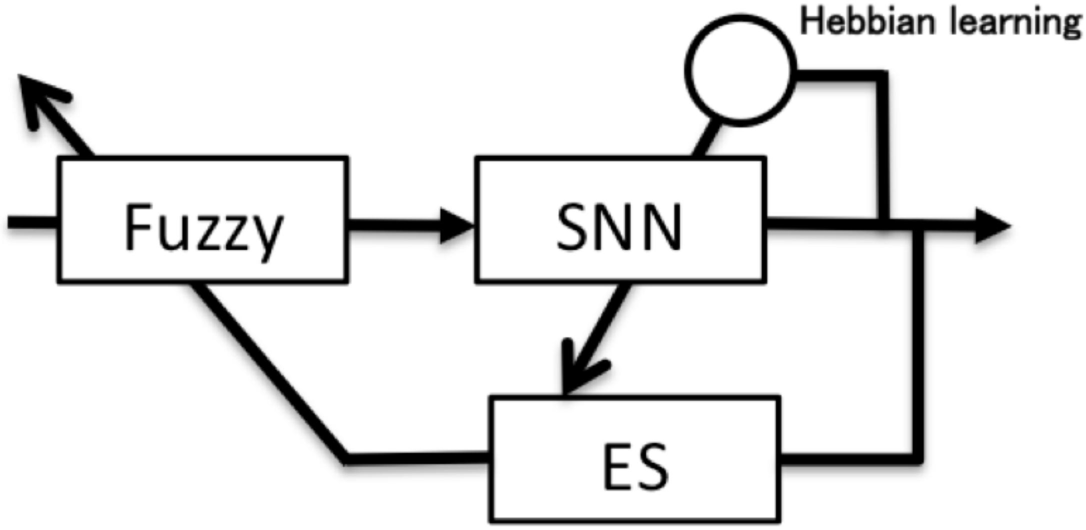



Figure 5.9: Fuzzy spiking neural network model

Table 5.1: Human Behavior

| No. | Human behavior |
|---|---|
| 0 | sleep |
| 1 | get up |
| 2 | rest |
| 3 | go to toilet |
| 4 | have breakfast |
| 5 | have lunch |
| 6 | have dinner |
| 7 | get home |
| 8 | out home |

prevention of any major health problem. Medical health consultant can act on the early warning on elderly people daily life activities pattern changes.

After an abnormal activity is detected, the proposed abnormality detection system will trigger the robot partner to ask questions to the elderly people. For example, the robot will ask the elderly people questions such as “Are you okay?”, “Do you need to call for help?”.

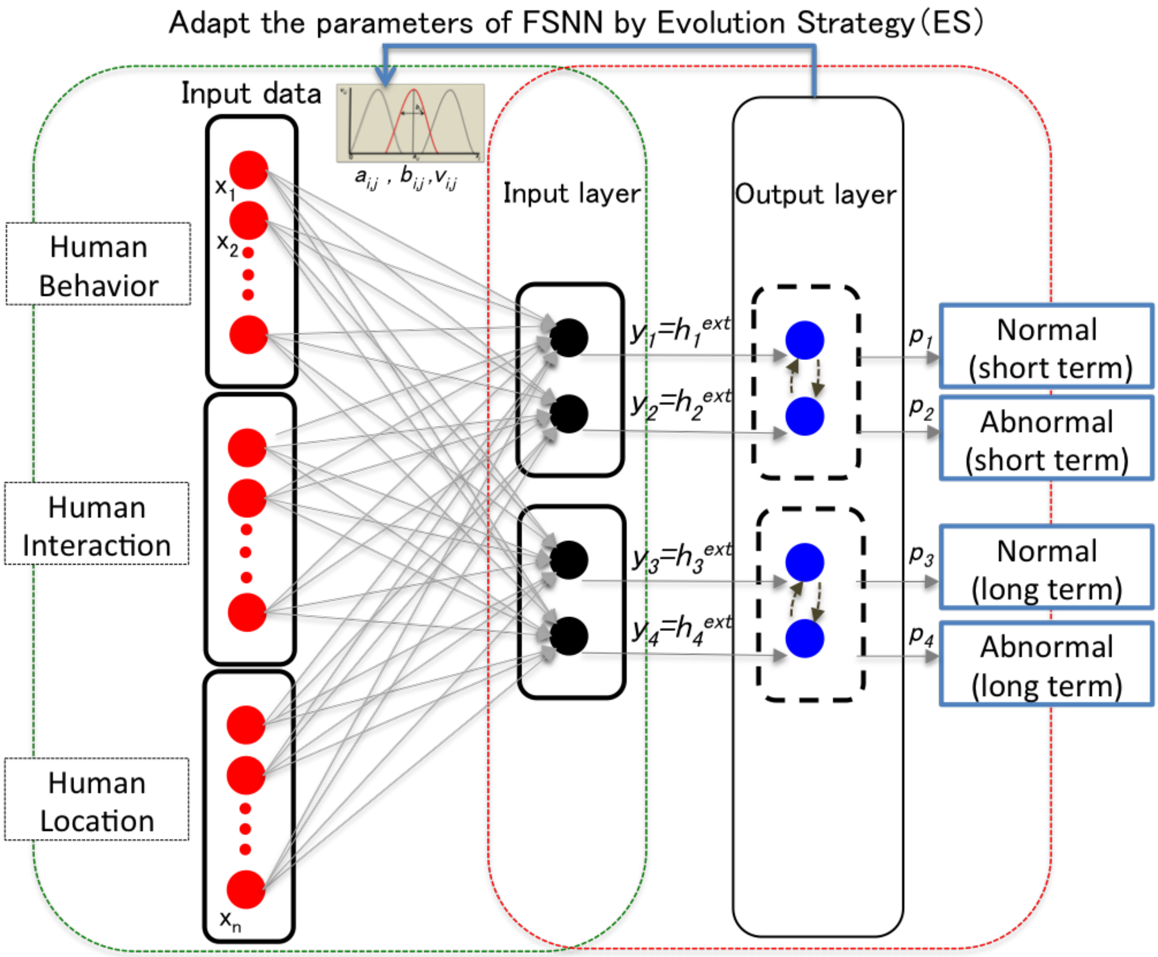



Figure 5.10: Structure of the FSNN

Table 5.2: Location

| No. | Location |
|---|---|
| 0 | outdoor |
| 1 | bedroom |
| 2 | living room |
| 3 | kitchen |
| 4 | toilet |
| 5 | entrance |

Table 5.3: Human-Robot interaction

| No. | Interaction |
|---|---|
| 0 | no interaction |
| 1 | OK |
| 2 | not good |

### 5.5.1 Fuzzy Spiking Neural Network for Abnormality Detection

My model predicts the abnormal activity of the elderly people by fuzzy spiking neurons. One of the vital properties of spiking neurons is the temporal coding feature. In addition to that, many different types of spiking neural networks (SNN) have been applied for remembering temporal and spatial context [5, 79, 37]. In order to reduce the computational cost, a simple spike response model is used. The SSN is composed of fuzzy inputs values: it is a fuzzy spiking neural network (FSNN) [119, 118, 14]. For this research, I will detect the abnormal elderly people activities.Fig. 5.9 illustrates the FSNN model. Evolution strategy (ES) is used to adapt the parameters of the fuzzy membership functions applied as inputs to the spiking neural network. Furthermore, Hebbian learning [42] is used to modify the weights between the spiking neurons. The structure of the model is depicted in Fig. 5.10. In this proposed FSNN model, the model has two different spiking neural network modules. One is applied for the short-term abnormal activities detection (the upper boxes) and the other one is applied for the long-term abnormal activities detection (the lower boxes).

The inputs of the model are presented in Tables 5.1, 5.2, and 5.3. The input fuzzy inference sensory information from the informationally structured space will be performed by:

$$\mu_{A_{i,j}}(x_j) = \exp\left(-\frac{(x_j - a_{i,j})^2}{b_{i,j}}\right) \tag{5.4}$$

$$y_i = \prod_{j=1}^{m} v_{i,j} \cdot \mu_{A_{i,j}}(x_j) \tag{5.5}$$

where $a_{i,j}$ is the middle value and $b_{i,j}$ is the width of the membership function. Next, $v_{i,j}$ and $A_{i,j}$ is the part of the $j$-th input to the estimation of the $i$-th human state. The $y_i$ is the product of fuzzy inference and it is also the input of the spiking neurons. The internal state $h_i(t)$ of the $i$-th spiking neuron (also known as membrane potential) at the discrete time $t$ is defined by:

$$h_i(t) = \tanh(h_i^{syn}(t) + h_i^{ext}(t) + h_i^{ref}(t)), \tag{5.6}$$

where, $h_i^{ext}(t)$ (computed by Eq. (5.7) is from the external environment input influences to the $i$-th neuron, $h_i^{syn}(t)$ (calculated in Eq. (5.8) incorporated the pulses from the other fully connected neurons outputs, $h_i^{ref}(t)$ (shown in Eq. (5.10)) is used for representing the refractoriness of the neuron. The hyperbolic tangent function is used to avoid the bursting of neuronal fires.

$$h_i^{ext}(t) = \prod_{j=1}^{m} v_{i,j} \cdot \exp\left(-\frac{(x_j - a_{i,j})^2}{b_{i,j}}\right) \tag{5.7}$$

In Fig. 5.10, $h_i^{syn}(t)$ is presented by dashed arrows.

$$h_i^{syn}(t) = \gamma^{syn} \cdot h_i(t-1) + \sum_{j=1, j\neq i}^{n} w_{j,i} \cdot h_j^{PSP}(t-1), \tag{5.8}$$

where $\gamma^{syn}$ is a temporal discount rate, $n$ is the number of spiking neurons, and $m$ in Eq. (5.7) is the total number of inputs.

When the internal state of the $i$-th neuron reaches a predefined threshold level, a pulse is outputted as follows:

$$p_i(t) = \begin{cases} 1 & \text{if } h_i(t) \geq \theta, \\ 0 & \text{otherwise,} \end{cases} \tag{5.9}$$

where $\theta$ is a threshold for firing. In case of firing, $R$ is subtracted from the $h_i^{ref}(t)$ value of neuron $i$ as follows:

$$h_i^{ref}(t) = \begin{cases} \gamma^{ref} \cdot h_i^{ref}(t-1) - R & \text{if } p_i(t-1) = 1, \\ \gamma^{ref} \cdot h_i^{ref}(t-1) & \text{otherwise,} \end{cases} \tag{5.10}$$

where $\gamma^{ref}$ is a discount rate of $h_i^{ref}$ and $R > 0$. The presynaptic spike output is transmitted to the connected neuron through the weight connection. The PSP is calculated as follows:

$$h_i^{PSP}(t) = \begin{cases} 1 & \text{if } p_i(t) = 1, \\ \gamma^{PSP} \cdot h_i^{PSP}(t-1) & \text{otherwise,} \end{cases} \tag{5.11}$$

where $\gamma^{PSP}$ is a discount rate of $h_i^{PSP}$.

### 5.5.2 Evolution Strategy for Optimizing The Parameters of FSNN

I apply $(\mu+\lambda)$-Evolution Strategy (ES) to optimize the fuzzy spiking neural network parameters in fuzzy membership functions. In $(\mu+\lambda)$-ES $\mu$ and $\lambda$ then state the number of parents and the number of offspring produced in an evolution generation correspondingly [105]. I apply $(\mu+1)$-ES for improving the local hill-climbing search as a continuous model of generations, which terminates and initializes one individual in one evolution generation. The $(\mu+1)$-ES can be considered as a steady-state genetic algorithm (SSGA) [114].

Table 5.4: Confusion matrix for a two-class problem

| | Positive prediction | Negative prediction |
|---|---|---|
| Positive class (abnormal data) | true positive (TP) | false negative (FN) |
| Negative class (normal data) | false positive (FP) | true negative (TN) |

A candidate solution contains the parameters of the fuzzy membership functions:

$$\boldsymbol{g}_k = \begin{bmatrix} g_{k,1} & g_{k,2} & g_{k,3} & \dots & g_{k,l} \end{bmatrix} \tag{5.12}$$

$$= \begin{bmatrix} a_{k,1,1} & b_{k,1,1} & v_{k,1,1} & \dots & v_{k,n,m} \end{bmatrix} \tag{5.13}$$

Where $n$ is the number of spiking neurons; $m$ is the total number of inputs; $l = n \cdot m$ is the chromosome length of the $k$-th candidate solution.

The amount of abnormal data is much smaller than the amount of normal data in general. It is imbalanced data. In this scenario, if I only use the accuracy information to evaluate the GA result, then it would not provide good information. The reason is that the comprehensiveness should be considered for the evaluation of the result. Therefore I apply F-measure metric for the fitness value evaluation. Table 5.4 shows the confusion matrix for a two-class problem. The fitness value of the $k$-th candidate solution is calculated as:

$$f_k = \frac{2P_kR_k}{(P_k + R_k)} \tag{5.14}$$

where $P_k$, $R_k$ are precision and recall of the $k$-th candidate solution.

Precision is calculated as:

$$P_k = \frac{TP}{(TP + FP)} \tag{5.15}$$

Recall is calculated as:

$$R_k = \frac{TP}{(TP + FN)} \tag{5.16}$$

where $TP$, $FP$ and $FN$ are the total number of true positive, false positive, and false negative cases in the $k$-th candidate solution, respectively.

The number of correct estimation rates is calculated as:

$$c_k = \sum_{i=1}^{n} c_{k,i} \tag{5.17}$$

where $c_{k,i}$ is the number of correct estimation rates of the $i$-th neuron. I compare each output of the FSNN in the time sequence with the corresponding desired output. If the FSNN's output is the same as the desired output, then I count this as a matching condition. The number of matchings for the $i$-th neuron is $c_{k,i}$. In $(\mu + 1)$-ES, only one existing solution is replaced with the candidate solution generated by crossover and mutation. I use elitist crossover and adaptive mutation. Elitist crossover randomly selects one individual, and generates one individual by combining genetic information between the selected individual and the best individual in order to obtain feasible solutions from the previous estimation result rapidly. The newly generated individual replaces the worst individual in the population after applying adaptive mutation on the newly generated individual. The inheritance probability of the genes corresponding to the $i$-th rule ($i$-th spiking neuron) of the best individual is calculated by:

$$p_i = \frac{1}{2} \cdot (1 + c_{best,i} - c_{k,i}) \tag{5.18}$$

where $c_{best,i}$ and $c_{k,i}$ are the number of correct estimation rates of the best individual and the randomly selected $k$-th individuals, respectively. With Eq. (5.18) I can bias the selection probability of the $i$-th genes from 0.5 to the direction of the better individual's $i$-th genes between the best individual and the $k$-th individual. Thus, the newly generated individual can inherit the $i$-th genes ($i$-th rule) from that individual which the better $i$-th gene has. After the crossover operation, an adaptive mutation is performed on the generated individual:

$$g_{k,h} \leftarrow g_{k,h} + \alpha_h \cdot (1 - t/T) \cdot N(0,1) \tag{5.19}$$

where $N(0,1)$ indicates a normal random value; $\alpha_h$ is a parameter of the mutation operator ($h$ stands for identifying the three subgroups in the individual related to $a$, $b$, and $v$); $t$ is the current generation; and $T$ is the maximum number of generations. The computational cost of ES is generation size + population size, $T + P$.

### 5.5.3 Hebbian Learning

As illustrated in Fig. 5.9, besides optimizing the parameters of the fuzzy membership functions by evolution strategy, the weights between the spiking neurons are modified by Hebbian learning. The weights can be adjusted dynamically in the simulation process. If the condition $0 < h_j^{PSP}(t-1) < h_i^{PSP}(t)$ is satisfied, the weight parameter, $w_{j,i}$ is trained based on the Hebbian learning rule [42]:

$$w_{j,i} \leftarrow \tanh\left(\gamma^{wgt} w_{j,i} + \xi^{wgt} h_j^{PSP}(t-1) h_i^{PSP}(t)\right), \tag{5.20}$$

where $\gamma^{wgt}$ is a discount rate of the weights and $\xi^{wgt}$ is a learning rate.

## 5.6 Experimental Results

### 5.6.1 Experiments for Daily Life Model Estimation

I use (1) overall daily temporal pattern, (2) seasonal daily temporal pattern, (3) monthly daily temporal pattern, and (4) daily temporal pattern by the day of week. I show a simulation example using artificial life log data obtained by the human life simulator. The size of human life log data is 90 days. The learning rate, $\alpha$ is 0.1 and the threshold, $\theta$ is 0.15. Figure 5.11 shows a simulation result of life logs for 90 days; (a) Life logs for 90 days and (b) Life logs by the day of week for 90 days. The simulation results show that the person almost takes a regular life by the day of week.

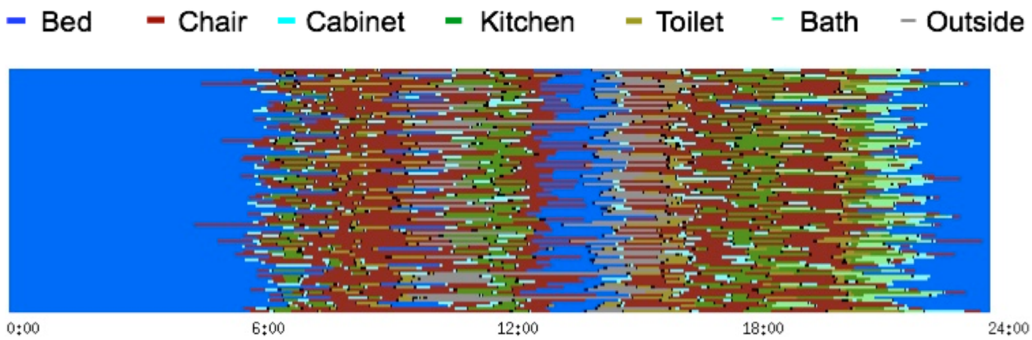
Fig. 5.12 shows temporal visualization of behaviors in ISS; (a) Temporal pattern of each behavior for the first day, (b) Temporal pattern of each behavior for 90 days and (c) Temporal pattern of each behavior on Sunday. These results show that the proposed method can add clusters of each behavior to the temporal life pattern step by step. The comparison result shows several differences, e.g., the wake-up time on Sunday is a little later than that of all days; the person takes short nap in the afternoon everyday; the person goes out on Sunday afternoon.

### 5.6.2 Experiments for Sensor Failure Detection

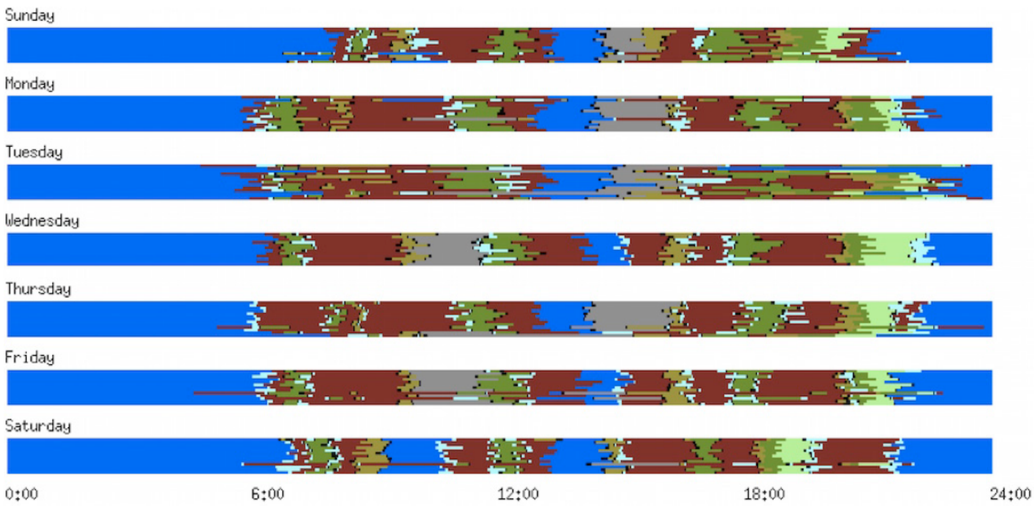
Figure 5.13 illustrates simulation result for sensor state and behavior state. The sensor can be on (green), off (white), fired (red), and error (blue). We can see the behavior's firing state, and the relation between behaviors and sensors. Here, after the human opens chest and changes clothes, SunSPOT2 installed on the chest fired, however SensorTag2 did not fire. Since the change clothes activity and SensorTag2 have connection, SensorTag2 is evaluated as having an error (Fig. 5.14).

### 5.6.3 Experiments for Sensor Sampling Interval Control

In this section, I realized sensor sampling interval control by simulation experiment. I show the sampling interval control changing when the person goes to bed (Fig. 5.15).

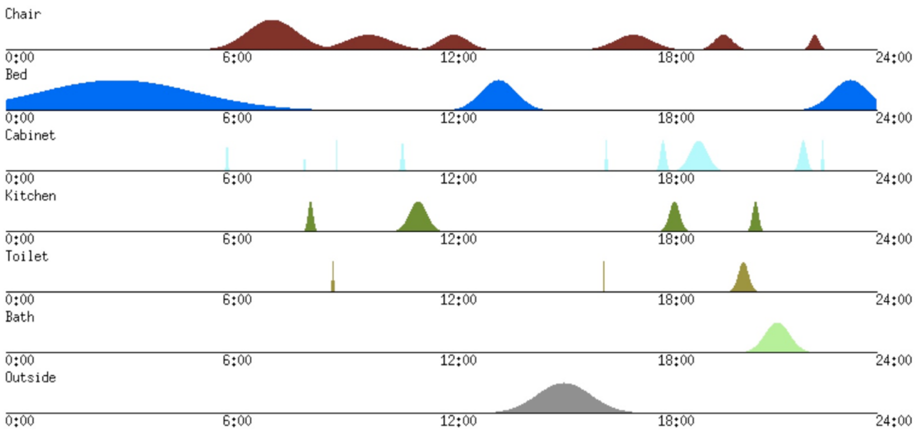



(a) Life logs for 90 days

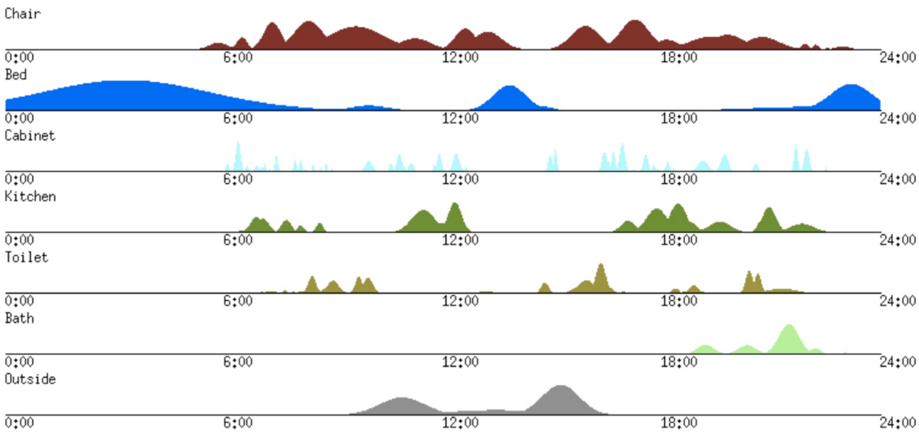



(b) Life logs by day of week for 90 days

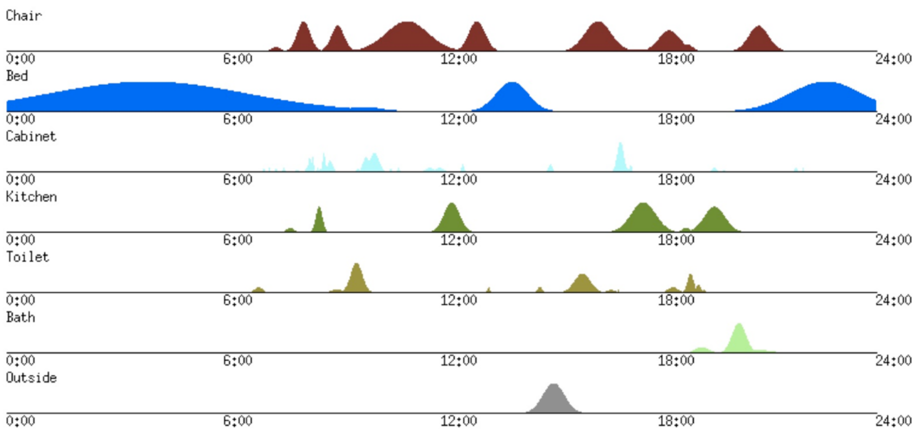
Figure 5.11: Visualization of life logs in ISS




(a) Temporal pattern of each behavior for the first day




(b) Temporal pattern of each behavior for 90 days (overall)




(c) Temporal pattern of each behavior on Sunday

Figure 5.12: Temporal visualization of behaviors in informationally structured space

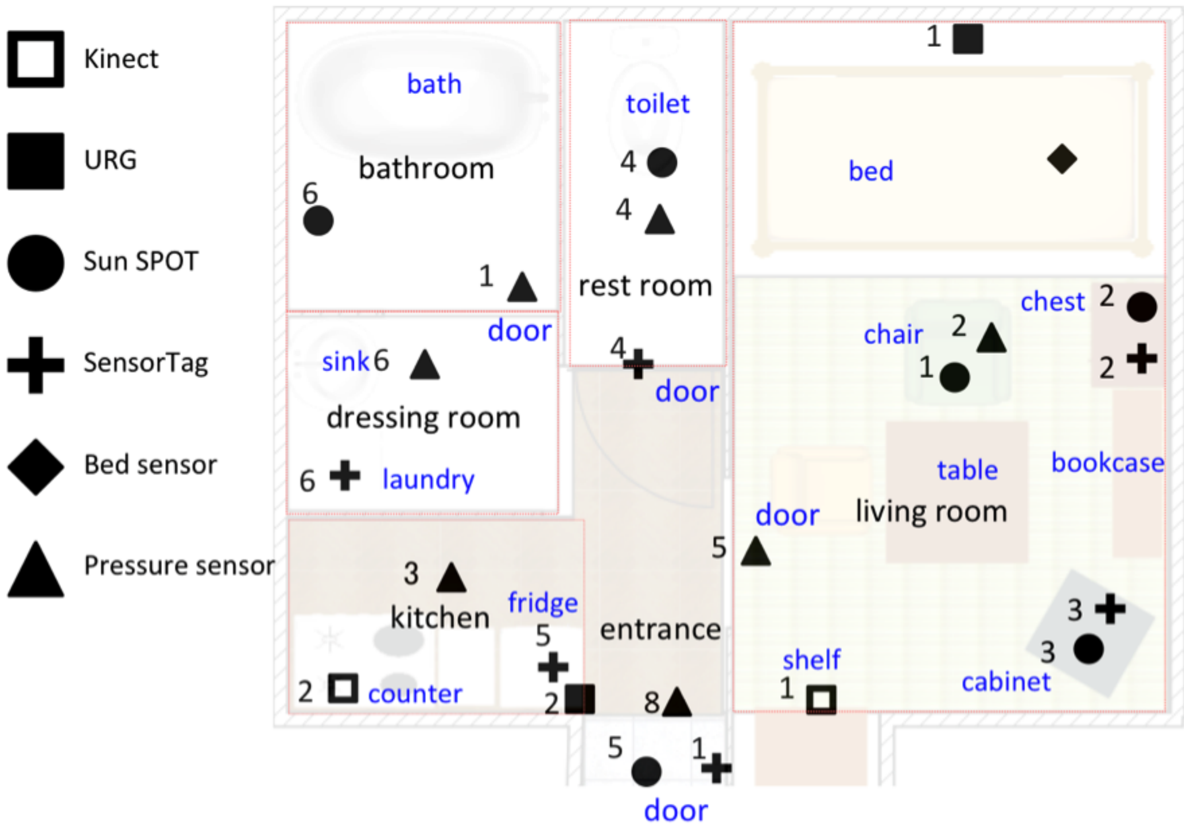



Figure 5.13: Sensor position in sensor failure detection experiment

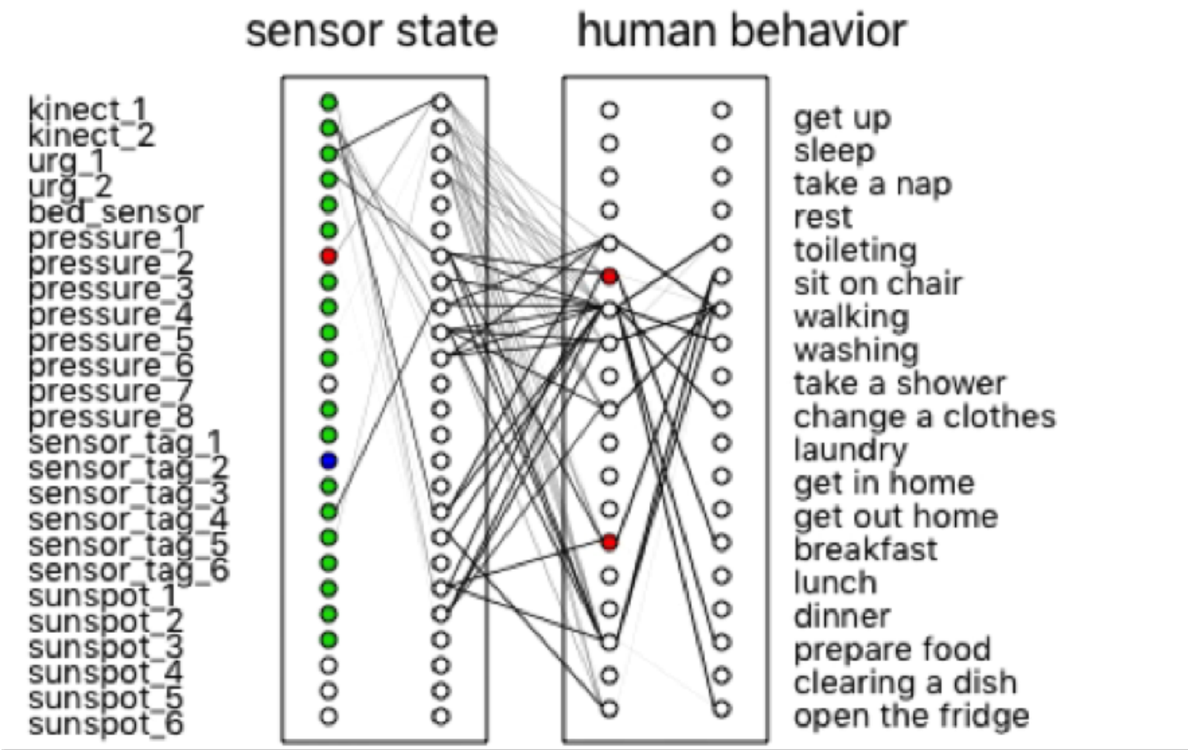



Figure 5.14: Experiment result for sensor failure detection

All of sensors sampling interval are 0.5 second before the person goes to bed. After the person goes to bed, only the sensor sampling interval of the Kinect sensors and SunSPOT sensor attached on bed kept the 0.5 second. The sampling interval of the other sensors became 1.5 second, because the other sensors do not require high precision when the person goes to bed. In this way, we can control sensor sampling interval to flexibly measure human activity considering energy consumption and computation cost. This time I use a rule base to control the sensors. As a future work, I will consider the relation between activity and activity prediction.

### 5.6.4 Experiments for Abnormal State Estimation

This section shows comparison results and analyzes the performance of the proposed method. The graph in Fig. 5.16 is the input data description; there are 18 inputs: 9 human behavior inputs, 6 human location inputs, and 3 human interaction inputs. In the proposed structure there are 2 SNN modules: the short- and the long-term modules. In the output layer there are 4 outputs: the short-term normal, short-term abnormal, long-term normal and the long-term abnormal output. 22 days training data set and 2 days test data set were used in the experiments. The parameters of

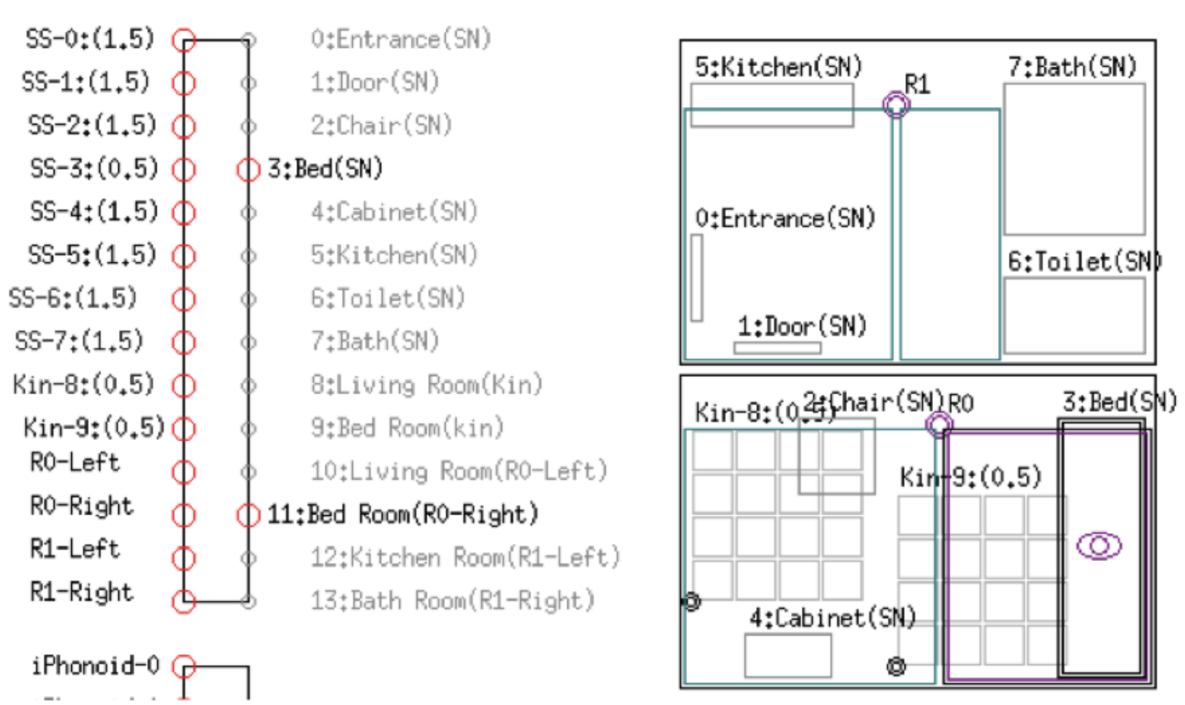



Figure 5.15: Experiment result for sensor sampling interval control

the neural network are presented in Table 5.5, where S stands for short-term and L stands for long-term.

Figure 5.16 shows a simulation window of the experiment. The upper window shows 3 different kinds of input data: the human behavior, the human location and the human interaction input data. The blue line is the input data from life log; the red line is the PSP output for the SNN's input layer. The lower window shows the human state estimation process by the SNN module. The blue line is the input data of each neuron node; the red line is the internal state; the green line is the training data; the purple line is the estimation result.

Table 5.5: Parameters of the neural network

| | $M$ | $N$ | $\gamma^{syn}$ | $\gamma^{ref}$ | $\gamma^{psp}$ | $R$ | $\theta$ | $\gamma^{wgt}$ | $\xi^{wgt}$ |
|---|---|---|---|---|---|---|---|---|---|
| S | 18 | 2 | 0.59 | 0.95 | 0.40 | 1 | 0.80 | 1.0 | 0.05 |
| L | 18 | 2 | 0.70 | 0.90 | 0.90 | 1 | 0.99 | 1.0 | 0.8 |

In the long-term data experiment, I use adapted parameters from the training experiments in order to estimate the test data. Table 5.6 shows the experimental result for training datasets and test datasets. In this case, I calculate the average based on 10 simulation experiments for each data. For training data I also calculated the standard deviation. The number of total training data is 31680, with 2059 abnormal

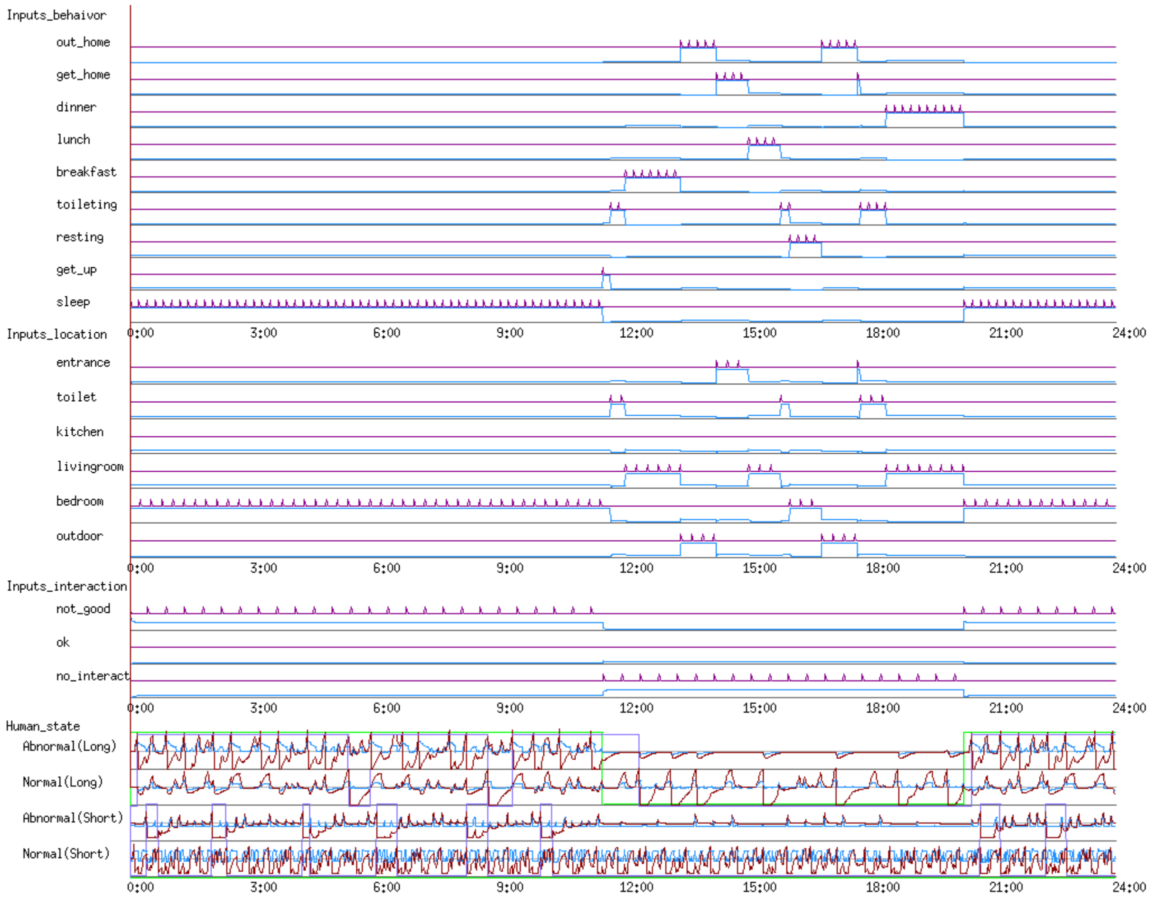



Figure 5.16: Simulation window of the experiment

and 29621 normal data. The number of total test data is 2880, with 108 abnormal and 2772 normal data. The test data experiment is as follows. Figure 5.17 illustrates the input of long-term test data in one-day human activity. There is one abnormal state when the human spends quite a long time in the toilet after lunch, and during lunch time the interaction is "not good".

Figure 5.18 shows the estimated result by using SNN. It can be understood, that the estimation result (purple line) is nearly at dose moreover, it does not match the teaching data (green line), and the F-measure, accuracy and fitting rate are not good as well. The reason is, that the feature of SNN's input data does not fit the output state. Figures 5.19 and 5.21 show the result by using GA to update the SNN parameters, where $T$ is the number of generations and $P$ is the population size. According to the many times experiment by applying GA, I knew that the experimental result does not change when $T = 20000$ and $P = 500$. Because of this, I present experiment result here when $T = 2000$ $P = 200$ (Figure 5.19) and $T = 20000$, $P = 500$ (Figure 5.21). I can see, that the estimation result (purple line) is almost matching the teaching data (green line), when the number of generations and the population size are increasing. The F-measure, accuracy and fitting rate became better than applying only SNN. With the use of GA, I'm able to optimize the parameter of membership function for SNN's input data, and it follows, that the feature of SNN's input data fits the output state. Figures 5.20 and 5.22 show the application of GA and Hebbian learning. F-measure, accuracy and fitting rate have not improved that much and the result is not as better as applying SNN and GA. I will explain the reason in short-term result.

Table 5.6: Long-term experimental result

| | F-Measure | Accuracy | Fitting rate (abnormal) | Fitting rate (normal) | F-Measure (Hebbian) | Accuracy (Hebbian) | Fitting rate of abnormal (Hebbian) | Fitting rate of normal (Hebbian) |
|---|---|---|---|---|---|---|---|---|
| Training data | | | | | | | | |
| SNN only | 0.251 | 0.694 (21987/31680) | 0.835 (1719/2059) | 0.685 (20285/29621) | 0.115 | 0.081 (2567/31680) | 0.970 (1997/2059) | 0.023 (677/29621) |
| After GA T=2000 P=200 | 0.679 ±0.148 | 0.965 ±0.080 (30583/31680) | 0.636 ±0.200 (1310/2059) | 0.987 ±0.088 (2923/29621) | 0.683 ±0.183 | 0.977 ±0.253 (30951/31680) | 0.697 ±0.183 (1434/2059) | 0.978 ±0.275 (28960/29621) |
| After GA T=8000 P=200 | 0.713 ±0.105 | 0.977 ±0.008 (30951/31680) | 0.692 ±0.215 (1424/2059) | 0.987 ±0.010 (29226/29621) | 0.773 ±0.079 | 0.977 ±0.010 (30951/31680) | 0.837 ±0.156 (1724/2059) | 0.981 ±0.007 (29063/29621) |
| After GA T=20000 P=500 | 0.792 ±0.112 | 0.977 ±0.007 (30951/31680) | 0.816 ±0.182 (1680/2059) | 0.984 ±0.006 (29133/29621) | 0.958 ±0.075 | 0.989 ±0.004 (30951/31680) | 0.923 ±0.058 (1839/2059) | 0.991 ±0.002 (29040/29621) |
| Test data | | | | | | | | |
| SNN only | 0 | 0.583 (1679/2880 | 0 (0/108) | 0.606 (1679/2772) | 0 | 0.583 (1679/2880 | 0 (0/108) | 0.606 (1679/2772) |
| After GA T=2000 P=200 | 0.512 | 0.993 (2589/2880 | 0.861 (93/108) | 0.996 (2760/2772) | 0.868 | 0.990 (2852/2880 | 0.852 (92/108) | 0.996 (2760/2772) |
| After GA T=8000 P=200 | 0.835 | 0.988 (2846/2880 | 0.796 (86/108) | 0.996 (2760/2772) | 0.874 | 0.991 (2853/2880 | 0.870 (94/108) | 0.995 (2759/2772) |
| After GA T=20000 P=500 | 0.869 | 0.990 (2852/2880 | 0.861 (93/108) | 0.995 (2759/2772) | 0.883 | 0.991 (2855/2880 | 0.870 (94/108) | 0.996 (2761/2772) |

Table 5.7: Short-term experimental result

| | F-Measure | Accuracy | Fitting rate of abnormal | Fitting rate of normal | F-Measure (Hebbian) | Accuracy (Hebbian) | Fitting rate of abnormal (Hebbian) | Fitting rate of normal (Hebbian) |
|---|---|---|---|---|---|---|---|---|
| Training data | | | | | | | | |
| SNN only | 0.167 | 0.984 (31161/31680) | 0.510 (52/102) | 0.985 (31109/31578) | 0.167 | 0.984 (31161/31680) | 0.510 (52/102) | 0.985 (31109/31578) |
| After GA T=2000 P=200 | 0.818 $\pm 0.206$ | 0.999 $\pm 0.004$ (31639/31680) | 0.870 $\pm 0.123$ (89/102) | 0.999 $\pm 0.004$ (31550/31578) | 0.827 $\pm 0.206$ | 0.999 $\pm 0.0004$ (31646/31680) | 0.844 $\pm 0.166$ (86/102) | 0.999 $\pm 0.004$ (31560/31578) |
| After GA T=8000 P=200 | 0.847 $\pm 0.053$ | 0.999 $\pm 0.001$ (31646/31680) | 0.888 $\pm 0.026$ (91/102) | 0.999 $\pm 0.001$ (31556/31578) | 0.869 $\pm 0.206$ | 0.999 $\pm 0.004$ (31653/31680) | 0.892 $\pm 0.123$ (91/102) | 0.999 $\pm 0.004$ (31562/31578) |
| After GA T=20000 P=500 | 0.855 $\pm 0.044$ | 0.999 $\pm 0.000$ (31649/31680) | 0.883 $\pm 0.017$ (90/102) | 0.999 $\pm 0.000$ (31559/31578) | 0.877 $\pm 0.011$ | 0.999 $\pm 0.000$ (31654/31680) | 0.912 $\pm 0.020$ (93/102) | 0.999 $\pm 0.000$ (31561/31578) |
| Test data | | | | | | | | |
| SNN only | 0.051 | 0.974 (2806/2880) | 0.143 (2/14) | 0.978 (2804/2866) | 0.051 | 0.974 (2806/2880) | 0.143 (2/14) | 0.978 (2804/2866) |
| After GA T=2000 P=200 | 0.373 | 0.987 (2843/2880) | 0.786 (11/14) | 0.988 (2832/2866) | 0.595 | 0.995 (2865/2880) | 0.786 (11/14) | 0.996 (2854/2866) |
| After GA T=8000 P=200 | 0.512 | 0.993 (2859/2880) | 0.786 (11/14) | 0.994 (2848/2866) | 0.629 | 0.995 (2867/2880) | 0.786 (11/14) | 0.997 (2856/2866) |
| After GA T=20000 P=500 | 0.647 | 0.996 (2868/2880) | 0.786 (11/14) | 0.997 (2857/2866) | 0.846 | 0.999 (2876/2880) | 0.786 (11/14) | 1.000 (2865/2866) |

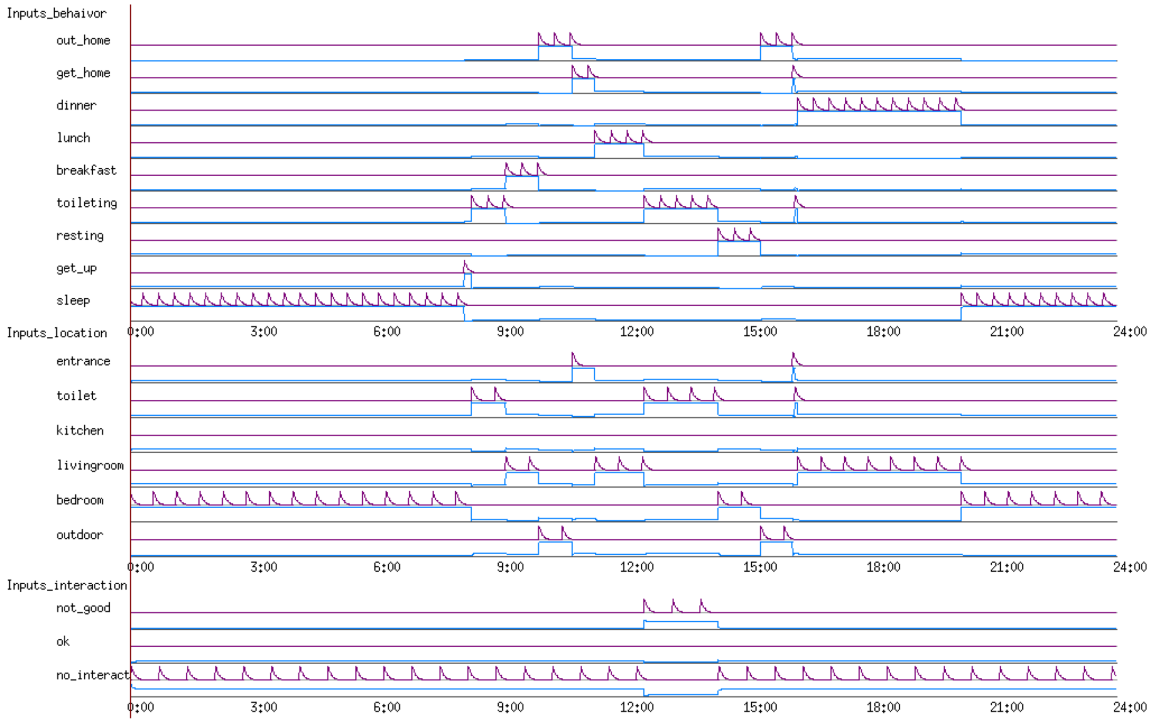



Figure 5.17: Input for long-term test data

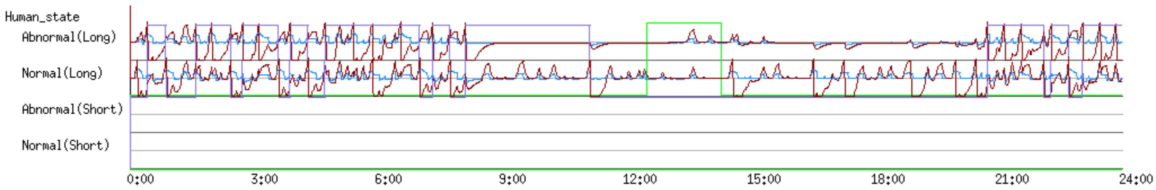



Figure 5.18: Experimental result by using SNN for long-term test data

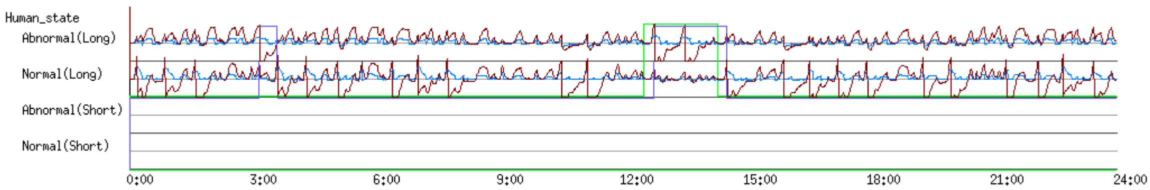



Figure 5.19: Experimental result after using GA for long-term test data (T=2000, P=200)

In the experiment for short-term test data, I use adapted parameters from training experiments in order to estimate the test data. Table 5.7 shows the experimental result for training and test datasets. In this case, I calculate the average based on 10 simulation experiments for each data, and for training data I also calculated the standard deviation. The number of total training data is 31680, with 102 abnormal

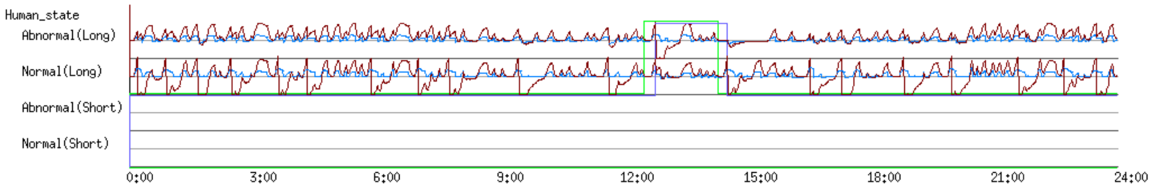



Figure 5.20: Experimental result after using Hebbian learning and GA for long-term test data (T=2000, P=200)

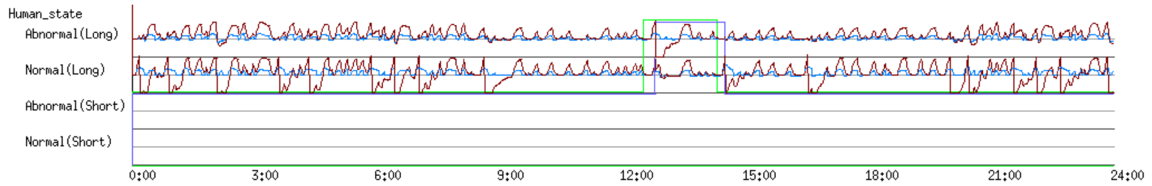



Figure 5.21: Experimental result after using GA for long-term test data (T=20000, P=500)

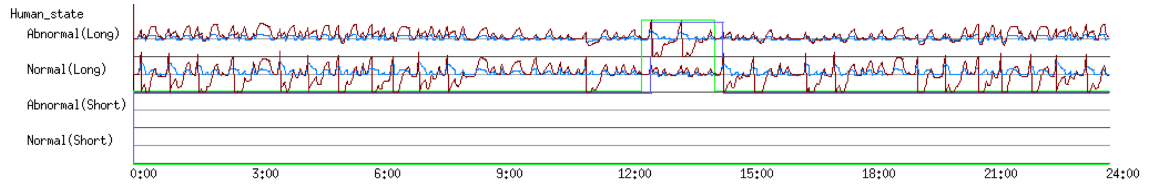



Figure 5.22: Experimental result after using Hebbian learning and GA for long-term test data (T=20000, P=500)

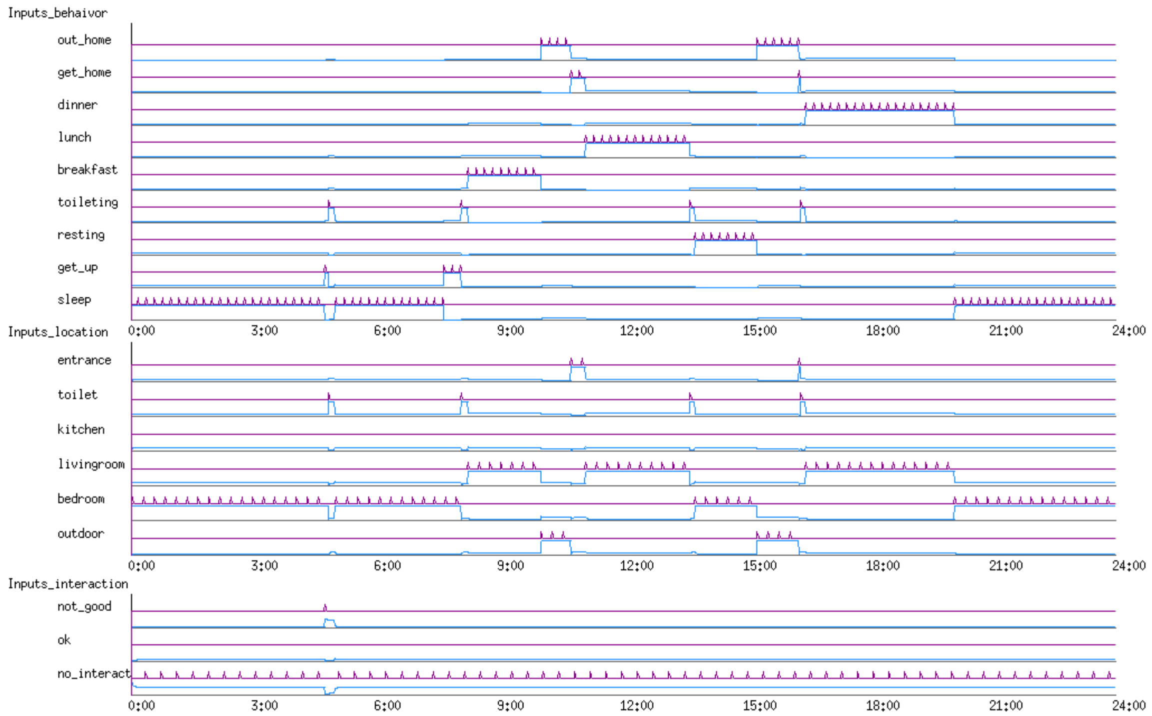



Figure 5.23: Input for short-term test data

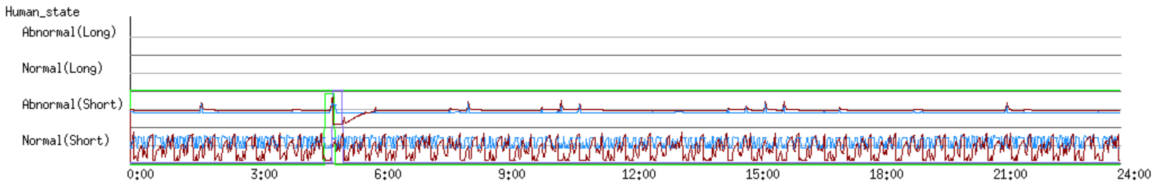



Figure 5.24: Experimental result by using SNN for short-term test data

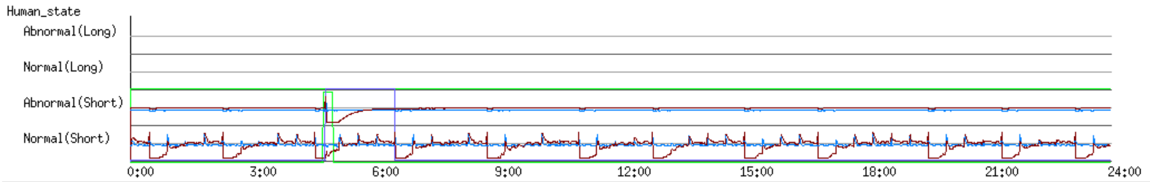



Figure 5.25: Experimental result after using GA for short-term test data (T=2000, P=200)

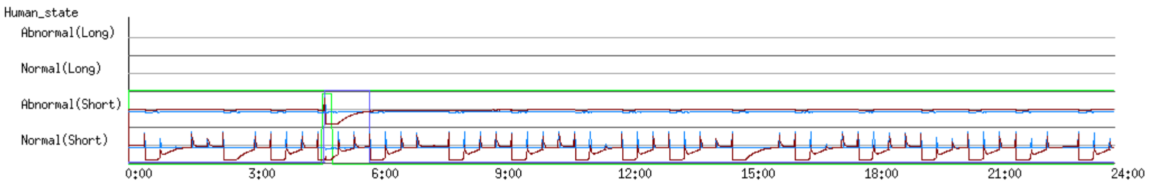



Figure 5.26: Experimental result after using Hebbian learning and GA for short-term test data (T=2000, P=200)

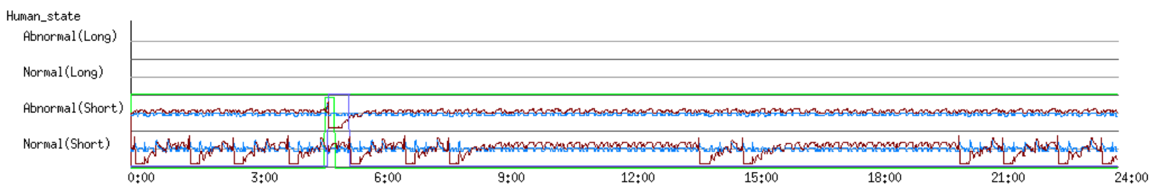



Figure 5.27: Experimental result after using GA for short-term test data (T=20000, P=500)

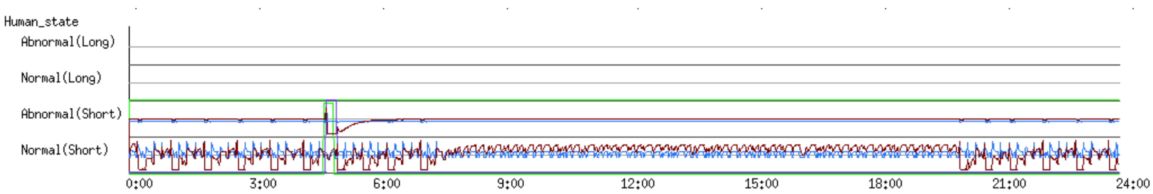



Figure 5.28: Experimental result after using Hebbian learning and GA for short-term test data (T=20000, P=500)

and 31578 normal data. The number of total test data is 2880, with 14 abnormal and 2866 normal data. Figure 5.23 illustrates the input of short-term test data in one-day human activity. The test data experiment is as follows. Figure 5.24 shows the estimated result by using SNN. Estimation result is similar to long-term result, which means that, the F-measure, accuracy and fitting rate are not good. The reason is, that feature of SNN's input data does not fit the output state. Figures 5.25 and 5.27 show the estimation result by using GA in order to update the SNN parameters. Estimation result is similar to the long-term result, I can see, that the estimation result (purple line) is nearly matching the teaching data (green line), when the number of generations and the size of population are increasing. The F-measure, accuracy and fitting rate became better than applying only SNN. To optimize the parameter of membership function for SNN's input data I use GA, and the feature of SNN's input data is going to fit the output state. Furthermore, the estimation result converges when $T = 20000$ and $P = 500$. Figures 5.26 and 5.28 show the application of Hebbian learning and GA. Estimation result has a different value with long-term result. In this case I can see, that Hebbian learning is effective. Estimation result (purple line) is nearly matching the teaching data (green line), and the F-measure, accuracy and fitting rate became better than applying GA and SNN, when the number of generations and the population size are increasing. It is because, I use Hebbian learning in order to influence the learning mechanism of neurons in a short period of time. The Hebbian learning is only effective in a short period of time, but as the time has passed PSP value was forgotten. This is why Hebbian learning is effective in short-term and not so effective in long-term.

## 5.7 Summary

This chapter discussed informationally structured space for monitoring layer in order to realize long-term human behavior measurement. I proposed daily life model estimation method using fuzzy modeling from long-term human behavior information. Moreover, I construct live models with different granularities by day, week and so on. I also propose a method that specifies the difference between the life models. I show several simulation results using artificial data obtained by the human life simulator. These simulation results show that the proposed method can extract human temporal life patterns.

I developed a simulator system to imitate the informationally structured space, and I use this simulator system to perform numerical experiments in order to show the effectiveness of the proposed method. The monitoring system is redundant, because several sensors often cover the same area with different resolutions. For example, the camera of a robot partner monitors the same area with the Kinect sensor. If a breakdown or error happens to a sensor node, I can detect different pattern in the simultaneous firing of sensor nodes. The informationally structured space can flexibly handle such situations.

I proposed an evolutionary computation approach to optimize spiking neural network for detecting abnormal activities in the elderly people's daily life. The initial experimental results showed that the proposed method is able to estimate abnormal activities based on human behavior, human location and human interaction data. As a future work, I intend to propose the method for anomaly detection from the life log data.

# Chapter 6

# Summary and outlook

The goal of the thesis is to realize the concept of informationally structured space for daily life monitoring. This thesis explained the concept of life hub to connect a person with information, and proposed the total architecture of an ISS. It also explained the life hub using smart devices and robot partners based on ISS. While discussing ISS, this thesis also explained the elements constructed the system such as sensor network, web system, and robot partner including gesture recognition and emotional model. Furthermore, ISS was also discussed from the human point of view for state estimation using smart devices. Human transport mode and interaction mode were estimated by evolution strategy and fuzzy spiking neural network. The proposed concept and methodology were also applied for life log measurement and visualization. The proposed approach was also able to conduct abnormal activities detection in the people's daily life.

Chapter 3 presented informationally structured space for distributed sensing system. I proposed human behavior measurement by applying spiking neural network, while realizing the structuring and updating of informationally structured space based on the relationship between location and human behavior. For indoor human activity measurement, I proposed a methodology for human behavior estimation by wireless sensor networks in informationally structured space. I provided several experimental results and discussed the effectiveness of the proposed method. In outdoor human activity measurement, I analyzed the performance of human transport mode and interaction mode estimation by fuzzy spiking neural network in informationally structured space using smart phone sensors. First, I applied a fuzzy spiking neural network to extract the human activity outdoors and smart phone states. Next, I discussed the method for updating the base value by conducting a preprocessing to generate input values to the spiking neurons. After that, I explained the learning method of fuzzy spiking neural network based on the time series of measured data using evolution strategy. Experimental results showed the effectiveness of the proposed method. However, there is a trade-off or balance between the update of the base values and the enhancement of the output mode estimation by weight connection between neurons. The update of the base values can inhibit incorrect fires by external noise, however it might also inhibit suitable fires according to small change of the measured data. The weight connection can enhance the suitable fires based on the prediction as a result of the temporal learning by time-series of the measured data, while that might cause incorrect fires by external noise. Therefore as a future work, I intend to improve the learning method to realize the suitable balance of enhancement and inhibition.

Chapter 4 explained informationally structured space for robot partner to enable human behavior active measurement. First, I discussed human behavior measurement by robot partners in feature extraction layer. Next, I proposed methodologies that complementary use behavior information measured by the distributed sensing system and behavior information estimated by the robot partner. The experimental result through time dependent conversation shows, that the robot partner can estimate human behavior that is difficult to measure by the distributed sensing system.

Chapter 5 discussed informationally structured space for monitoring layer in order to realize long-term human behavior measurement. First, I discussed daily life model estimation method using fuzzy modeling from long-term human behavior information. Then, I constructed live models with different granularities based on day, week and so on. I also proposed a method that specifies the differences between the life models. The experimental result shows the changing of the life models. In order to realize flexible adaptation to the changing of the environmental condition and system configuration, informationally structured space can flexibly change the way of signal processing and the network structure in the sensing layer, according to the access state of the devices, in the case of some events such as a device is broken or the battery is low. The experimental results showed the capability of informationally structured space in flexibly handling such situations. Additionally, I developed a simulator system to imitate the informationally structured space, and I used this simulator system to perform numerical experiments in order to show the effectiveness of the proposed method.

There are many possible developments of the system as future works. In order to understand human behavior, I intend to build mutual cognitive environment focused on human. I will also extend the capability of sensor and robot partner. Apart from them, additional information such as location and furniture information will be included. The integration and fusion between sensors can also be realized to extend sensor capability. With this method, the sensor measurement errors can be avoided, which will raise the robustness, accuracy and efficiency of the sensor.

# Bibliography


[1] E. Aarts and R. Wichert. *Ambient intelligence.* Springer, 2009.

[2] G. Acampora, D. J. Cook, P. Rashidi, and A. V. Vasilakos. A survey on ambient intelligence in healthcare. *Proceedings of the IEEE*, 101(12):2470–2494, 2013.

[3] P. Afsar, P. Cortez, and H. Santos. Automatic visual detection of human behavior: A review from 2000 to 2014. *Expert Systems with Applications*, 42(20):6935–6956, 2015.

[4] Continua Health Alliance. Continua health alliance. http://www.continua.jp/, 2015.

[5] J. A. Anderson and E.Rosenfeld. *Neurocomputing.* MIT Press Cambridge, 1988.

[6] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, et al. A view of cloud computing. *Communications of the ACM*, 53(4):50–58, 2010.

[7] A. T. Azar. *Fuzzy systems.* Sciyo. com, 2010.

[8] J. Bae, K. Kong, N. Byl, and M. Tomizuka. A mobile gait monitoring system for gait analysis. In *Rehabilitation Robotics, 2009. ICORR 2009. IEEE International Conference on*, pages 73–79. IEEE, 2009.

[9] T. Banerjee, J. M. Keller, and M. Skubie. Building a framework for recognition of activities of daily living from depth images using fuzzy logic. In *Fuzzy Systems (FUZZ-IEEE), 2014 IEEE International Conference on*, pages 540–547, 2014.

[10] Z. Z. Bien and H. Y. Lee. Effective learning system techniques for human-robot interaction in service environment. *Knowl.-Based Syst.*, 20(5):439–456, 2007.

[11] H. J. Böhme, T. Wilhelm, J. Key, C. Schauer, C. Schröter, H.M. Gross, and T. Hempel. An approach to multi-modal human-machine interaction for intelligent service robots. *Robotics and Autonomous Systems*, 44(1):83–96, 2003.

[12] J. Botzheim and N. Kubota. Growing neural gas for information extraction in gesture recognition and reproduction of robot partners. In *Proc. of the 23rd International Symposium on Micro-NanoMechatronics and Human Science*, pages 149–154, 2012.

[13] J. Botzheim, T. Obo, and N. Kubota. Human gesture recognition for robot partners by spiking neural networks and classification learning. In *Proc. of the 6th International Conference on Soft Computing and Intelligent Systems and the 13th International Symposium on Advanced Intelligent Systems*, pages 1954–1958, 2012.

[14] J. Botzheim, D. Tang, B. Yusuf, T. Obo, N. Kubota, and T. Yamaguchi. Extraction of daily life log measured by smart phone sensors using neural computing. *Procedia Computer Science*, 22:883–892, 2013.

[15] V. Callaghan, M. Colley, H. Hagras, F. Doctor J. Chin, and G. Clarke. Programming ispacesa tale of two paradigms. In *Intelligent Spaces: The Application of Pervasive ICT*, pages 389–421. Springer, 2005.

[16] J. L. C. Candás, V. Peláez, G. López, M. A. Fernández, E. Álvarez, and G. Díaz. An automatic data mining method to detect abnormal human behaviour using physical activity measurements. *Pervasive and Mobile Computing*, 15:228–241, 2014.

[17] J. L. C. Cands, V. Pelez, G. Lpez, M. . Fernndez, E. lvarez, and G. Daz. An automatic data mining method to detect abnormal human behaviour using physical activity measurements. *Pervasive and Mobile Computing*, 15:228–241, 2014.

[18] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao. Habitat monitoring: application driver for wireless communications technology. *Computer Communication Review*, 31(2):20–41, 2001.

[19] S. Chernbumroong, S. Cang, A. Atkins, and H. Yu. Elderly activities recognition and classification for applications in assisted living. *Expert Systems with Applications*, 40(5):1662–1674, 2013.

[20] D. J. Cook, J. C. Augusto, and V. R. Jakkula. Ambient intelligence: Technologies, applications, and opportunities. *Pervasive and Mobile Computing*, 5(4):277–298, 2009.

[21] D. J. Cook, J. C. Augusto, and V. R. Jakkula. Ambient intelligence: Technologies, applications, and opportunities. *Pervasive and Mobile Computing*, 5:277–298, 2009.

[22] International Business Machines Corporation(IBM). Navigating the ibm cloud. http://www.ibm.com/developerworks/websphere/techjournal, 2015.

[23] A. Costa, J. C. Castillo, P. Novais, A. F. Caballero, and R. Simoes. Sensor-driven agenda for intelligent home care of the elderly. *Expert Systems with Applications*, 39(15):12192–12204, 2012.

[24] N. Daosodsai and T. Maneewarn. Fuzzy based emotion generation mechanism for an emoticon robot. In *Proc. of the 13th International Conference on Control, Automation and Systems*, pages 1073–1078, Oct 2013.

[25] Y. Doi. Introduction-conceptual issues on quality of life (qol) and importance of qol research. *J. Natl. Inst. Public Health*, 53:3, 2004.

[26] A. P. Engelbrecht. *Computational intelligence: an introduction.* John Wiley & Sons, 2007.

[27] H. Fang, L. He, H. Si, P. Liu, and X. Xie. Human activity recognition based on feature selection in smart home using back-propagation algorithm. {*ISA*} *Transactions*, 53(5):1629–1638, 2014.

[28] A. Forkan, I. Khalil, and Z. Tari. Cocamaal: A cloud-oriented context-aware middleware in ambient assisted living. *Future Gener. Comput. Syst.*, 35:114–127, jun 2014.

[29] A. Forkan, I. Khalil, Z. Tari, S. Foufou, and A. Bouras. A context-aware approach for long-term behavioural change detection and abnormality prediction in ambient assisted living. *Pattern Recognition*, 48(3):628–641, 2015.

[30] I. Foster and C. Kesselman. What is the grid. *A three point checklist*, 20, 2003.

[31] B. Fritzke. Growing cell structures – a self-organizing network in $k$ dimensions. *Artificial Neural Networks*, 2(2):1051–1056, 1994.

[32] B. Fritzke. A growing neural gas network learns topologies. *Advances in Neural Information Processing Systems*, 7:625–632, 1995.

[33] B. Fritzke. Growing self-organizing networks – why? In *Proc. of the European Symposium on Artificial Neural Networks*, pages 61–72, 1996.

[34] Rudolph G. Convergence analysis of canonical genetic algorithms. *Neural Networks, IEEE Transactions on*, 5(1):96–101, 1994.

[35] W. Gerstner. *Spiking Neurons, In the book entitled Pulsed Neural Networks (Eds: W. Maass and C. M. Bishop)*. MIT press, 1999.

[36] W. Gerstner and W. M. Kistler. *Spiking neuron models*. Cambridge university press, 2002.

[37] W. Gerstner and W. M. Kistler. *Spiking Neuron Models*. Cambridge University Press, New York, USA, 2002.

[38] W. Gerstner, W. Maass, and C. M. Bishop. *Pulsed Neurol Networks*. MIT Press, 1999.

[39] J. J. Grefenstette. Optimization of control parameters for genetic algorithms. *Systems, Man and Cybernetics, IEEE Transactions on*, 16(1):122–128, 1986.

[40] M. M. Gupta and T. Yamakawa. Fuzzy computing. *Theory, Hardware, and Applications, Noth Holland, Amsterdam*, 1988.

[41] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning; Data mining, Inference, and Prediction*. Springer Verlag, 2001.

[42] D. O. Hebb. *The Organization of Behavior*. Wiley and Sons, New York, USA, 1949.

[43] K. Hirota and W. Pedrycz. Fuzzy computing for data mining. *Proceedings of the IEEE*, 87(9):1575–1600, 1999.

[44] J. H. Holland. *Adaption in Natural and Artificial Systems*. The MIT Press, Cambridge, Massachusetts, 1992.

[45] Houghton Mifflin Harcourt Publishing Company. The American Heritage Dictionary:Meaning of Community. https://www.ahdictionary.com/, 2015.

[46] S. Iengo, S. Rossi, M. Staffa, and A. Finzi. Continuous gesture recognition for flexible human-robot interaction. In *Proc. of the 2014 IEEE International Conference on Robotics and Automation*, pages 4863–4868, May 2014.

[47] F. Ingelrest, G. Barrenetxea, G. Schaefer, M. Vetterli, O. Couach, and M. Parlange. Sensorscope: Application-specific sensor network for environmental monitoring. *ACM Transactions on Sensor Networks*, 6(2):1–32, 2010.

[48] J. Intagliata. Improving the quality of community care for the chronically mentally disabled: The role of case management. *Schizophrenia Bulletin*, 8(4):655–674, 1982.

[49] T. Jeng. Toward a ubiquitous smart space design framework. *J. Inf. Sci. Eng.*, 25(3):675–686, 2009.

[50] W. Jitviriya and E. Hayashi. Design of emotion generation model and action selection for robots using a self organizing map. In *Proc. of the 11th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, pages 1–6, May 2014.

[51] B. Johanson, A. Fox, and T. Winograd. The interactive workspaces project: Experiences with ubiquitous computing rooms. *IEEE Pervasive Computing*, 1(2), 2002.

[52] D. A. Karnofsky. The clinical evaluation of chemotherapeutic agents in cancer. *Evaluation of chemotherapeutic agents*, 1949.

[53] K. Kelly. We are the web. *Wired magazine*, 13(8):113–23, 2005.

[54] K. Kemmotsu, T. Tomonaka, S. Shiotani, Y. Koketsu, and M. Iehara. Recognizing human behaviors with vision sensors in a network robot system. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 1274–1279. IEEE, 2006.

[55] J. Kemp, E. Gaura, R. Rednic, and J. Brusey. Long-term behavioural change detection through pervasive sensing. In *Proc. of the 14th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, pages 629–634, 2013.

[56] I. Khemapech, I. Duncan, and A. Mille. A survey of wireless sensor networks technology. In *Proc. of The 6th Annual PostGraduate Symposium on The Convergence of Telecommunications, Networking and Broadcasting*, 2005.

[57] H. G. Kim, J. Y Yang, and D. S. Kwon. Episodic memory system of affective agent with emotion for long-term human-robot interaction. In *Proc. of the 10th International Conference on Ubiquitous Robots and Ambient Intelligence*, pages 720–722, Oct 2013.

[58] J.H. Kim, Y.D. Kim, and K.H. Lee. The third generation of robotics: Ubiquitous robot. In *International Conference on Autonomous Robots and Agents*, pages 1–12, 2004.

[59] H. Kimura, N. Kubota, and J. Cao. Natural communication for robot partners based on computational intelligence for edutainment. *Mechatronics*, 2010:610–615, 2010.

[60] T. Kohonen. *Self-Organizing Maps*. Springer, Berlin, 2001.

[61] N. Kubota. Computational intelligence for structured learning of a partner robot based on imitation. *Information Science*, 171(4):403–429, 2005.

[62] N. Kubota. Cognitive development of partner robots based on interaction with people. In *Proc. of the Joint 4th International Conference on Soft Computing and Intelligent Systems and 9th International Symposium on Advanced Intelligent System*, pages 820–825, 2008.

[63] N. Kubota and T. Mori. Conversation system based on boltzmann selection and bayesian networks for a partner robot. In *Robot and Human Interactive Communication, 2009. RO-MAN 2009. The 18th IEEE International Symposium on*, pages 763–768, 2009.

[64] N. Kubota, T. Mori, and A. Yorita. Conversation system for robot partners based on informationally structured space. In *Robotic Intelligence In Informationally Structured Space (RiiSS), 2011 IEEE Workshop on*, pages 77–84. IEEE, 2011.

[65] N. Kubota and K. Nishida. Cooperative perceptual systems for partner robots based on sensor network. *International Journal of Computer Science and Network Security*, 6(11):19–28, 2002.

[66] N. Kubota, Y. Nojima, F. Kojima, and T. Fukuda. Multiple fuzzy state-value functions for human evaluation through interactive trajectory planning of a partner robot. *Sof Computing*, 10(10):891–901, 2006.

[67] N. Kubota, D. Tang, T. Obo, and S. Wakisaka. Localization of human based on fuzzy spiking neural network in informationally structured space. In *Proc. of the IEEE World Congress on Computational Intelligence*, pages 2209–2214, 2010.

[68] N. Kubota, Y. Toda, J. Botzheim, and B. Tudjarov. Multi-modal perception for human-friendly robot partners with smart phones based on computational intelligence. In *Proc. of the Fifth International Conference of South-West University, Faculty of Mathematics & Natural Sciences*, pages 17–25, 2013.

[69] N. Kubota and S. Wakisaka. Emotional model based on computational intelligence for partner robots. In T. Nishida, editor, *Modeling Machine Emotions for Realizing Intelligence*, pages 89–108. Springer-Verlag, Berlin Heidelberg, 2010.

[70] N. Kubota and A. Yorita. Topological environment reconstruction in informationally structured space for pocket robot partners. In *Proc. of the 2009 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, 2009.

[71] K. V. Laerhoven, D. Kilian, and B. Schiele. Using rhythm awareness in long-term activity recognition. In *Proc. of the 12th IEEE International Symposium on Wearable Computers (ISWC 2008)*, pages 63–66, 2008.

[72] J. H. Lee, H. Lee, M. J. Kim, X. Wang, and P. E.D. Love. Context-aware inference in ubiquitous residential environments. *Computers in Industry*, 65(1):148–157, 2014.

[73] G. Leinand, A. K. Bourke, and N. John. Evaluation of accelerometer based multi-sensor versus single-sensor activity recognition systems. *Medical Engineering a Physics*, 36(6):779–785, 2014.

[74] M. K. Lim, S. Tang, and C. S. Chan. isurveillance: Intelligent framework for multiple events detection in surveillance videos. *Expert Systems with Applications*, 41(10):4704–4715, 2014.

[75] J. Liu, P. Pasupat, S. Cyphers, and J. Glass. Asgard: A portable architecture for multilingual dialogue systems. In *ICASSP'13*, pages 8386–8390, 2013.

[76] Z. Liu, F. Hu, D. Luo, and X. Wu. Visual gesture recognition for human robot interaction using dynamic movement primitives. In *Proc. of the 2014 IEEE*

*International Conference on Systems, Man and Cybernetics*, pages 2094–2100, Oct 2014.

[77] J. Lopes, M. Eskenazi, and I. Trancoso. From rule-based to data-driven lexical entrainment models in spoken dialog systems. *Computer Speech and Language*, 31(1):87–112, 2015.

[78] K. Lyytinen and Y. Yoo. Ubiquitous computing. *Communications of the ACM*, 45(12):63–96, 2002.

[79] W. Maass and C. M. Bishop. *Pulsed Neural Networks.* MIT Press, Cambridge, Massachusetts, USA, 1999.

[80] S. Mahadev. Pervasive computing: Vision and challenges. *Personal Communications, IEEE*, 8(4):10–17, 2001.

[81] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson. Wireless sensor networks for habitat monitoring. In *Proc. of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pages 88–97, 2002.

[82] Mark Peters. Smartphone price drop continues. http://www.letsgodigital.org/en/23646/smartphone-price/, 2009.

[83] T. M. Martinetz and K. J. Schulten. A 'neural gas' network learns topologies. *Artificial Neural Networks*, 1:397–402, 1991.

[84] A. F. Montes, J.A. Ortega, J.I. S. Venzala, and L. G. Abril. Software reference architecture for smart environments: Perception. *Computer Standards and Interfaces*, 36(6):928–940, 2014.

[85] H. Mori. A consideration toward a human-centered ubiquitous society based on a three layered model. *Journal of the Society of Instrument and Control Engineers*, 47(2):77–81, 2008.

[86] T. Mori, R. Urushibata, M. Shimosaka, H. Noguchi, and T. Sato. Anomaly detection algorithm based on life pattern extraction from accumulated pyroelectric sensor data. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 2545–2552. IEEE, 2008.

[87] K. Morioka and H. Hashimoto. Appearance based object identification for distributed vision sensors in intelligent space. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 1, pages 199–204. IEEE, 2004.

[88] C. K. Narayanan and J. C. Diane. Activity recognition on streaming sensor data. *Pervasive and Mobile Computing*, 10, Part B:138–154, 2014.

[89] R. Nesselrath. Towards a cognitive load aware multimodal dialogue framework for the automotive domain. In *Proc. of the 9th International Conference on Intelligent Environments*, pages 266–269, July 2013.

[90] Y.H. Noh, G. H. Hwang, and D. U. Jeong. Implementation of real-time abnormal ecg detection algorithm for wearable healthcare. In *Computer Sciences and Convergence Information Technology (ICCIT), 2011 6th International Conference on*, pages 111–114. IEEE, 2011.

[91] World Health Organization et al. World health organization constitution. *Basic documents*, 1, 1948.

[92] A. Pantelopoulos and N. G. Bourbakis. A survey on wearable sensor-based systems for health monitoring and prognosis. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 40(1):1–12, 2010.

[93] D. Preuveneers, J.V. den Bergh, D. Wagelaar, A. Georges, P. Rigole, T. Clerckx, Y. Berbers, K. Coninx, V. Jonckers, and K.D. Bosschere. Towards an extensible context ontology for ambient intelligence. In *Lecture Notes In Computer Science*, volume 3295, pages 148–159. Springer, Nov 8-11 2004.

[94] B. Qureshi and A. Kouba. Five traits of performance enhancement using cloud robotics: A survey. *Procedia Computer Science*, 37:220–227, 2014.

[95] M. A. Rappa. The utility business model and the future of computing services. *IBM Systems Journal*, 43(1):32–42, 2004.

[96] J. Rasmussen, A.M.Pejtersen, and L.P.Goodstein. *Cognitive Systems Engineering*. Wiley Interscience Publication, Canada, 1994.

[97] P. Remagnino, H. Hagras, N. Monekosso, and S. Velastin. Ambient intelligence: A gentle introduction. *Ambient intelligence: a novel paradigm*, 2005.

[98] K. Römer, T. Schoch, F. Mattern, and T. Dübendorfer. Smart identification frameworks for ubiquitous computing applications. *Wireless Networks*, 10(6):689–700, 2004.

[99] W. Rueangsirarak, A.S. Atkins, B. Sharp, N. Chakpitak, and K. Meksamoot. Fall-risk screening system framework for physiotherapy care of elderly. *Expert Systems with Applications*, 39(10):8859–8864, 2012.

[100] L. Rutkowski. *Computational intelligence: methods and techniques*. Springer Science & Business Media, 2008.

[101] M. A. Sager, N. C. Dunham, A. Schwantes, L. Mecum, K. Halverson, and D. Harlowe. Measurement of activities of daily living in hospitalized elderly: A comparison of self-report and performance-based methods. *Journal of the American Geriatrics Society*, 40(5):457–462, 1992.

[102] D. Saha and A. Mukherjee. Pervasive computing: a paradigm for the 21st century. *Computer*, 36(3):25–31, 2003.

[103] K. Sakamura. The tron project. *Microprocessors and Microsystems*, 13(8):493–502, 1989.

[104] I. Satoh. Location-based services in ubiquitous computing environments. *International Journal on Digital Libraries*, 6(3):280–291, 2006.

[105] H.-P. Schwefel. *Numerical optimization of computer models*. John Wiley & Sons, Inc., 1981.

[106] S. O. Shahdi and S. A. R. Abu Bakar. Face recognition: Robust approach under varying and low resolution head poses. *International Journal of Imaging and Robotics*, 7:70–87, 2012.

[107] Y. Shimizu, S. Yoshida, J. Shimazaki, and N. Kubota. An interactive support system for activating shopping streets using robot partners in informationally structured space. In *Advanced Robotics and its Social Impacts (ARSO), 2013 IEEE Workshop on*, pages 70–75, 2013.

[108] J. H. Shin, B. Lee, and K. S. Park. Detection of abnormal living patterns for elderly living alone using support vector data description. *Information Technology in Biomedicine, IEEE Transactions on*, 15(3):438–448, 2011.

[109] D. Sperber and D. Wilson. *Relevance: Communication and cognition*. Blackwell Publishing Ltd., 1995.

[110] D. Sperber and D. Wilson. *Relevance: Communication and Cognition*. Blackwell Publishing, 1995.

[111] B. Spilker. *Quality of life and pharmacoeconomics in clinical trials*. Lippincott-Raven, 1996.

[112] B. Spilker and D. A. Revicki. Taxonomy of quality of life. *Quality of life and pharmacoeconomics in clinical trials*, 2:25–36, 1996.

[113] C. Sugimoto and R. Kohno. Development of wireless sensing system monitoring physiological information for healthcare in daily life. In *Sensing Technology (ICST), 2011 Fifth International Conference on*, pages 488–493. IEEE, 2011.

[114] G. Syswerda. A study of reproduction in generational and steady state genetic algorithms. In G. J. E. Rawlings, editor, *Foundations of genetic algorithms*, pages 94–101. Morgan Kaufmann, Indiana University, 1991.

[115] G. Syswerda. A study of reproduction in generational and steady state genetic algorithms. *Foundations of genetic algorithms*, 2:94–101, 1991.

[116] Y. Takemura and K.Ishii. Auto color calibration algorithm using neural networks and its application to robocup robot vision. *International Journal of Artificial Intelligence*, A11:368–383, 2011.

[117] W. R. Tan, C. S. Chan, Y. Pratheepan, and J. Condell. A fusion approach for efficient human skin detection. *IEEE Trans. Industrial Informatics*, 8(1):138–147, 2012.

[118] D. Tang, J. Botzheim, N. Kubota, and T. Yamaguchi. Estimation of human transport modes by fuzzy spiking neural network and evolution strategy in informationally structured space. In *Proc. of the IEEE International Workshop on Genetic and Evolutionary Fuzzy Systems*, pages 36–43, 2013.

[119] D. Tang and N. Kubota. Human localization by fuzzy spiking neural network based on informationally structured space. In *Neural Information Processing. Theory and Algorithms*, pages 25–32. Springer, 2010.

[120] D. Tang and N. Kubota. Information support system based on sensor networks. In *World Automation Congress (WAC), 2010*, pages 1–7. IEEE, 2010.

[121] D. Tang, B. Yusuf, J. Botzheim, N. Kubota, and C. S. Chan. A novel multimodal communication framework using robot partner for aging population. *Expert Systems with Applications*, 42(9):4540–4555, 2015.

[122] B. Tanvi, M. K. James, P. Mihail, and S. Marjorie. Recognizing complex instrumental activities of daily living using scene information and fuzzy logic. *Computer Vision and Image Understanding*, 140:68–82, 2015.

[123] B. Tay, Y. Jung, and T. Park. When stereotypes meet robots: The double-edge sword of robot gender and personality in human-robot interaction. *Computers in Human Behavior*, 38(0):75–84, 2014.

[124] M. Weiser. The computer for the 21st century. *Scientific american*, 265(3):94–104, 1991.

[125] M. Weiser. Some computer science issues in ubiquitous computing. *Communications of the ACM*, 36(7):75–84, 1993.

[126] M. Weiser and J. S. Brown. Designing calm technology. *PowerGrid Journal*, 1(1):75–85, 1996.

[127] J. M. Wiener, R. J. Hanley, R. Clark, and J. F. Van Nostrand. Measuring the activities of daily living: Comparisons across national surveys. *Journal of Gerontology*, 45(6):S229–S237, 1990.

[128] X. Yang, Y. Junsong, and T. Daniel. Human-virtual human interaction by upper body gesture understanding. In *Proc. of the 19th ACM Symposium on Virtual Reality Software and Technology*, pages 133–142, 2013.

[129] A. Yorita, J. Botzheim, and N. Kubota. Emotional models for multi-modal communication of robot partners. In *Proc. of the 2013 IEEE International Symposium on Industrial Electronics*, 2013.

[130] A. Yorita and N. Kubota. Cognitive development in partner robots for information support to elderly people. *Autonomous Mental Development, IEEE Transactions on*, 3(1):64–73, 2011.

[131] A. Yorita and N. Kubota. Cognitive development in partner robots for information support to elderly people. *Autonomous Mental Development, IEEE Transactions on*, 3(1):64–73, 2011.

[132] M. Yu, Y. Yu, A. Rhumal, S. M. R. Naqvi, L. Wang, J. Chambers, et al. An online one class support vector machine-based person-specific fall detection system for monitoring an elderly individual in a room environment. *Biomedical and Health Informatics, IEEE Journal of*, 17(6):1002–1014, 2013.

[133] L. A. Zadeh. Fuzzy sets. *Information and control*, 8(3):338–353, 1965.

[134] L. Zhang, M. Jiang, D. Farid, and M. A. Hossain. Intelligent facial emotion recognition and semantic-based topic detection for a humanoid robot. *Expert Systems with Applications*, 40(13):5160–5168, 2013.