Doctor of Philosophy

Multi-view Evolutionary Robot Vision for Exercise with Entertainment

2021, May

Department of Intelligent Mechanical Systems
Graduate School of System Design
Tokyo Metropolitan University

Wei Quan

Doctor of Philosophy

Multi-view Evolutionary Robot Vision for Exercise with Entertainment

2021, May

Department of Intelligent Mechanical Systems
Graduate School of System Design
Tokyo Metropolitan University

15929508

Wei Quan

Supervisor: Professor Naoyuki Kubota

Abstract

The aging society has become a serious issue in the world. In order to extend healthy life expectancy of elderly people, physical exercise is one of the most important solutions to prevent the decline of physical functions. In fact, elderly people also try to build a small size of community to conduct physical exercise in a friendly way by themselves. However, the contents of typical physical exercises are too monotonous for elderly people to maintain their motivation for practicing frequently. Furthermore, elderly people cannot continue physical exercises because they cannot understand the effect of physical exercises. Exertainment, which means the combination of exercise and entertainment, represents for a form of exercise that includes aspects of entertainment, especially exergaming by game machines. Although full-body exercises can prevent muscle disorders, most of exercising games measure only user's hand motions by accelerometers. Therefore, it is very important to measure human full-body postures significant for physical exercise. Furthermore, most of elderly people are not interested in game machines. On the other hand, since the number of elderly people using the reasonable price of smart phones is increasing in recent years, the introduction of smart phones to elderly people can be an alternative efficient solution to realize exertainment in a local community.

In this thesis, I focus on human posture estimation in the exertainment for elderly people using smart devices. Various types of human posture estimation methods have been proposed so far, but most of them estimate human joint positions, not joint angles. Furthermore, the computational cost of most previous methods is very expensive for inexpensive smart phones. Evolutionary robot vision is one of efficient human posture estimation methods with effective computational cost, but there are still several ill-posed problems such as occlusions and singular postures which are unmeasurable. In order to improve the estimation performance of human postures, I propose multi-view evolutionary robot vision by using multiple smart devices. First, I propose a method for estimating three-dimensional rotational joint angel of human postures from two-dimensional human motion measurement results. Next, I propose an evolutionary strategy for estimating the internal parameters by obtaining correct corresponding points in multi-view in order to improve the estimation performance of hu-

man postures. Finally, I develop an exertainment system based on human posture estimation, and show the effectiveness of the proposed method through various types of experiments on exertainment.

The thesis is organized as 6 chapters.

Chapter 1 introduces the social and theoretic background the current issues. Next, the contribution and structure of this dissertation are explained.

Chapter 2 explains the concept and current state of exertainment and the theory and methodology of computer vision, and clarifies the goal of this thesis and the importance of human motion analysis in exertainment.

In Chapter 3, I explain several estimation methods of joint angles of human posture proposed in this study, and discuss the estimation performance of evolutionary robot vision between RGB-D camera and monocular camera. First, I propose a method for estimating human postures by growing neural gas and evolutionary algorithm from the point cloud measured by a RGB-D camera. Next, I propose another method for estimating human postures by particle swarm optimization to reduce the computational cost. Furthermore, I propose a method for estimating human postures by evolutionary algorithms with a monocular camera. I compare the performance of the proposed method with conventional methods. Experimental results show that the performance obtained by the monocular camera is almost the same as that of other methods, but the computational cost of the proposed human posture estimation by monocular camera is the lowest among them.

In chapter 4, I propose multi-view evolutionary robot vision for the human posture estimation. First, I propose a method of evolutionary strategy sample consensus (ESSAC) for selecting correct pairs of corresponding points, and estimating internal parameters of cameras in two or more smart devices set with different views. As a result, it is possible to estimate internal parameters of cameras embedded in two or more smart devices set with different views. Next, I propose a method for estimating human postures from the measurement result of multi-view human motions. Experimental results show that the proposed method can reduce computational cost, while achieving similar or higher accuracy of estimating human postures in ill-posed conditions.

In chapter 5, I develop an exertainment system based on human posture estimation in order that two or more people enjoy physical exercise together in a local community. I implement several physical exercises developed for elderly people on smart devices. Next, I develop an exertainment system using postures including in the above physical exercises that two or more people play together. Furthermore, I develop an exertainment system using robot balls which is simulated as Boccia. Preliminary experimental results show that the proposed system can evaluate the rhythm-motion synchronization by two people. Finally, I

show the effectiveness of the proposed method by the multi-view evolutionary robot vision through various types of experiments on exertainment.

Chapter 6 concludes the thesis, and discusses future works towards the social implementation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

With the development of technology, human enjoy much longer lives than ever before. Nevertheless, this situation leads to a serious problem of aging population. According to the survey from World Health Organization, the population that elder than 65 would rise to 28% at the year of 2050, which is shown in Figure 1.1.

Aging population leads to the problems that is cased by elderly age such as heart disease is also rising at the same time, especially for elders who are living alone. Besides this, accidents are another threat to the elders' health. For instance, falls for elders who living alone are a substantial problem in individuals older than 65 years, occurring in 32% of those aged 65 to 74 years, in 35% of those aged 75 to 84 years, and in 51% of those older than 85 years each year in US[1].

In order to solve these problems, it is necessary to distribute health-givers (doctors and therapists, etc.) to elders for taking care their daily life. Study has shown that cognitive capability changes to less with the normal aging of human, and this situation would cause much more accidents cognition-associated diseases [2].

Nevertheless, there is a strict situation about the shortage of health-givers. Elderly people who living alone should under the nursing of health-givers, for not only handling emergent issues but also rehabilitation. Nevertheless, consider the cost and human resource, it is difficult to send health-givers to every one, therefore alternative solutions are under consideration. Based on this situation, robot turned out to be one of the most ideal solutions for solving this problem.

Considering this situation, robot assistance has been a more and more significant way for solving this issue, and has also been applied with various kinds of performances. For

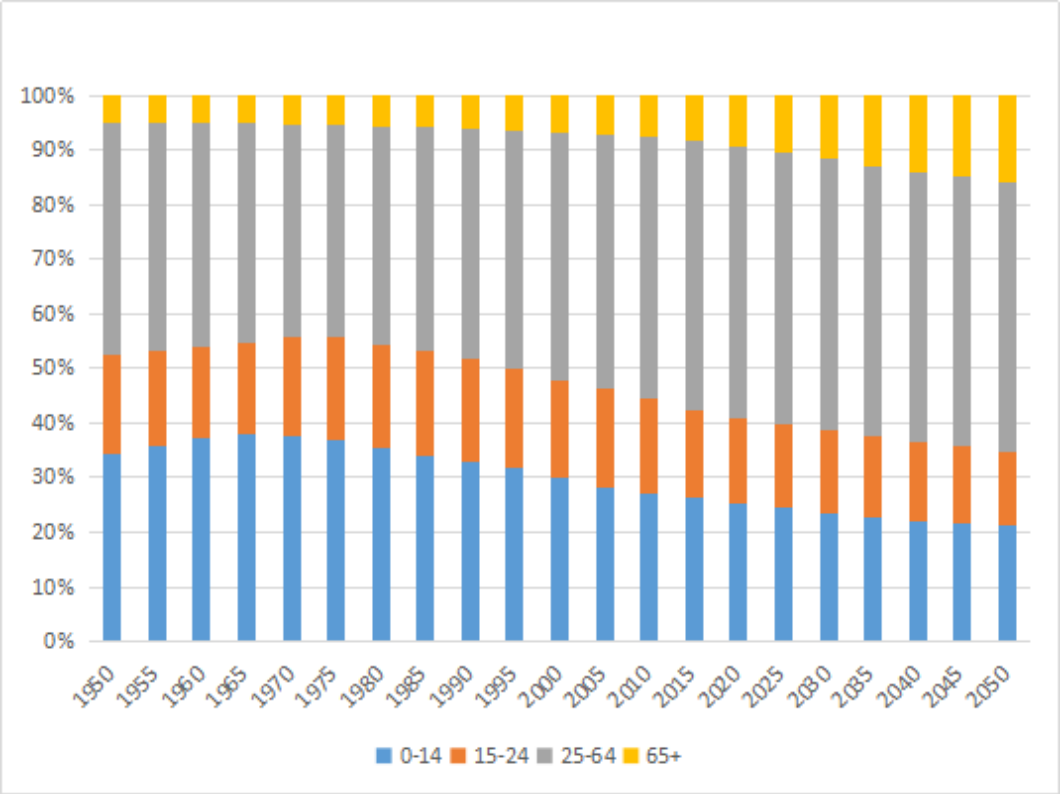


Figure 1.1: *Tendency of world population for elderly people.*

instance, [3, 4] introduced robot partners to help elderly people in Japan, and [5] also proposed a Socially Assistive Robot that engage, coach, assess and motivate the older adults in physical exercises in UK, and also capable of detecting anomaly activities of daily living by the assistive robot[6]. In this robot partners play the role of not only as the therapists or assistants, but also the communicator with the elder. Moreover, the robot can encourage the elderly people to engage in light to moderate physical activity.

Robotics is an interdisciplinary branch of engineering and science that includes mechanical engineering, electronic engineering, information engineering, computer science, and others. And it has already been applied into various kind of fields, such as health care.

Robots have been applied into health care since decades ago. Health care robots can be roughly divided into several categories:

Healthcare robots for clinical & hospital

Healthcare robots for health analysis

Robots in Outdoor Public Hygiene

Robots in Clinical Diagnosis and Epidemic Control

Robots in Delivery Errands

Despite the success of these robots, there are still several issues that are not being taken

into consideration: health care robots for personal use, especially for elders who requiring of physical exercises and rehabilitation are not be concerned deeply, and the cost of the robots are usually too high to be accepted for normal individual.

Assistant robots for personal use should fulfill the following requirements:

Non-large scale:

Low price on hardware:

In order to evaluate the human states for calculating the health status, . As one of the most common strategies, computer vision especially robot vision is been widely used. Due to the development of human technology, cheap and small digital cameras and signal processing boards with low energy consumption have been proposed.

According to David Marr, information processing should handling the issue of three levels: computational theory level, representation and algorithm level and hardware implementation level. Computational theory mainly handles the problem that what is the goal of this computation, why is it appropriate, and what is the logic of the strategy by which it can be carried out. On the other hand, for representation and algorithm level, it solves the problems such as how can this computation theory be implemented, and what is representation for the input and output in particular. As for hardware implementation level, it consider how can the representation and algorithm be realized physically.

Even though the top level, i.e. computation theory such as algorithms and mechanisms is essential,

1.2 Physical exercise for elderly people

Furthermore, even in the elderly, physical activity in daily life such as walking has the effect of reducing sickness and death. ”The effect of extending lifespan is not related to the history of competitive sports when young, but health at the hobby level of middle-aged and elderly people. It is believed that exercise habits have a great effect. It is desirable to continue gentle exercise that does not cause excessive active oxygen. Excessive exercise is harmful, but moderate exercise is beneficial.

physical exercise is a protective factor for noncommunicable diseases such as cardiovascular disease, stroke, diabetes, and some types of cancer and physical exercise is associated with improved mental health, delay in the onset of dementia, and improved quality of life and well being. Te health benefits of physical exercise are well documented with higher levels and greater frequency of physical exercise being associated with reduced risk and improved health in a number of key areas.

1.3 Importance of human pose estimation for physical exercise

1.4 Introduction to Exertainment

The exertainment's roots can be found in game peripherals released in the eighties. By June 2009, health games were generating revenues of $2 billion, largely due to Wii Fit's 18.22 million sales at the time. The term exertainment entered the Collins English Dictionary in 2007.

The exertainment has been promoted as a way to improve users' health through exercise, but few studies have been undertaken to measure the health benefits. Smaller trials have yielded mixed results and have shown that the respective traditional methods of exercise are superior to their video game equivalents. Design considerations for exertainment include the need to balance the physical effectiveness of the exercise with the attractiveness of the game play, with both factors needed to be adapted to the abilities of the player, referred to as 'dual flow'.

Laboratory studies have demonstrated that some exertainment can provide light to moderate intensity physical activity.

Exercise games have also proven to be an effective supplement for rehabilitation programs during the COVID-19 pandemic, including balance rehabilitation for the elderly. Children are oftentimes more receptive to the idea of exertainment, making it an especially helpful tool in motivating ill children in their rehabilitation efforts.

A 2018 systematic review in the Journal of Medical Internet Research of 10 randomized trials studying the "Social Effects of Exertainment on Older Adults" found that "the majority of exertainment studies demonstrated promising results for enhanced social well-being, such as reduction of loneliness, increased social connection, and positive attitudes towards others".

Another 2018 systematic review of 10 randomised controlled trials of exertainment in overweight children found that they can produce a small reduction in body mass index.

As of 2016, exertainment for those with neurological disabilities had been studied in around 140 small clinical trials in people of all ages, to see if exertainment can help this group get enough physical exercise to maintain their health. This mode of getting exercise appears attractive in this population from a public health perspective because of its low cost and accessibility. Exertainment have the potential to provide moderate intensity exercises in this population, but the evidence was too weak on long-term follow-up to draw strong

conclusions.

There is significant evidence across multiple random controlled trials relating exertainment to improved cognitive functioning in healthy older adults (with a mean age of 69), and attenuated deterioration or improvement in adults with cognitive impairment from neurodegenerative diseases such as Alzheimer's disease.

In addition, studies investigated if exertainment can lead to improvements in cognitive performance in clinical and non-clinical populations such as those who have ADHD and depression. There are first encouraging results, but the empirical evidence still is limited.

Studies have shown that exertainment helps manage anxiety in several ways. Exertainment help lower anxiety levels in various clinical populations such as patients with Parkinson's disease, enrolled in cardiac rehabilitation, with fibromyalgia, and with systemic lupus erythematosus by introducing more permanent positive physiological changes than methods that do not involve exercise do.

Exertainment are accessible to many differently-abled patients, as some have settings that allow the game to remember a person's range of motion, whether they have any assistive devices, and general physical ability. These games can also be beneficial for blind or low-vision people, as spaces for physical fitness are often inaccessible for them. Exertainment have proven to be effective in teaching these groups new exercises and have been able to give audio feedback.

1.5 Contributions and goals

With the development of human technologies, especially for robotics and cyber technologies, more and more roles came to appeared. There is a stronger and stronger relationship between cyber and physical, and it is shown in Figure 1.2, there are a circular for the cyber-physical space for people, especially elderly and disabled people.

For the first contribution of this dissertation, we mainly focus on the theorem and implementation between robot and people, which can be seen in the red area. In this dissertation, we proposed a systematic framework for elder care and rehabilitation for indoor use. The framework includes observation of elder posture, physical exercise and rehabilitation & entertainment.

From the academic point of view, we focus on the discussion of initialization strategies of evolutionary computation for time series on robot vision. Even though Evolutionary Computation follow the principle of the nature that the fittest individual would be survival, there is a issue that is often not being taken seriously: evolution in nature cost years and years for waiting for the fittest individual from different kinds of possibility. On the other hand,

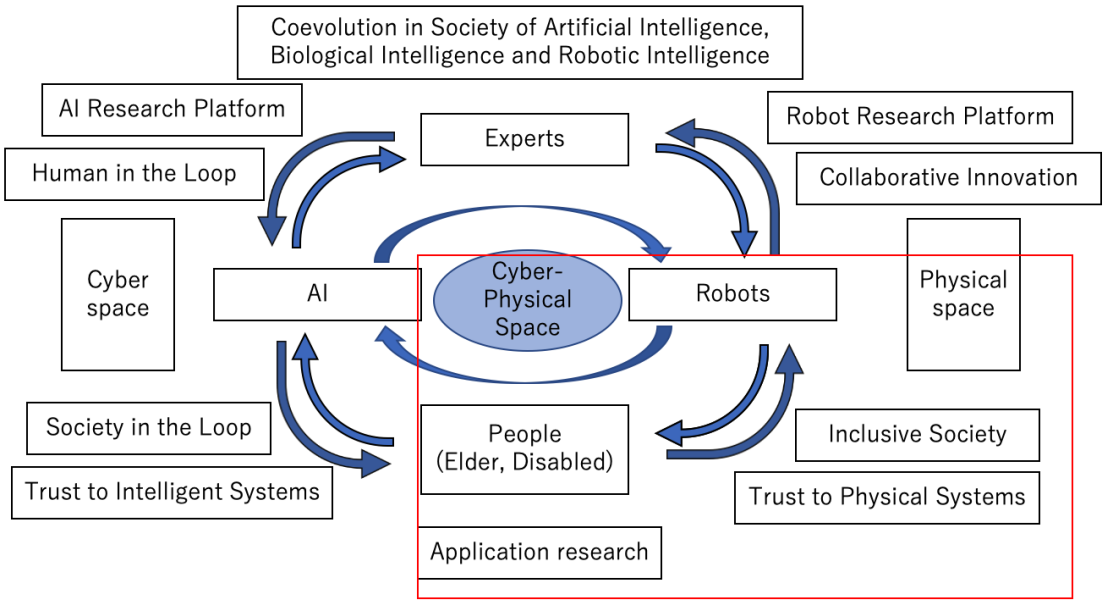


Figure 1.2: *Illustration of structure for current fundamental research.*

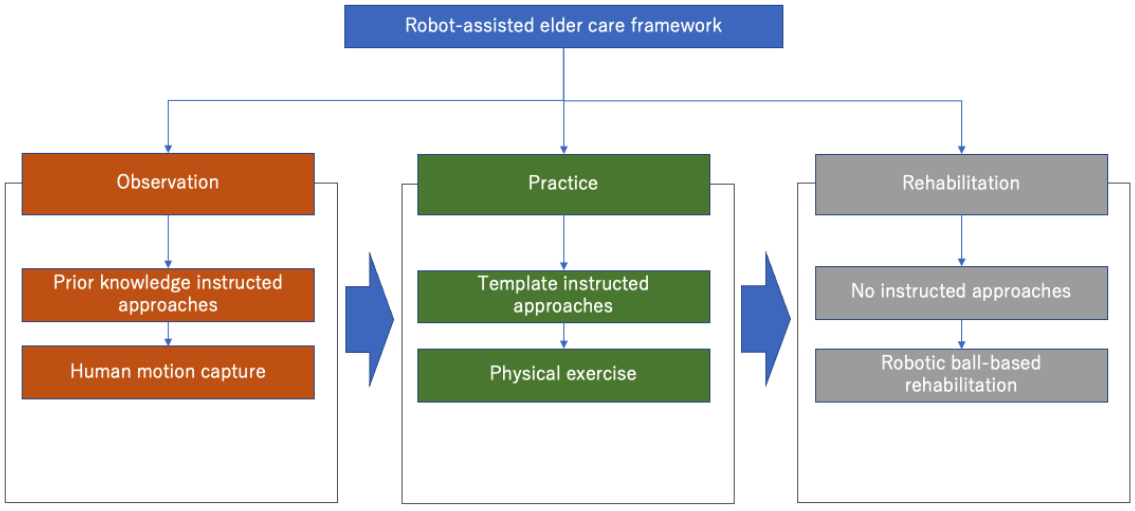


Figure 1.3: *A framework of concept in this dissertation.*

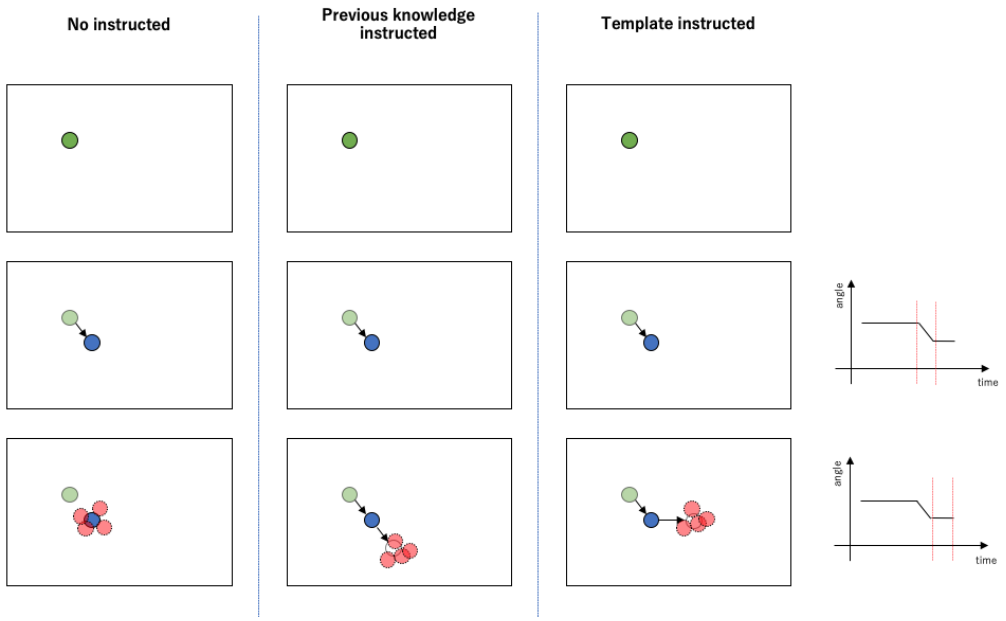


Figure 1.4: *Illustration of the differences among no instructed, previous knowledge instructed and template instructed EC.*

evolution by menu intervention would be much faster on the contrary.

In this dissertation, we discussed Evolutionary Computations and their performances in robotic vision, and category them with three parts based on the evolution instructions: ECs with no instructions, ECs with previous knowledge instructed, and ECs with template instructed. The illustration is shown in Figure 1.4. For ECs with no instructions, populations mutation from various kind of possibilities, which mean there is no limitation for the revolution tendency. In this case, a stable optimization issue would fit to this . This is most similar to the nature principle: creatures evolution to different various possibilities with no specific directions, and selected by nature with fittest evolution.

On the other hand, previous knowledge provide important information for the evolution tendency. For instance, particle swarm optimization calculate the deviation between population and personal best and global best respectively, then force the population to the direction.

1.6 Organization of this dissertation

The thesis is basically organized as 6 chapters, which is shown in Figure 1.5.

Chapter 1 introduces the social and theoretic background the current issues. Next, the contribution and structure of this dissertation are explained.

Chapter 2 explains the concept and current state of exertainment and the theory and

methodology of computer vision, and clarifies the goal of this thesis and the importance of human motion analysis in exertainment.

In Chapter 3, I explain several estimation methods of joint angles of human posture proposed in this study, and discuss the estimation performance of evolutionary robot vision between RGB-D camera and monocular camera. First, I propose a method for estimating human postures by growing neural gas and evolutionary algorithm from the point cloud measured by a RGB-D camera. Next, I propose another method for estimating human postures by particle swarm optimization to reduce the computational cost. Furthermore, I propose a method for estimating human postures by evolutionary algorithms with a monocular camera. I compare the performance of the proposed method with conventional methods. Experimental results show that the performance obtained by the monocular camera is almost the same as that of other methods, but the computational cost of the proposed human posture estimation by monocular camera is the lowest among them.

In chapter 4, I propose multi-view evolutionary robot vision for the human posture estimation. First, I propose a method of evolutionary strategy sample consensus (ESSAC) for selecting correct pairs of corresponding points, and estimating internal parameters of cameras in two or more smart devices set with different views. As a result, it is possible to estimate internal parameters of cameras embedded in two or more smart devices set with different views. Next, I propose a method for estimating human postures from the measurement result of multi-view human motions. Experimental results show that the proposed method can reduce computational cost, while achieving similar or higher accuracy of estimating human postures in ill-posed conditions.

In chapter 5, I develop an exertainment system based on human posture estimation in order that two or more people enjoy physical exercise together in a local community. I implement several physical exercises developed for elderly people on smart devices. Next, I develop an exertainment system using postures including in the above physical exercises that two or more people play together. Furthermore, I develop an exertainment system using robot balls which is simulated as Boccia. Preliminary experimental results show that the proposed system can evaluate the rhythm-motion synchronization by two people. Finally, I show the effectiveness of the proposed method by the multi-view evolutionary robot vision through various types of experiments on exertainment.

Chapter 6 concludes the thesis, and discusses future works towards the social implementation.

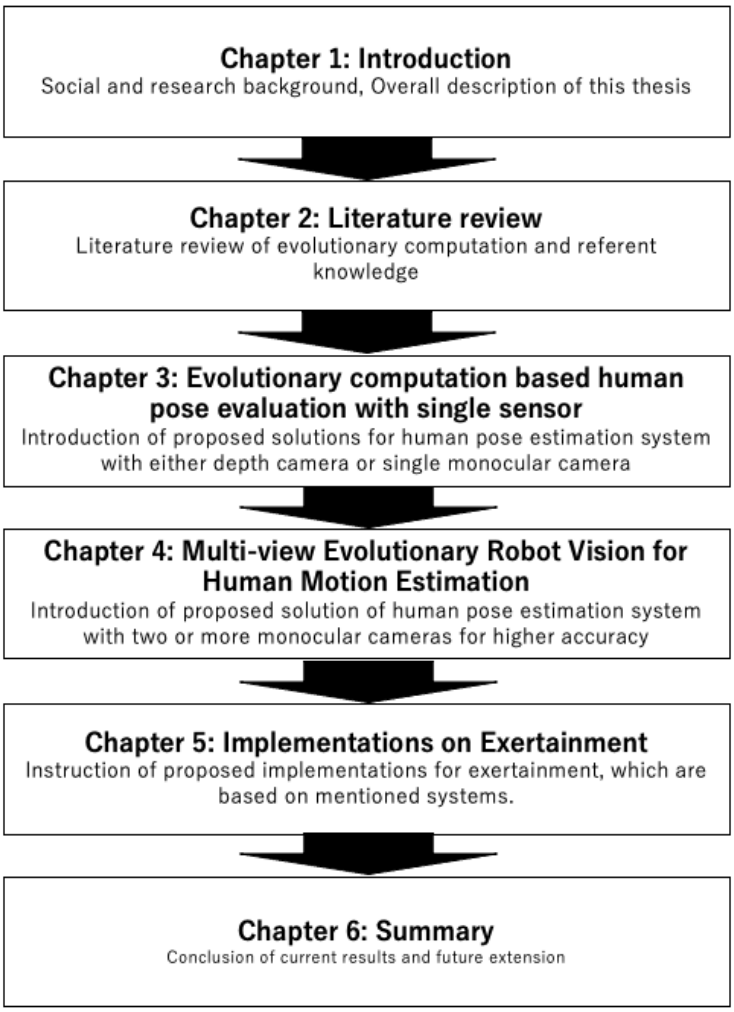


Figure 1.5: *Construction of this dissertation.*

Chapter 2

Referent knowledge for Exercise with Entertainment

In this chapter, I briefly introduce evolutionary computation and some typical evolutionary algorithms such as genetic algorithm (GA), particle swarm optimization (PSO), etc. I also introduced other referent knowledge which is related to this thesis.

2.1 Vision

Vision - according to Oxford English Dictionary - means the ability to see, or the area that you can see from a particular position. It is a human sense of sight, which is also considered as the most complex one of all human sense. And it turned out to be one of the final frontiers research field since the breakthrough of computer technologies. In 1982, David Marr described it from another kind of perspective: vision is a process that produces from images of the external world a description that is useful to the viewer and not cluttered with irrelevant information[7].

2.1.1 Computer vision

As the development of technology, humans enjoy the convenience of the lives which is brought by the computers. Human also enjoy living with assistance of computers. As a significant field of computer science, computer visions also plays an important role .

Computer vision is scientific field of vision that deals with how computers can understanding from digital images or videos. It can be regarded as a combination of image processing and pattern recognition [8]. The input of computer vision is definitely digital images

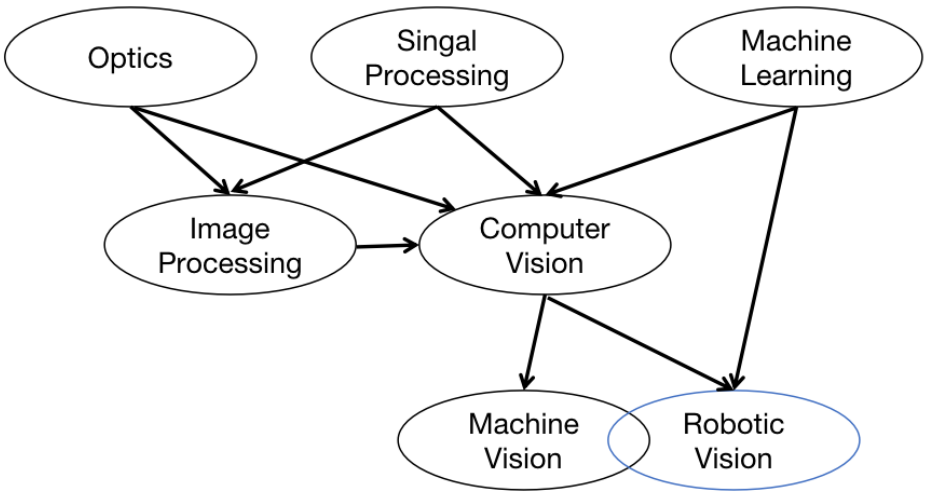


Figure 2.1: *Data processing sequence for robotic vision*

and videos, and the output of the computer vision process is the information for the understanding of images. Development of this field is done by adapting the ability of human vision in taking information. Computer vision is the discipline of extracting information from images, as opposed to Computer Graphics. The development of computer vision depends on the computer technology system, whether about image quality improvement or image recognition. There is an overlap with Image Processing on basic techniques, and some authors use both terms interchangeably.

2.1.2 Robotic vision

The concept of robot vision was firstly proposed in the 1980s. Robot vision is a combination of camera hardware and computer algorithms which process captured visual data. A programmed robot which is capable of visualizing the surrounding environment was developed by the researchers. The robot vision enables the self controlled robots to track and detect many objects at the same time. The data process sequence of robot vision is shown in Figure 2.1.

In basic terms, robot vision involves using a combination of camera-like hardware and computer algorithms to allow robots to process visual data from the world. Generally, robot vision is also often used with machine vision interchangeably.

Even though, there are still a few differences between robotic vision and others. Unlike research of pure computer vision, robot vision should incorporate aspects of robotics into techniques and algorithms such as kinematics, reference frame calibration and the robot's ability to physically affect the environment. The differences between robot vision and other

Table 2.1: *Comparison of different processing schemes.*

Processing Scheme	Input	Output
Signal processing	Electrical signals	Electrical signals
Image processing	Images	Images
Computer vision	Image	Information/feature from images
Machine vision	Images	Information from images
Robot vision	Images	Response on robots

main processing schemes is shown in Table 2.1.

Also, robot vision is not only an engineering domain. It is a science with its own specific areas of research.

At an abstract level, the goal of computer vision issues is to utilize the observed image data to infer something about the world[8]. Concretely, it has been applied into various fields such as optical character recognition, surveillance system, motion capture and so on.

As a sub field of computer vision,

2.1.3 Input data for robotic vision

As one of the significant part for robots, sensors capture the data from the environment in order to make a response. There are various kind of sensors for capturing input data, such as laser scanner, global positioning system, etc. However, for robotic vision, camera is always the most chosen one as the frame capture. According to the captured data type, cameras can be categorised into the following types:

1. Monocular Camera: most common type of cameras in our society. As its name, it is constructed with a monocular, which is a modified refracting telescope used to magnify the images of distant objects by passing light through a series of lenses and usually prisms.

2. Stereo Camera: camera with two or more lenses with a separate image sensor or film frame for each lens. This allows the camera to simulate human binocular vision, and therefore gives it the ability to capture three-dimensional images, a process known as stereo photography. Stereo cameras may be used for making stereo views and 3D pictures for movies, or for range imaging. The distance between the lenses in a typical stereo camera (the intra-axial distance) is about the distance between one's eyes (known as the intra-ocular distance) and is about 6.35 cm, though a longer base line (greater inter-camera distance) produces more extreme 3-dimensional.

3. Time-of-Flight Camera: a range imaging camera system that employs time-of-flight techniques to resolve distance between the camera and the subject for each point of the image, by measuring the round trip time of an artificial light signal provided by a laser or an LED. Laser-based time-of-flight cameras are part of a broader class of scannerless LIDAR, in which the entire scene is captured with each laser pulse, as opposed to point-by-point with a laser beam such as in scanning LIDAR systems.

4. Others: Despite the cameras that are introduced above, there are also some kinds of special cameras for special use, such as thermographic camera, 3D laser scanner, etc.

Different kinds of cameras lead to the different data types, and also lead to the different processing methods.

2.1.4 Time series on robot vision

Time series analysis is a branch of statistics which deals with techniques developed for drawing inferences from time series. Time series denotes a data storing format, it is a sequence of data points, which are collected at regular intervals. As its name, it consists of two mandatory components: time units and the corresponding value assigned for the given time unit. Regarding to the format of time unit, it can be either a discrete stochastic process or a continuous stochastic process.

The basic objective usually is to determine a model that describes the pattern of the time series. Uses for such a model are:

To describe the important features of the time series pattern.

To explain how the past affects the future or how two time series can “interact”.

To forecast future values of the series.

To possibly serve as a control standard for a variable that measures the quality of product in some manufacturing situations.

Time series exits in almost everywhere in robot vision. For instance, dynamic gesture recognition is one of the significant research field in robot vision, which enable robots to recognition the commands from human with different dynamic gestures. The essence of dynamic gesture is just a time series of postures. A good understanding of this time series leads to high performance on human-robot interactions.

For robot vision, robots needs to learn from environment continuously. What is time series on robot vision.

What is the problem of time series on robot vision.

2.2 Evolutionary computation

2.2.1 Soft Computing

The main characteristics of traditional computing (hard computing) are strictness, certainty and precision. But hard computing is not suitable for dealing with many problems in real life, such as driving a car. Soft computing (SC) achieves low-cost solutions and robustness through fault tolerance for uncertain, inaccurate, and incomplete truth values. It simulates the biochemical processes (human perception, brain structure, evolution, immunity, etc.) of intelligent systems in nature to effectively handle daily tasks. Soft computing includes several calculation modes: fuzzy logic, artificial neural network, genetic algorithm and chaos theory. These modes are complementary and coordinate with each other, so they are used in combination in many application systems.

Soft computing (soft computing) is the general term for several applicable computing technologies that have emerged with the development of information technology and computer intelligence, namely fuzzy logic control (fuzzy logic control), neural network (neural network) and genetic algorithm (genetic algorithm). Unlike traditional ”hard computing”, soft computing does not pursue the exact solution of the problem, but allows for imprecision and uncertainty. What is obtained is an approximate solution to the exact or inexact problem. This is the problem solved by the human brain. Embodiment.

Another basic content of soft computing is genetic algorithm. In a broader sense, genetic algorithm is an evolutionary computation technology, which also includes evolutionary programming and evolutionary strategy.

2.2.2 Evolutionary computation

As its name, evolutionary computation (EC) is the application of Darwinian principles to automated problem, and it can be tracked back to the 1940s [9], which is long before the breakthrough of computer technologies. The concept of EC is quite similar to the nature principle: given a population of individuals within some environment that has limited resources, competition for those resources causes natural selection (survival of the fittest). This in turn causes a rise in the fitness of the population. Given a quality function to be maximised, we can randomly create a set of candidate solutions, i.e., elements of the function’s domain.

For theoretic definition, evolutionary computation is a family of algorithms for global optimization inspired by biological evolution, and the sub-field of artificial intelligence and soft computing studying these algorithms in computer science.

Comparing with other kind of approaches, EC mainly shows the following advantages:

1. **Best solution sometimes:** ECs perform powerful performances for solving a wide variety of problems.

2. **Easy implemented:** ECs require little time from the specification of the data structure for candidate solutions (which must be done anyway) to a running algorithm providing reasonably good results in general cases.

3. **Easy parallelized and distributed:** EAs can be parallelized and distributed really well: Since all the individuals in their populations are created and evaluated independently, the most time-consuming parts of handling a population of n individuals can be distributed to up to n processors or computers. The time required by one iteration of an EA can thus be reduced to almost the same time required by one step of a local search method, while providing the power of global optimization. This is particularly useful in cases where the objective function is costly to evaluate, e.g., where testing a candidate solution involves long-running simulations or computations.

4. **Easy combined:** Although plain Evolutionary Algorithms often perform worse than local search methods, they provide a global optimization ability that may discover better solutions if given enough (a lot of!) time. Integrating local search (or other algorithms like branch and bound) into EAs may combine both positive traits and is the aforementioned active area of research of hybrid or memetic algorithms.

5. **Inherently suitable for multi-objective optimization:** To the best of our knowledge, ECs are the only optimization methods which are directly and inherently suitable to deal with multi-objective optimization tasks, i.e., tasks where we have to trade-off between multiple, conflicting optimization goals.

The contemporary terminology denotes the whole field by evolutionary computing, the algorithms involved are termed evolutionary algorithms, and it considers evolutionary programming, evolution strategies, genetic algorithms, and genetic programming as subareas belonging to the corresponding algorithm variants.

2.2.3 Evolutionary computation on robotic vision

We then apply the quality function to these as an abstract fitness measure – the higher the better. On the basis of these fitness values some of the better candidates are chosen to seed the next generation. This is done by applying recombination and/or mutation to them. Recombination is an operator that is applied to two or more selected candidates (the so-called parents), producing one or more new candidates (the children). Mutation is applied

to one candidate and results in one new candidate. Therefore executing the operations of recombination and mutation on the parents leads to the creation of a set of new candidates (the offspring).

Nowadays, we can say that Evolutionary Computer Vision (ECV) represents a new research avenue towards the design of autonomous systems with visual abilities using the main paradigm the art, theory and technology of evolutionary computing. Figure 2.10 depicts a futuristic scenario where a humanoid robot is in charge of domestic tasks like starting a fire in a cozy fireplace. Today, however, robots like Nao are unable to match the abilities of a domestic animal such as a dog. The reason is the lack of a viable methodology that serves to create truly intelligent agents. In this way, the challenge of endowing a humanoid robot with cognitive and mental abilities has a direct relationship with the goals of computer vision. Moreover, an autonomous system should be able to know its goals, understand itself, be aware of its environment, understand the priorities of its goals, and direct itself to achieving its goals according to these priorities.

Today, it seems incontrovertible, at least in the computer vision and robotics communities, that the human visual system exhibits complex design. Moreover, explanations involving concepts such as purpose and evolution have been discussed without any controversies; see the dialogue and the replies in [257, 61, 232]. Here, the authors question the right approach to follow, between reconstructivism and purposivism, in order to accomplish with success the multiple tasks of computer vision. The discussion is also linked to the notion of behavior from the standpoint of developing a well-defined set of complex visual functionalities. These concepts have been discussed and are actually the foundation of cybernetics; see [225]. Note that in this last example the exposition follows an Aristotelian argument.

2.3 Introduction of Evolutionary Algorithms

Evolutionary computation is named according to its inspiration from the evolution process of nature: individuals that can fit to the environment would have more possibility of survival. And algorithms that followed the rules of EC are so-called Evolutionary algorithms (EAs).

EAs are algorithms that perform optimization or learning tasks with the ability to evolve. They have three main characteristics:

Population-based: EAs maintain a group of solutions, called a population, to optimize or learn the problem in a parallel way. The population is a basic principle of the evolutionary process.

Fitness-oriented: Every solution in a population is called an individual. Every individ-

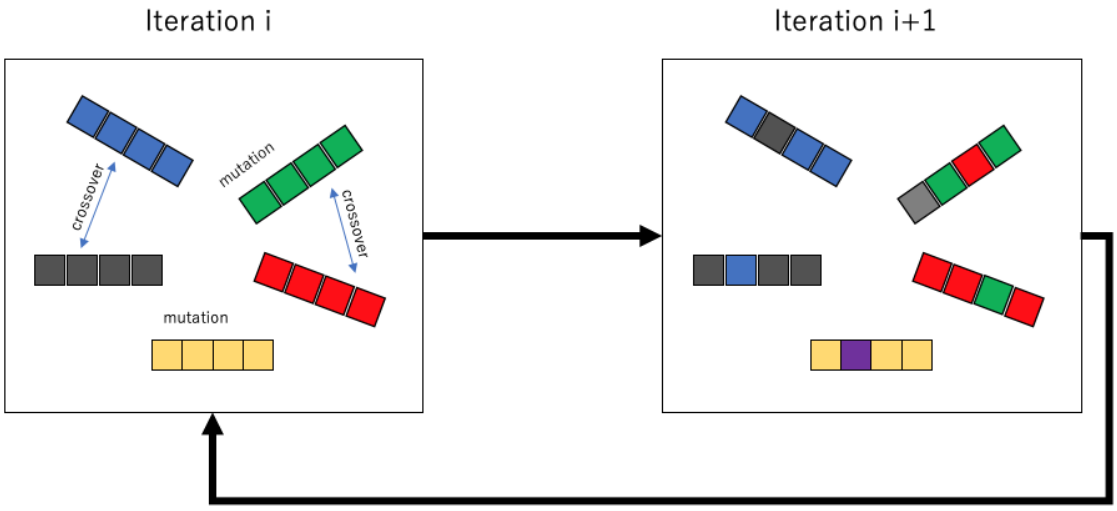


Figure 2.2: *Illustration of genetic algorithm*

ual has its gene representation, called its code, and performance evaluation, called its fitness value. EAs prefer fitter individuals, which is the foundation of the optimization and convergence of the algorithms.

Variation-driven: Individuals will undergo a number of variation operations to mimic genetic gene changes, which is fundamental to searching the solution space.

In 1950s and 1960s, computer scientists started to introduced EC as computation methods. In the early time, ECs were proposed and developed from several different categories: Fogel, Owens, and Walsh introduced evolutionary programming [10]; while Holland proposed the famous genetic algorithm [11]. Meanwhile, in Germany, Rechenberg and Schwefel invented evolution strategies. In the early 1990s a fourth stream following the general ideas emerged, genetic programming, championed by Koza [12]. For about 15 years these areas developed separately, but since the early 1990s they have been viewed as different perspective of one technology that has come to be known as evolutionary computing.

2.3.1 Genetic Algorithm

Genetic Algorithms(GAs) is proposed by John H. Holland[13] and it is a search-based optimization technique. They are a particular class of evolutionary algorithms that use techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover. The fittest individuals are selected for reproduction in order to produce offspring of the next generation.

The main procedure for genetic algorithm is basically constructed by the following steps:

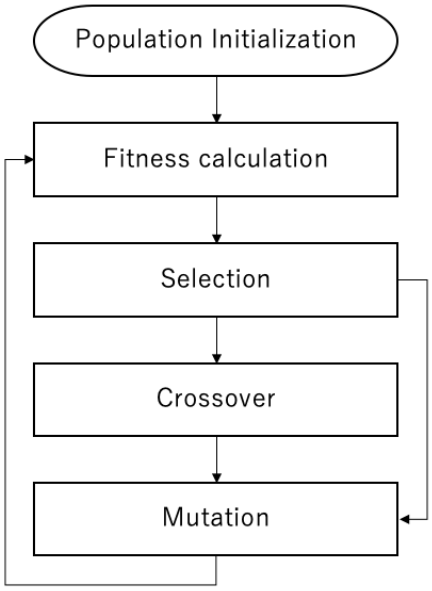


Figure 2.3: *Procedure of genetic algorithm*

1. Population initialization
2. Fitness calculation
3. Selection
4. Crossover
5. Mutation

The whole procedure is shown in Figure 2.3

2.3.1.1 Population initialization

As the very first step of genetic algorithm, initialization is a significant phase. Good initialization strategy would made the optimization procedure much faster and avoidance of local optimal, whereas a bed initialization would be move to unexpected results such as the condition known as Premature Convergence.

In a genetic algorithm, the set of genes of an individual is represented using a string, in terms of an alphabet. Usually, binary values are used (string of 1s and 0s). I say that I encode the genes in a chromosome.

An individual is characterized by a set of parameters (variables) known as Genes. Genes are joined into a string to form a Chromosome (solution).

2.3.1.2 Fitness calculation

As mentioned previously, in genetic algorithms, each candidate solution is generally represented as a string of binary numbers, known as chromosome. And I have to test the performance of these candidate solutions and select the best one as the fittest solution for the given problem. Therefore, it is necessary to numerically define the performance of these candidate solutions.

The fitness score just determines how fit an individual is. In order to evaluate fitness score, it is necessary to design a fitness function for calculating.

2.3.1.3 Selection

As by following Darwin's theory of evolutionary, only best individuals can survive in the procedure of reproduction. In the process of genetic algorithm, selection is a step of finding out the best individuals for mating process.

2.3.1.4 Crossover

Crossover might be the most significant phase in a genetic algorithm. It can be regarded as an exploration operator. For each pair of parents to be mated, a crossover point is chosen at random from within the genes. Performance of genetic algorithms mainly depends on selection of genetic operators which involve crossover and mutation operators. The following paragraphs present some typical crossover strategies.

2.3.1.5 Mutation

In certain new offspring formed, some of their genes can be subjected to a mutation with a low random probability. The purpose of mutation operation is to change the genes of the offspring and to increase the diversity of the population. This phase enables GA to jump out of local optimal.

2.3.2 Particle Swarm Optimization

As a subcategories of EC, swarm intelligence (SI) has been widely implemented into various kind of fields. The concept of swarm intelligence is inspired from nature world, which is always a large group of animals such as ants, bees, birds, fishes, etc. These different communities inspired various kinds of SI algorithms such as particle swarm optimization (PSO), ant colony optimization (ACO), differential evolution(DE), etc. Despite this diversity,

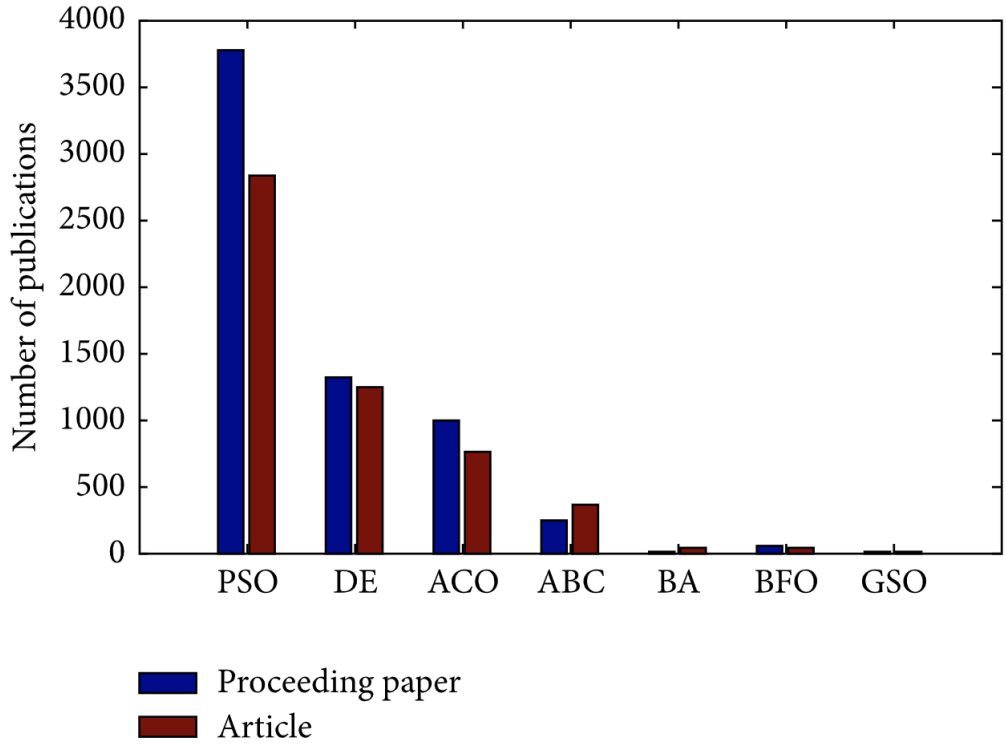


Figure 2.4: *Number of publications for each SI-based algorithms*

concentration on PSO is much higher than other algorithms, and this situation is impressed by number of related publication shows this tendency in Figure 2.4 [14].

Several studies regarding the social behavior of animal groups were developed in the early of 1990s. These studies showed that some animals belonging to a certain group, that is, birds and fishes, are able to share information among their group, and such capability confers these animals a great survival advantage [15]. Inspired by these works, Kennedy and Eberhart proposed in 1995 the PSO algorithm [16], a meta-heuristic algorithm that is appropriate to optimize nonlinear continuous functions. Seemingly no evolutionary events happen in a PSO procedure, but algorithmically it does fit in the general EA framework.

2.3.3 Standard Particle Swarm Optimization

In a PSO algorithm, a set of candidate solutions(which are also called particles generally) to an optimal problems are updated according to some form of interaction among them. The PSO is initialized with a set of random particles $\{x^i\}_{i=1}^N$, where N is the number of particles. And the search for optimal solution iteratively in the search space. Each particle has a corresponding fitness value as well as its own velocity . The same as GA, the fitness value is calculated by an observation model and the velocity provides the direction of particle

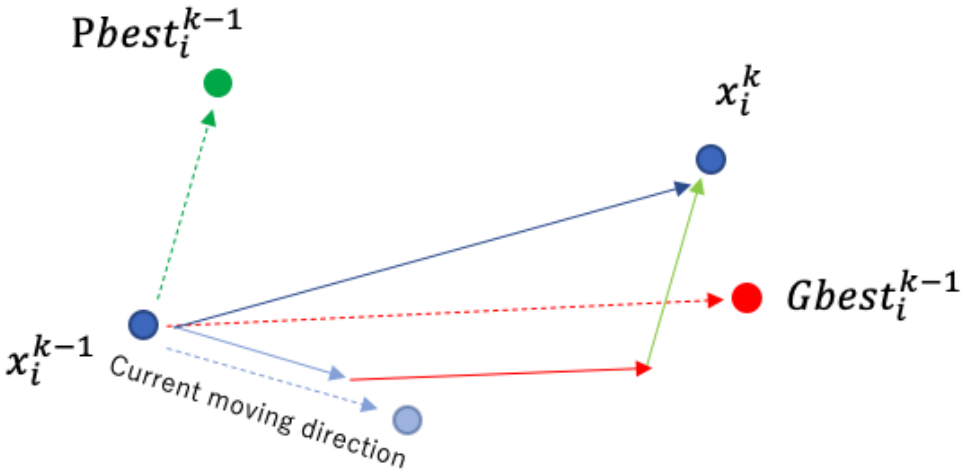


Figure 2.5: *Illustration of movement direction define particle swarm optimization.*

movement.

In each iteration, the i_{th} the movement of particles mainly depends on two factors: its individual best position $Pbest_i$, which is originated by i_{th} particle so far; the global best position $Gbest$, is the overall global best position that has been generated by entire swarm. In the k_{th} iteration, each particle updates the position and velocity by utilizing the following equations:

$$v_i^k = w_1 v_i^{k-1} + w_2 r_1 (Pbest_i^{k-1} - x_i^{k-1}) + w_3 r_2 (Gbest^{k-1} - x_i^{k-1}) \tag{2.1}$$

where v_i^k is a vector representing the angle velocities of the i_{th} agent at k_{th} iteration. v_i^k is controlled by three factors: the global best velocities $Gbest_i$, the personal best velocities $Pbest_i$ and the previous velocities v_i^{k-1}. Therefore, the current direction for x_i^k will be calculated as

$$x_i^k = x_i^{k-1} + v_i^k \tag{2.2}$$

One may notice that the idea behind this term is that as the particle gets more distant from the $Pbest_i^{k-1}$ position, the difference $(Pbest_i^{k-1} - x_i^{k-1})$ must increase; therefore, this term increases, attracting the particle to its best own position. The parameter c1 existing as a product in this term is a positive constant and it is an individual-cognition parameter, and it weighs the importance of particle's own previous experiences. The other parameter that composes the product of second term is r1, and this is a random value parameter with [0,1]

range. This random parameter plays an important role, as it avoids premature convergences, increasing the most likely global optima.

2.4 Robot kinematics

Kinematics is the study of the relationship between a robot's joint coordinates and its spatial layout, and is a fundamental and classical topic in robotics, it is the branch of mechanics that studies the motion of a body or a system of bodies without consideration given to its mass or the forces acting on it, which is similar as human's

introduction to kinematics...

Chapter 3

Evolutionary computation based human pose evaluation with single sensor

In this chapter, I introduced the previous knowledge based strategy for robotic vision, and its implementation.

3.1 Related Works

Researchers have been working on the referent field for a long time[17, 18, 19]. Consequently, various methods have been proposed. In this section, I give a brief review for several typical methods which have been widely accepted and implemented. In [20], Aggarwal, J. developed a taxonomy that divided all the proposed methods into two main groups: single-layer approaches and hierarchical approaches.

In [21], the authors proposed the supervised pose recognition method, which can be considered the most popular method today. The process described in the paper is mainly divided into two steps: first, the body part is marked from a single depth image for segmenting, and then the key nodes are marked. The body joint positioning is then performed, and the marked human body parts are remapped into the three-dimensional space to form a highly reliable spatial position for the key nodes.

The authors in [22] proposed a method for human full-body pose estimation from depth data that can be obtained using TOF cameras and the Kinect device. Their approach consists of robustly detecting anatomical landmarks in the 3D data and fitting a skeleton body model using constrained inverse kinematics. Instead of relying on appearance-based features for interest point detection, which can vary strongly with illumination and pose changes, they built a graph-based representation of the depth data to measure geodesic distances between

body parts. As these distances do not change with body movement, it is able to localize anatomical landmarks independently of the pose.

Recently, in[23], the authors proposed a vector-shaped pose descriptor, which allows for the retrieval of similar poses and is easier to use with many machine learning libraries by constructing a feature space for appearances of human poses. This method has improved the limited scope of many methods based on a kinematic or surface mesh model, and performed efficiently in experiment.

Nevertheless, the methods mentioned above are based on supervised learning algorithms, which require a large amount of data for training before they are applied. Differently from the concept of these methods, in this paper, I propose a method for human posture recognition by a series of unsupervised algorithms, which does not require collection of training data.

3.2 Implementation on prior knowledge instructed evolutionary computation

Human posture recognition has been a popular research topic ever since first computer computational recognition method appeared, and many state-of-the-art methods have been proposed by researchers and engineers around the world. It has been applied to various areas, such as human-robot interaction[24, 25, 26], operating simulations, and games development.

Human-robot interaction might be the area with the most applications of posture recognition. It makes it possible for humans to communicate with robots not only though cold commands inputted from the keyboard, but also though gesture language understood by computers[27, 28].

At the same time, with the increase of the number of elderly people all over the world, health care issues are getting more and more important. To solve these issues, human motion capture systems are required for the elders who live alone. And elders' state of health can be monitored by determining his or her posture, and an alert can be given in the case that high risk postures, such as fall down, are detected. These series of systems will reduce the burden of human resources while improving the efficiency of posture recognition[29].

Previous human skeleton recognition research was conducted using standard pin-hole digital cameras. However, it is impossible to detect a human's real spatial posture during a period of time owning to the natural weakness of the pinhole camera. The recently developed electronic devices, such as the RGB-D camera, stereo camera, and depth sensors, have made it possible to capture objects in a real-world shape. In recent years, the most widely used sensors in the referent field are depth cameras.

In general, depth cameras are divided into two types: stereo cameras and time of flight (TOF) cameras. Stereo cameras capture the detail of depth based on the binocular stereo vision theorem[30, 31], which calculates the distance of points in the real world by the principle of parallax and applies two images for measuring the targets that are captured by two parallel cameras from different positions. The method calculates the positional deviation between the corresponding points of the image for obtaining the three-dimensional geometric information of an object. By combining the two images and observing the differences between them, it is possible to obtain a clear sense of depth, establish the correspondence between the features, and map the same spatial physical point in different images.

By contrast, TOF cameras, also referred to flight time cameras, obtain the targets' distance by continuously transmitting light pulses to the targets and receiving the light returned from the object. Next, the camera measures the flight (round-trip) time of the light pulse[32, 33, 34]. This technique is essentially similar to the principle of a 3D laser sensor; however, while the 3D laser sensor on;y scans the target point-by-point, the TOF cameras also obtain the information of flight time from a 2D area.

In this paper, I propose a framework for recognizing human postures by simulating the human body skeleton and its movements according to the 3-dimensional points cloud data using a series of unsupervised learning algorithms.

3.2.1 Proposed Method

In this secrtion, I propose a human posture recognition method based on the concept of unsupervised learning. The framework of this method is shown in Fig 3.1. The learning process can be generally divided into three steps: preprocessing, growing neural gas(GNG) based rough structure generation, and parameters optimization by particle swarm optimization(PSO). Detail of each step will be given in the following section.

3.2.1.1 GNG for Human Structure Construction

The particle points cloud data is computationally costly without peprocessing the data. The GNG is used for representing the points cloud data to a lower density structure, thus I utilized it for the construction of humans' rough structure.

The GNG is a typical self-organizing map (SOM) algorithms for unsupervised learning. It is known that unsupervised learning algorithms are a series of learning methods that work without any prior input data for training and give the desired output. Input data are consecutively represented by SOM in the form of input signals and the SOM changes its topological structure for representing the input data with the self-adaptation mechanism. Next, a growing

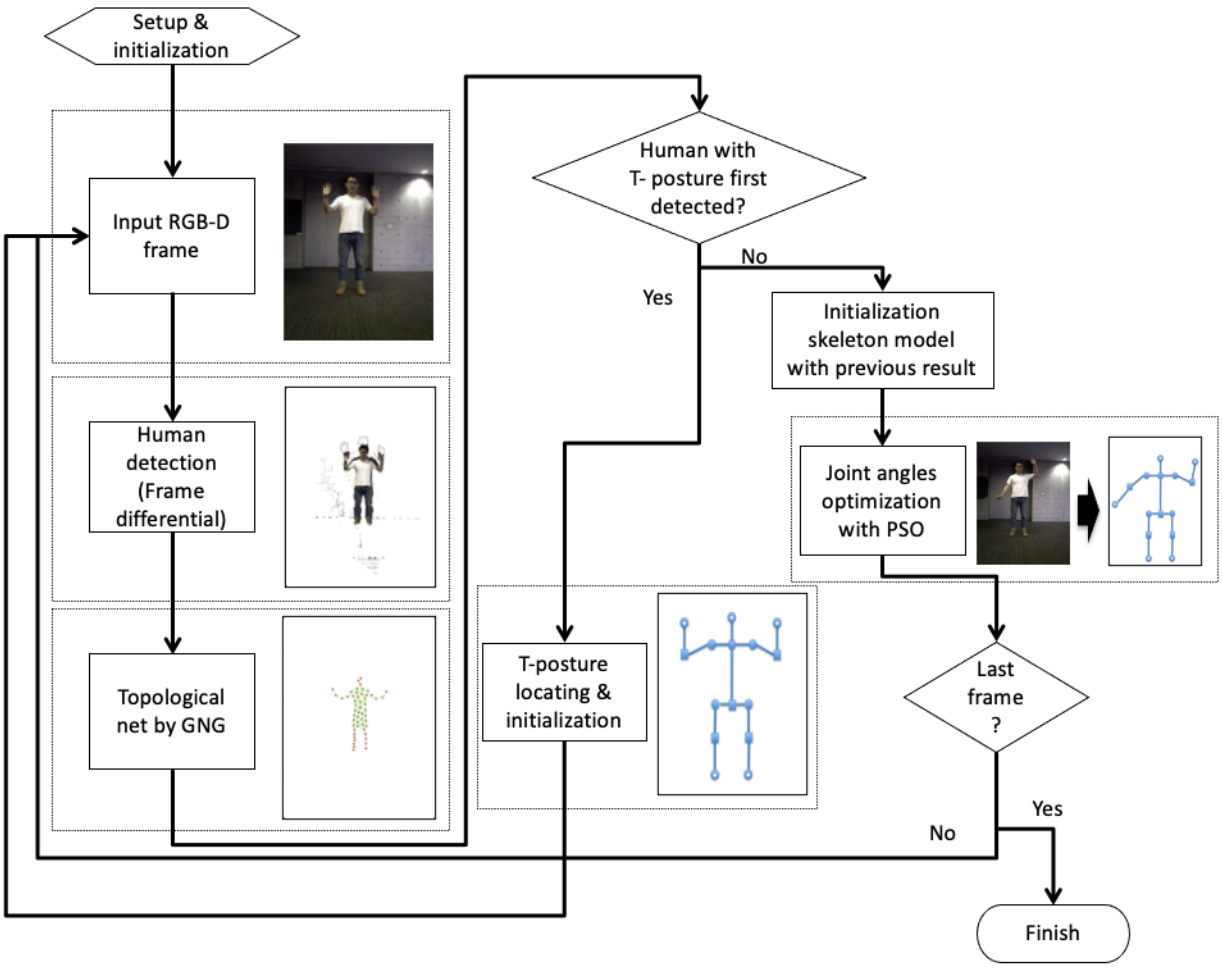


Figure 3.1: *Framework of proposed method.*

mechanism is used for gradual adaptation and self-adjustment of size. The growing neural network starts in some minimal state (e.g., with some minimal number of neurons in the network), which is adapted to the input data. Then, it continually grows (increases its size) and adapts again. This cycle is repeated until the desired resolution of the neural network is achieved.

In the GNG learning algorithm, the following notations are used:

w_i: n dimensional vector of a node ($w_i \in Rn$)

G: set of nodes

N_i: set of nodes connected to the i_{th} node

c: set of edges

$a_{i,j}$: age of the edge between the i_{th} and the j_{th} node

The steps of the standard GNG algorithm are as follows:

Step 0. Initialize the network by creating two nodes at random positions, w_{c_1} and w_{c_2} in Rn. Then, set the connection between them.

Step 1. Randomly generate an input data v according to selecting function $p(v)$, which is the probability density function of data v.

Step 2. Select the nearest unit (winner) g_1 and the second-nearest unit g_2 by:

$$g_1 = argmin_{i \in G} \parallel v - w_i \parallel \tag{3.1}$$

$$g_2 = argmin_{i \in G/g_1} \parallel v - w_i \parallel \tag{3.2}$$

Step 3. Generate the connection a connection between g_1 and g_2, is such connection does not exist already. Set the age of the connection between g_1 and g_2 to zero:

$$a_{g_1,g_2} = 0 \tag{3.3}$$

Step 4. Add the squared distance between the input data and the winner to a local error variable:

$$E_{g_1} \leftarrow E_{g_1} + \parallel v - g_2 \parallel^2 \tag{3.4}$$

Step 5. Update the reference vectors of the winner and its direct topological neighbors by the learning rate η_1 and η_2 respectively, of the total distance to the input data:

$$w_{g_1} \leftarrow w_{g_1} + \eta_1 \cdot (v - w_{g_1}) \tag{3.5}$$

$$w_j \leftarrow w_j + \eta_2 \cdot (v - w_j) \qquad if \quad c_{g_{1,j}} = 1 \tag{3.6}$$

Step 6. Increment the age of all edges emanating from s_1:

$$a_{g_{1,j}} \leftarrow a_{g_{1,j}} + 1 \qquad if \quad c_{g_{1,j}} = 1 \tag{3.7}$$

Step 7. Remove edges with an age larger than a pre-defined threshold. If this results in units having no more emanating edges, remove those units as well.

Step 8. If the error E_q is higher than the predefined threshold, insert a new unit as follows:

Select the unit f with the maximum accumulated error among the neighbors of q.

Add a new unit r to the network and interpolate its reference vector from q and f:

$$W_r = 0.5 \cdot (w_q + w_f) \tag{3.8}$$

Create a new edge that connects the new unit r with units q and f, and remove the existing edge between q and f.

Decrease the error variables of q and f by a fraction α:

$$E_q \leftarrow E_q - \alpha E_q \tag{3.9}$$

$$E_f \leftarrow E_f - \alpha E_f \tag{3.10}$$

Interpolate the error variable of r from q and f:

$$E_r = 0.1 \cdot (E_q + E_f) \tag{3.11}$$

Step 9. Decrease the error variables of all units:

$$E_i \leftarrow E_i - \beta E_i \quad (\forall i \in G) \tag{3.12}$$

Step 10. Continue with step 1 if a stopping criterion (e.g., net size or some performance measure) is not yet fulfilled.

The number of point could be reduced largely from the original point cloud. Therefore the computation time would also be cut down.

Nevertheless, the standard GNG does not perform in dynamic environments. Considering this, an improved GNG, named GNG with utility (GNG-U) was developed by Bernd Fritzke [**?**]. It is only slightly the standard GNG in that it updates not only the local errors but also

the utility U_{g_1} with

$$U_{g_1} \leftarrow U_{g_1} + E_{g_2} - E_{g_1} \tag{3.13}$$

Then, it removes the node g_i if the following inequality is satisfied:

$$\frac{E_{g_i}}{U_{g_i}} > \gamma \tag{3.14}$$

where γ is a parameter that controls the number of nodes.

GNG-U can perform dynamic distributions. Based on this, Y. Toda [**?**] proposed the modified GNG-U (GNG-U2) by introducing the weight vector and has achieved the superior results in 3D structures.

3.2.1.2 Human Skeleton Modeling

In this paper, I utilized a simplified kinematic model for representing the human skeleton. The i_{th} skeleton is constructed as $m \in M$ joints ($M = 15$ in this paper):

$$\begin{aligned} J_i = \{j_i^k\} = \{&center\_shoulder, left\_shoulder, \\ &right\_shoulder, left\_elbow, left\_hand, \\ &right\_elbow, right\_nand, center\_torso, \\ &center\_hip, left\_hip, left\_knee, left\_foot, \\ &right\_hip, right\_knee, right\_foot\} \end{aligned} \tag{3.15}$$

The positions of joints are shown in Fig 3. Each joint in $j_i^k \in J_i$ is represented by 3D coordinates $(x_{j_i^k}, y_{j_i^k}, x_{j_i^k})$.

Subsequently the length of the links between each joints is described as:

$$L_i = \{l_1, l_2 ... l_n\} \tag{3.16}$$

and the spatial coordinates of the skeleton are represented as

$$P_i = \{x_i, y_i, z_i\} \tag{3.17}$$

Despite being more convenient than the previous model in terms of computation, it is still unstable for generating the same skeleton for the same person in different frames. This is because there are two parameters in a skeleton model: the angles and length of each pair of joints. I know that the angles of the joints change dynamicallywhen changing posture, but the length of each pair of joints is always a constant for one person.

Thus, I try to make the preliminary experiment of evaluating the length and angels separately. In this model, there are 16 joints for controlling human posture. I do not calculate the positions of these joints directly. Instead, I generate the lengths and angles for each part as shown in Fig. 2 (initial step), and I then calculate the position of each joints by forward kinematics.

The first part of initialization will solve the parameters of lengths. Given a special posture (e.g. T posture, as shown in Fig 2), I randomly generate different values, and then choose the best one (because the angles are fixed owing to the fixed posture). In the second part, it randomly generates angles and selects the fittest one according to the fitness function.

To compute the fitness of the skeleton model, I introduced the joints of ribs for auxiliary calculation, even though these rib joints have no meanings and can be hidden for human representation.

Because edged nodes contain much more information than the nodes in the center, as they have more probabilities of representing the limbs of the human, it is necessary to give them a higher weight than other nodes.

To weight them, first I search for their geographic center.

Given a weighted skeleton model, our target is to find the optimized parameters of rotation angles to obtain the fitness skeleton. The proposed method is divided into two parts, i.e., the initial step and the prediction step.

3.2.1.3 Denavit-Hartenberg Parameters for Human Skeleton Modeling

Even though the number of nodes is limited, the construction of the skeleton can to achieved. It is difficult to directly model the skeleton. Therefore, I applied a more convenient way for the representation of skeleton parameters.

The Denavit-Hartenberg(DH) parameters representation is the most widely used in kinematics today. It is named after Jacques Denavit and Richard Hartenberg who introduced this representation in 1955 [35, 36]. It calculates the coordinate transformation frame by frame making a list of parameters, with four parameters for each transformation:

1. Rotation angle α about X axis
2. Translation a along X axis
3. Translation d along Z axis
4. Rotation angle θ along Z axis.

Because all of the coordinate systems satisfy the constraint, all of the transformation can be represented by a set of quadruple of parameters as

Table 3.1: *DH parameters of the human body's joints.*

position	joint ID	index	α	θ	a	d
left arm	3	1	0	0	l_2	0
	4	2	θ_1	0	0	0
		3	$\theta_2 + 90^o$	0	0	0
		4	θ_3	0	l_3	0
	5	5	θ_4	0	l_3	0
right arm	6	6	0	0	l_2	0
	7	7	$\theta_5 + 90^o$	0	0	0
		8	0	$\theta_6 + 90^o$	0	0
		9	θ_7	0	l_3	0
	8	10	θ_8	0	l_4	0
head	2	11	$\theta_9 - 90^o$	0	0	0
		12	θ_{10}	-90^o	l_1	0
torso	9	13	$\theta_{11} - 90^o$	0	0	0
		14	θ_{12}	-90^o	l_5	0
left leg	10	15	-90^o	0	l_6	0
	11	16	$\theta_{13} - 90^o$	0	0	0
		17	0	$\theta_{14} + 90^o$	0	0
		18	θ_{15}	0	l_7	0
	12	19	θ_{16}	0	l_8	0
right leg	13	20	90^o	0	l_6	0
	14	21	$\theta_{17} - 90^o$	0	0	0
		22	0	$\theta_{18} + 90^o$	0	0
		23	θ_{19}	0	l_7	0
	15	24	θ_{20}	0	l_8	0

$$T_i = T_{\theta_j} T_{d_j} T_{a_j} T_{\alpha_j} \tag{3.18}$$

where $T_{\theta_j}, T_{d_j}, T_{a_j}, T_{\alpha_j}, (j \in m)$ represent the rotation matrix of the four steps listed above. The combined matrix T_i is

$$\begin{bmatrix} \cos\theta & -\cos\alpha\sin\theta & \sin\alpha\sin\theta & a\cos\theta \\ \sin\theta & \cos\alpha\cos\theta & -\sin\alpha\cos\theta & a\sin\theta \\ 0 & \sin\alpha & \cos\alpha & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.19}$$

I distribute all of the joints according to the parameters in Table 1. In this table, joint $1-4$, $5-8$, $9-12$, $13-16$ represent the DH parameters for the rotation angles of the left arm, right arm, left leg, and right leg, respectively. The indices of the joints and links are shown in Fig 3.2.

By applying the DH parameters shown in Table 1, it is easy to build up the simulated

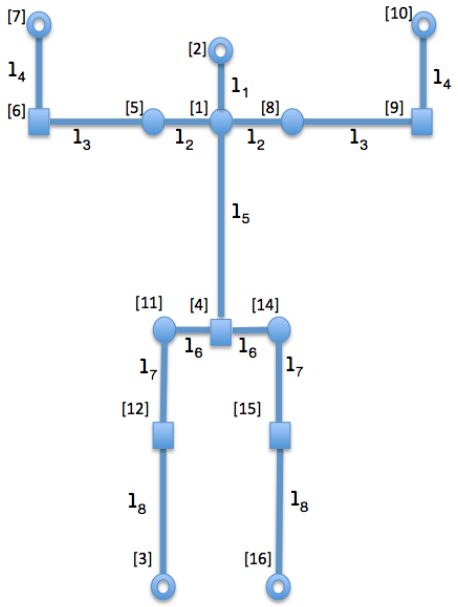


Figure 3.2: *Indices of human body's joints and links.*

human skeleton. Based on the DH parameters, the target for determining a human's posture is optimized by rotational angles of each joints.

3.2.1.4 PSO for Human Posture Recognition

I apply PSO to optimize all of the rotational angles to properly simulate the human posture. The PSO is a computational method that optimizes a problem by iteratively trying to improve a candidate solution according to given measures of quality. It was originally proposed by Kennedy Eberhard and Shi [**?**], and has been extended to a series of advanced solutions.

The motivation from the PSO algorithm is inspired by the flocking behavior of birds nature. It contains a set of particles, where each particle represents a bird in the flock. In this paper, I assume that each particle represents a skeleton candidate with a different series of rotation angles. The purpose is to iterate all of the particles with their velocities of rotational angle, and selecting the best series with the minimum value, which correspond to that the best skeleton for representing the human posture. The speed of the rotational angle in the PSO algorithm can be represented as

$$v_i^k = v_i^{k-1} + w_1 r_1 (Pbest_i - X_i) + w_2 r_2 (Gbest_i - X_i) \tag{3.20}$$

where v_i^k is a vector representing the angle velocities of the i_{th} agent at k_{th} iteration. v_i^k is controlled by three factors: the global best velocities $Gbest_i$, the personal best velocities $Pbest_i$ and the previous velocities v_i^{k-1}. Thus, the current angles for x_i^k will be calculated as

$$\theta_i^k = \theta_i^{k-1} + v_i^k \tag{3.21}$$

Different human postures are generated by the angles calculated from the above equation, depending on the previous knowledge and random factors.

Because the GNG nodes have roughly described structures of the human skeleton, it is necessary to optimize all of the parameters of the skeleton model to optimize the rotational angles. In order to calculate the value of the global best and the personal best, it is necessary to propose a proper evaluation function. Here I optimize the best skeleton by searching for the minimum value of the following evaluating function:

$$F(x) = \sum_{i=1}^{M} w_{g_i} d(g_i) \tag{3.22}$$

where M is the number of GNG nodes that were obtained from the previous step, w_{g_m} is the weight of the m_{th} GNG node g_m, and the function $d(g_m)$ is represented as

$$d(g_i) = \begin{cases} e^{-\frac{\|g_i - j_n\|^2}{2a^2}} & if \ \arg\min_{j_n \in J} e^{-\frac{\|g_i - j_n\|^2}{2a^2}} \leq K \\ \tau & otherwise \end{cases} \tag{3.23}$$

where a is a constant, and τ is a constant threshold with a large value. This evaluation function means that the skeleton with the smallest sum of distances for the total GNG nodes is more likely to be the fittest posture. Considering that each link is represented by a cylinder with a radial threshold of K, the value of $d(i)$ is valuable if and only if the node i is inside this cylindrical space. Nodes that are closer to the axis of the cylinder would have a higher possibility to be part of it.

Despite the advantages of GNG , it is obvious that GNG nodes are generated randomly, which means that each of the node has the same importance for constructing the whole GNG network. However, they should be separated into different levels based on their representations of human body. The reason is that different locations in human body have different importance when representing the human body. For example, the nodes that surround the elbow provide more information than the nodes located in the torso, therefore, these nodes should be given a higher weight when reconstructing the human body.

In this paper, I tried a simple but efficient weighted methods. It is obvious that nodes that are near to the center would have more edges than those located in the edge. Therefore

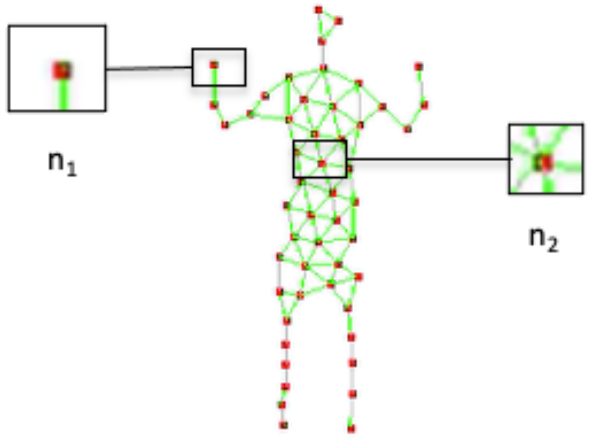


Figure 3.3: *Illustration of GNG node weights.*

I suppose that the weight of the i_{th} node is represented as w_i, then the weight is

$$w_{g_i} = \sum_{g_j \in G} D(g_i, g_j) \tag{3.24}$$

where $D(s_i, s_j)$ denotes

$$D(g_i, g_j) = \begin{cases} 0 & if\ c_{g_i,g_j} = 0 \\ 1 & otherwise \end{cases} \tag{3.25}$$

This also means that the weight of g_i depends on the number of edges linked to it, as shown in Fig 3.3. It is obvious from the figure that the n_1 that contains less edges than n_2 provides more significant information for generating the human skeleton.

3.2.2 Experiment setup and Results

For the experiment in this study, I applied ASUS Xtion PRO live depth camera as the frame capturing sensor, shown in Fig 3.4 and Table **??**. Compared with the other widely used depth cameras, such as Microsoft Kinect, Xtion, it has a low-budget projects or systems.

In the first step, I applied the frame differential algorithm to extract the foreground points, which is regarded as the construction part of human body. Considering that it contains a large amount of noise, I also applied the median filter to reduce them. The result can be seen in Fig 3.6.

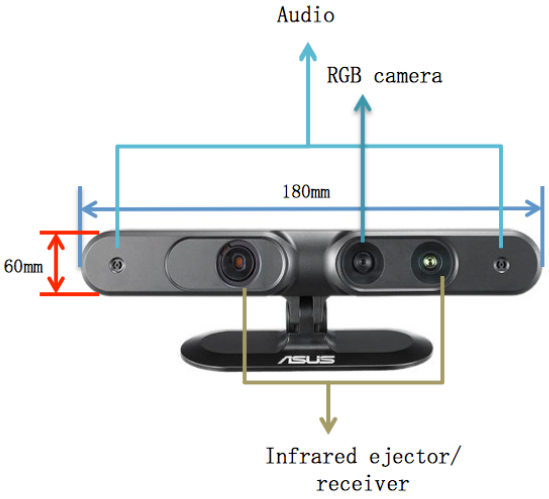


Figure 3.4: *Profile of the Xtion sensor.*

Table 3.2: *Features of the Xtion sensor.*

Weight	490g
Interface	USB2.0
Available view angle	70 degree
fps	30 to 60
Resolution	640*480

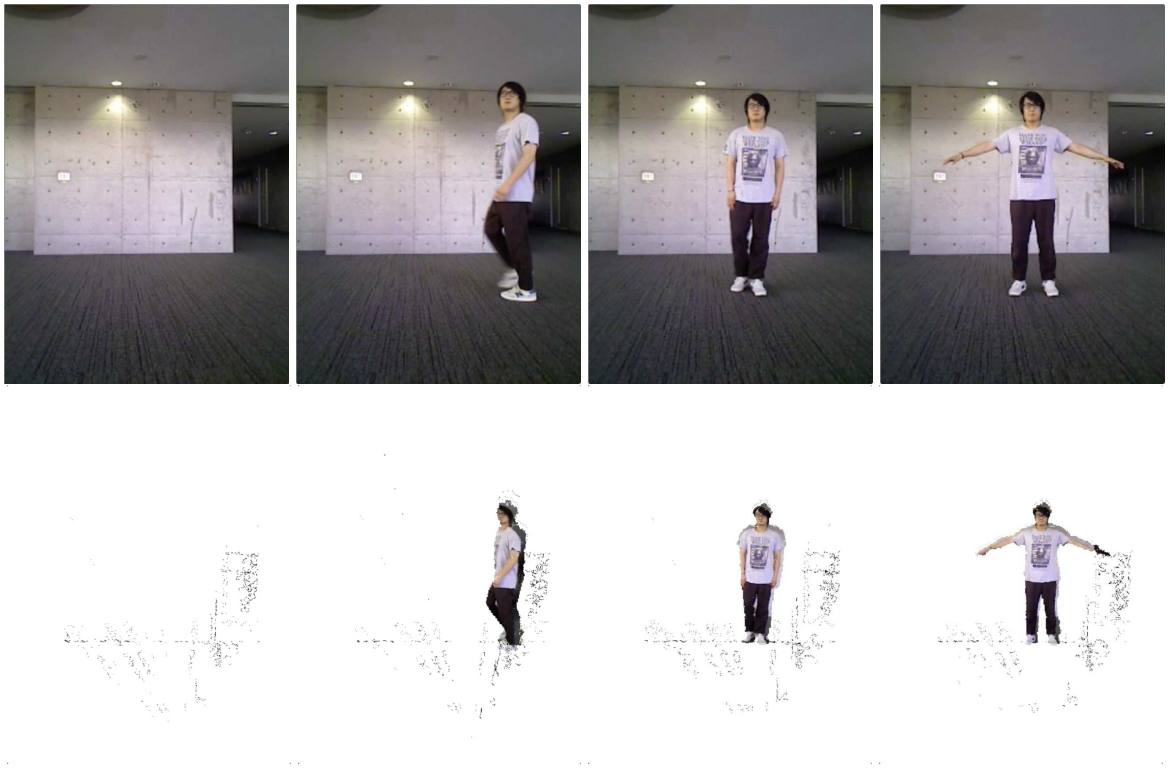


Figure 3.5: *Performance of foreground extraction.*

Figure 3.6: *Experiment result between the standard GNG and the GNG-U2 in a series of input frames. The first row represents the original frame, the middle row shows the standard GNG, and the last row shows the GNG-U2.*

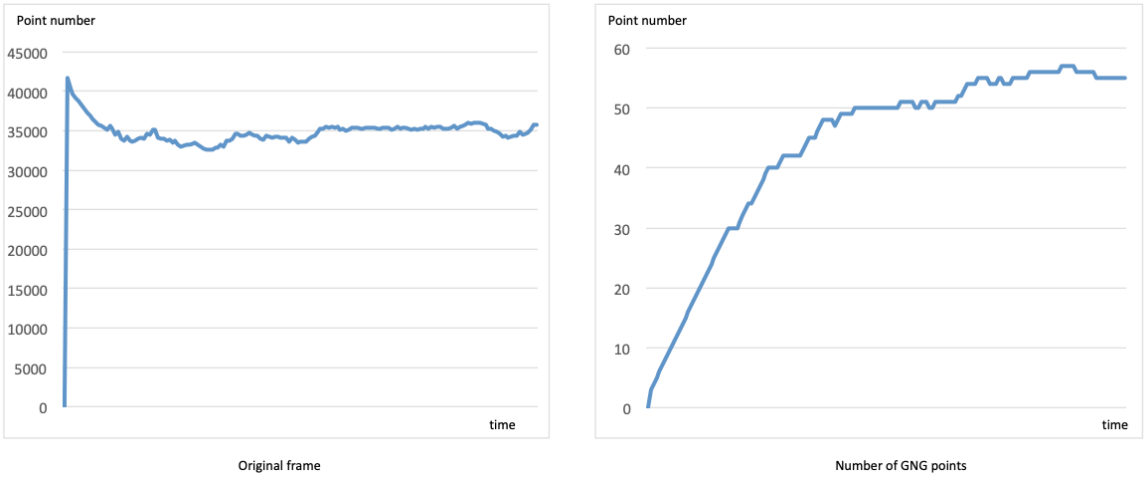


Figure 3.7: *Comparison of point number for processing for the same given input frames.*

Fig 6 shows the experiment result comparison between the standard GNG and the GNG-U2. It is obvious that the GNG-U2 provides a more stable network from a series of video frames compared with the standard GNG, the number of points have been reduced dramatically but the rough structure of the network remained the same. Next, Fig 7 shows that the number of points for processing is reduced dramatically after GNG learning. The left figure shows the point number of the original frame, whereas the right one shows the number of point for the GNG network. It is obvious that the number of points has been dramatically reduced after the GNG learning.

As mentioned earlier, our solution contains a large number of parameters that need to be optimized, and it is difficult to localize all of them in one step. To overcome this issue, I seek to optimize part of the parameters in the initial step, and make the prediction in the following steps.

To locate the human body as fast as possible, I made the restriction at the initial part, i.e., I defined a special initial posture. In this experiment, I suppose that the human skeleton as a special T-posture because it is easier to determine the length of each link of such posture. In this case, all of the parameters of the rotational angles are fixed at the initial part, and the optimized parameters are only the spatial coordinates of the skeleton. Once the original coordinates are located, the following predictions will be simpler.

The result of the human posture is shows in Fig. 8, where the top row shows the original structure and the bottom row shows the result of human posture recognition.

The computational time in PSO is affected by the number of particles and iterations. I evaluate the run time cost based on three different conditions with different iteration times. Fig 9 illustrates the time taken by different particles and iterations. It is obvious that the time

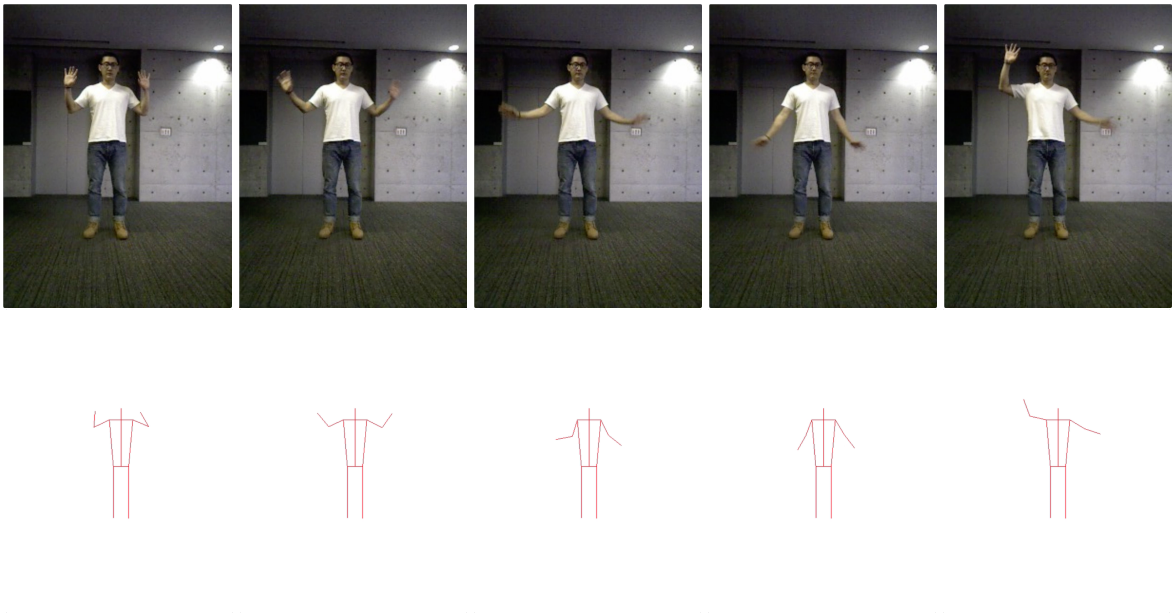


Figure 3.8: *Experiment results of human posture recognition generated by the proposed method.*

rises dramatically if the iterations increase. It is important to choose a proper iteration under different environment.

3.2.3 Discussion and Conclusion

In this section, I proposed an unsupervised human posture recognition method that is different from most of the previous proposed methods. The proposed method contains a series of unsupervised learning algorithms, such as GNG, and PSO. Thus, no pre-training data is required, which is crucial for real world applications. By applying GNG, it deduces the run time cost dramatically compared with tackling the whole point cloud directly. In addition, the PSO made it possible to find the best simulated posture without any training.

Overall, this section provides a preliminary method for human posture recognition. However, the run time cost for optimization would be a factor that limits the method for real-time implementation. In the next stage, I will focus on reducing the run time cost of optimization step.

3.3 Posture estimation by monocular camera

With the development of technology, human live a much longer life than ever before. Nevertheless, this situation leads to another situation that the percentage of elderly people in world population is getting larger and larger. According to the survey from World Health

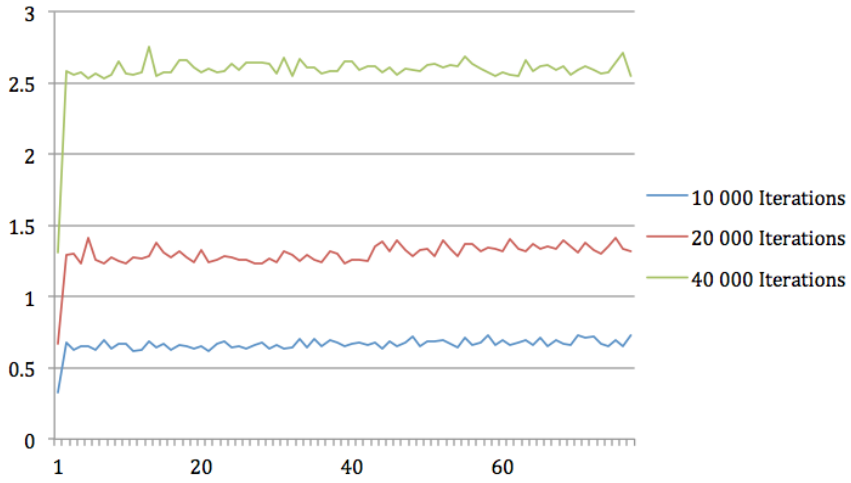


Figure 3.9: *Experiment result for different iteration times.*

Organization, the population that elder than 65 would rise to 28% at the year of 2050. On the other hand, falls are a substantial problem in individuals older than 65 years, occurring in 32% of those aged 65 to 74 years, in 35% of those aged 75 to 84 years, and in 51% of those older than 85 years each year in US[1].

Study has shown that cognitive capability changes to less with the normal aging of human, and this situation would cause much more accidents cognition-associated diseases [**?**]. On the other hand, physical exercise is considered to be one of the most effective solutions to solve this issue. Results of observational studies in [37] shows a strong relationship between physical exercise and cognitive performance, especially in elderly people. Elder who keeps taking exercises would have less possibility to get the cognitive diseases than people would is not.

Despite this truth, most of physical exercises usually require the assistance of therapists during the practising, whereas people who taking exercises at home have the less capability of evaluating their own performances correctly. And this leads to the rising requirement for the practice system which makes the elders able to practice anytime at home individually. For instance, [3, 4] introduced robot partners to help elderly people in Japan, and [5] also proposed a Socially Assistive Robot that engage, coach, assess and motivate the older adults in physical exercises in UK, and also capable of detecting anomaly activities of daily living by the assistive robot[6]. In this robot partners play the role of not only as the therapists or assistants, but also the communicator with the elder. Moreover, the robot can encourage the elderly people to engage in light to moderate physical activity.

Despite these advantages, current existing systems are usually based on special equipment such as robot partners and other motion capture devices, which is not acceptable for

most of ordinary families. To solve this issue, the system for personal use should be the better if it is constructed as simple as possible. For instance, it would be better for a personal use if the system is with a smart devices than with a complected devices.

In order to fulfill this requirement, in this paper, I proposed a monitoring and evaluating system of physical exercise. Different from other exiting systems which mainly utilize three dimensional image capture sensor such as Kinect, in this paper, I proposed a three dimensional posture recognition system which is based on 2D images. Therefore the system can be constructed by a simple low-cost devices such as smart phone, and starts in a real time.

This paper is organized as follows: section 2 introduces the related works for the referent research for this paper, whereas section 3 gives the description for our proposed frameworks. Section 4 shows the experiment environment and experiment result. In the last section, the conclusion is given and feature extension is also discussed for the further.

3.4 Related Work

As the essential part of human posture evaluating system, detection and recognition of human postures is the significant part, and it has been worked for a long time [38, 39, 40]. Consequently, various solutions have been proposed. According to [41], a taxonomy has been proposed that divided all the proposed solutions into two main groups: single-layer approaches and hierarchical approaches.

In [42], the authors proposed the supervised pose recognition method, which can be considered the most popular method today. The process described in the paper is mainly divided into two steps: first, the body part is marked from a single depth image for segmenting, and then the key nodes are marked. The body joint positioning is then performed, and the marked human body parts are remapped into the three-dimensional space to form a highly reliable spatial position for the key nodes.

The authors in [43] proposed a method for human full-body pose estimation from depth data that can be obtained using TOF cameras and the Kinect device. Their approach consists of robustly detecting anatomical landmarks in the 3D data and fitting a skeleton body model using constrained inverse kinematics. Instead of relying on appearance-based features for interest point detection, which can vary strongly with illumination and pose changes, they built a graph-based representation of the depth data to measure geodesic distances between body parts. As these distances do not change with body movement, it is able to localize anatomical landmarks independently of the pose.

Recently, in[44], the authors proposed a vector-shaped pose descriptor, which allows for the retrieval of similar poses and is easier to use with many machine learning libraries by

constructing a feature space for appearances of human poses. This method has improved the limited scope of many methods based on a kinematic or surface mesh model, and performed efficiently in experiment.

On the other hand, different from standard computation, research of soft computing research shows a strong pressure to search for new optimization techniques which are based on nature. It is often used to solve complex, multi-objective problems where the quality of a candidate solution is measured according to its performance on a large set of test cases[45, 46].

Evolutionary algorithm have been applied in various kinds of areas [47, 48, 49]. especially for robotics[50]. [51] propose a fast hybrid evolutionary approach for solving inverse kinematics for multiple end effectors simultaneously, leaving high flexibility for specifying full-body postures with different objectives. At the same time, in [52], the authors proposed fast memetic evolutionary algorithm for soling fully constrained generic inverse kinematics with multiple end effectors and goal objectives.

Despite the excellent performances of these proposed frameworks, most of them are based on the real space image capture such as Kinect, regardless of the situation of ordinary families. On the other hand, these exiting systems is always constructed with a image sensor and a connected computer, which is extremely inconvenience. Based on this situation, in this paper, I proposed a framework that is simplified.

3.5 System description

In this paper, I proposed an physical exercising evaluating system, which allow users taking physical exercises without the assistance of therapists. The structure of system is also easy to be established. The processing flow is shown in Figure 3.10.

Comparing with motion capture sensors in other exiting systems such as Kinect, in our system, I utilize normal RGB cameras. Therefore it is necessary to recognize human skeletons from RGB images. Recently, because of the development of deep learning algorithms, it is possible to recognize human pose by monocular cameras[53]. [54] proposed a robust and real-time monocular six degree of freedom relocalization convolutional neural network which is called PoseNet, and it is also suitable for detecting human body's joints for a image in real time [55].

It should be noticed that it is necessary to define the length of human's arms in the initial step of the system. In this step, users are instructed to take the T-pose in order to evaluate l_1 and l_2, which means the length of both lower and upper arms respectively. These two values would be used in the following steps.

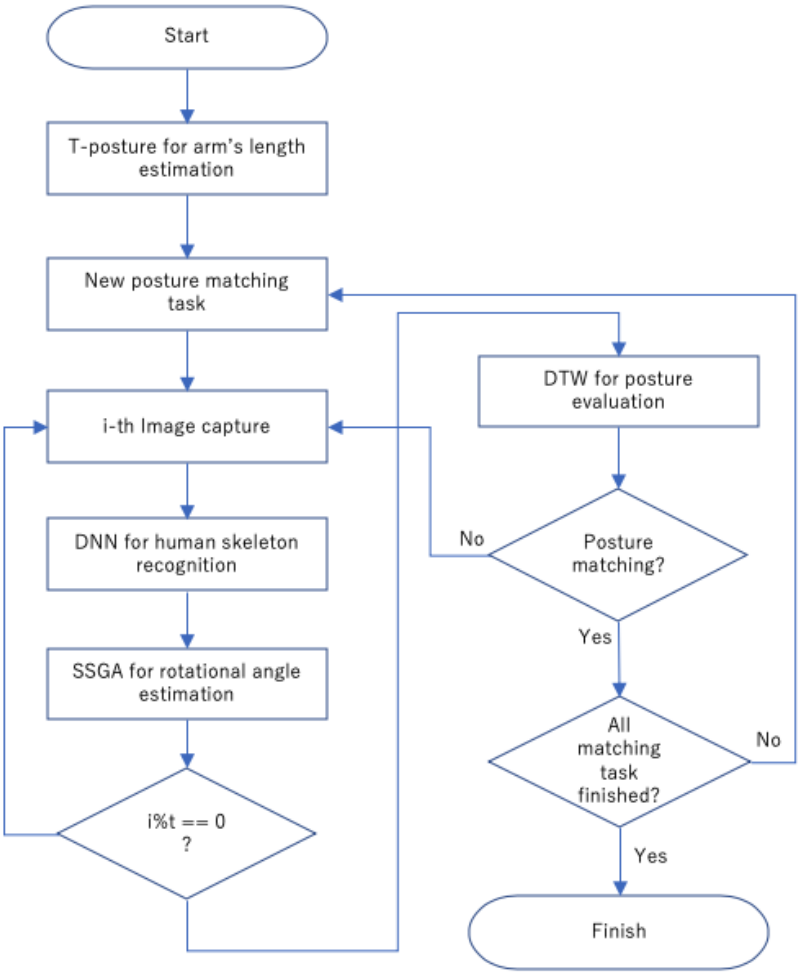


Figure 3.10: *Processing flow of proposed system.*

3.5.1 Modeling of upper limbs

Skeleton that is captured from previous section contains only image coordinates of human joints, and in order to evaluate the correctness of physical exercise, it is necessary to consider about the possible movement of upper limbs of three dimensional space in the physical exercises. Therefore I utilized a forward kinematics model of human arms, which is shown in Figure 3.11. This model includes four joint parameters, θ_1,θ_2,θ_3,θ_4, and they represent for the movement of upper rotation, forward/backward, upward/downward, rotation of elbow respectively.

Based on this, suppose that the length of arms is known and fixed, any posture for a single arm can be represented as:

$$movement = \{\theta_1, \theta_2, \theta_3, \theta_4\} \tag{3.26}$$

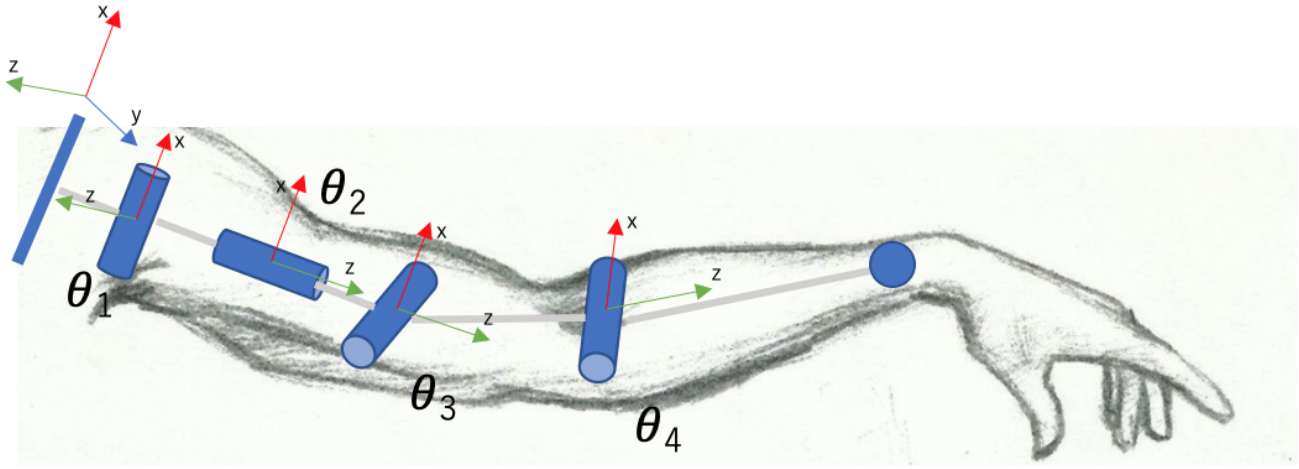


Figure 3.11: *Illustration of model for arm joints.* θ_1,θ_2,θ_3,θ_4*represents for the movement of upper rotation, forward/backward, upward/downward, rotation of elbow respectively.*

In order to confirm the movement by previous rotational angles, it is necessary to calculate the position of joints (elbow and wrist in this paper). As the common structure from [56], in this paper, I utilized Denavit-Hartenberg representation for calculating the position of elbow and wrist by modeling the forward kinematics of human upper limbs.

The Denavit-Hartenberg(DH) parameters is named after Jacques Denavit and Richard Hartenberg who introduced this representation in 1955 [57, 58].It calculates the coordinate transformation frame by frame making a list of parameters, with four parameters for each transformation:

1. Rotation angle α about X axis
2. Translation a along X axis
3. Translation d along Z axis
4. Rotation angle β along Z axis.

Because all of the coordinate systems satisfy the constraint, all of the transformation can be represented by a set of quadruple of parameters as

$$T_i = T_{\beta_j} T_{d_j} T_{a_j} T_{\alpha_j} \tag{3.27}$$

where $T_{\beta_j}, T_{d_j}, T_{a_j}, T_{\alpha_j}, (j \in m)$ represent the rotation matrix of the four steps listed above. The combined matrix T_i is

Table 3.3: *DH representation for left arm*

index	α	**a**	**d**	β
1	θ_1	0	0	0
2	0	$\theta_2 + \frac{\pi}{2}$	0	0
3	θ_3	0	l_1	0
4	θ_4	0	l_2	0

$$\begin{bmatrix} \cos\beta & -\cos\alpha\sin\beta & \sin\alpha\sin\beta & a\cos\beta \\ \sin\beta & \cos\alpha\cos\beta & -\sin\alpha\cos\beta & a\sin\beta \\ 0 & \sin\alpha & \cos\alpha & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.28}$$

Since that length of upper limbs are known and fixed, the coordinate of elbow and wrist from shoulder in real space can be calculated by DH representation as:

$$\{P_{elbow}, P_{wrist}\} = DH(\theta_1, \theta_2, \theta_3, \theta_4) \tag{3.29}$$

where P_{elbow} and P_{wrist} is constructed as:

$$P_{elbow} = \{p_1^1, p_2^1, p_3^1\} \tag{3.30}$$

$$P_{wrist} = \{p_1^2, p_2^2, p_3^2\} \tag{3.31}$$

which represent for ground truth three dimensional coordinates of elbow and wrist respectively. And the DH parameters are shown in Table 3.3

3.5.2 Recognition of joint variables

Since the position of human joints are known, the standard solution is to utilize backward kinematics to calculate the joint variables. Nevertheless, in this paper, it is possible to get two dimensional image coordinate, which is not possible to calculate the joint variables directly. Therefore in this paper, I utilized Steady-State Genetic Algorithm (SSGA) for the prediction of arm movement.

For standard genetic algorithm, the i_{th} candidate agent can be represented as:

$$g_i = \{\theta_1^i, \theta_2^i, \theta_3^i, \theta_4^i\} \tag{3.32}$$

According to the previous section, the position of elbow and wrist can be calculated according to the previous DH representations as:

$$Q(g_i) = \{Q_{elbow}^i, Q_{wrist}^i\} = DH(\theta_1^i, \theta_2^i, \theta_3^i, \theta_4^i) \tag{3.33}$$

where

$$Q_{elbow}^i = \{q_1^{i,1}, q_2^{i,1}, q_3^{i,1}\} \tag{3.34}$$

$$Q_{wrist}^i = \{q_1^{i,2}, q_2^{i,2}, q_3^{i,2}\} \tag{3.35}$$

represents for the prediction of three dimensional position of elbow and wrist respectively. Fitness value of i_{th} agent can be evaluated by fitness function:

$$f^i = \sum_{m=1}^{2} w_m \cdot \sum_{n=1}^{2} (p_n^m - q_n^{i,m})^2 \tag{3.36}$$

where w_m represents for the constant weight. It should be noticed that Q_{elbow}^i, Q_{wrist}^i only 2 defined values. It is obviously that when f^i is closed to 0, the more possibility of this agent would be.

Different from the standard GA that all the agents need to be updated, in SSGA, only the worst candidate is replaced with a candidate solution generated by the crossover and mutation. It is Elitist crossover that an individual is selected randomly and a new individual is generated by combining genetic information between the selected individual and the best one. The worst individual is updated by:

$$\theta_i^{f^{worst}} \leftarrow \theta_i^{f^c} + (\alpha * \frac{\theta_i^{f^c} - \theta_i^{f^{best}}}{\theta_i^{f^{worst}} - \theta_i^{f^{best}}} + \beta) * N(0,1) \tag{3.37}$$

Where c represents for randomly selected candidate, and $N(0,1)$ is random value of Gaussian distribution with 0 for means and 1 for variance. According to the calculation of SSGA, the best agent would be selected for representing the current arm gesture after the iterations.

3.5.3 Evaluation of physical exercise

Physical exercises requires the participators to perform by following sample postures as close as possible, therefore it is necessary to evaluate whether the participators followed correctly. In this paper, I utilized Dynamic Time Warping(DTW) to evaluate the performances.

In time series analysis, DTW is a robust algorithm for distance measure, which allowing similar shapes to match even if they are out of phase in the time axis [59].

Suppose there are two temporal sequences P_1 and P_2, which is represented as:

$$P_1 = p_1^1, p_2^1 ... p_i^1 ... p_m^1 \tag{3.38}$$

$$P_2 = p_1^2, p_2^2 ... p_j^2 ... p_n^2 \tag{3.39}$$

Where m and n are the length of P_1, P_2 respectively. DTW evaluate the similarity of two phases by calculating the nearest path. These two phases can be arranged to form a m-by-n grid , which is shown in right part Fig. 3.12 . Each grid point (i, j) corresponds to an alignment between p_i^1 and p_j^2, which can be calculated as:

$$\delta(i, j) = \|p_i^1,\ p_j^2\| \tag{3.40}$$

A warping path, W, maps or aligns the elements of S and T, which starts from lower left of the grid to the top right, and the "distance" between them is minimized.

$$W = w_1, w_2 ... w_k \tag{3.41}$$

Once a distance measure is defined, I can formally define the dynamic time warping problem a minimization over potential warping paths based on the cumulative distance for each path , which is shown in a gray path in left part of Fig. 3.12 .

The calculation of a DTW is represented as:

$$DTW(P_1, P_2) = min(\sum_{i=1}^{k} \delta(w_i)) \tag{3.42}$$

It is obvious that the when the DTW value is close to 0, these two phases would be more and more similar. As an extreme case, if the two phases are regarded as the same, the DTW value would be 0.

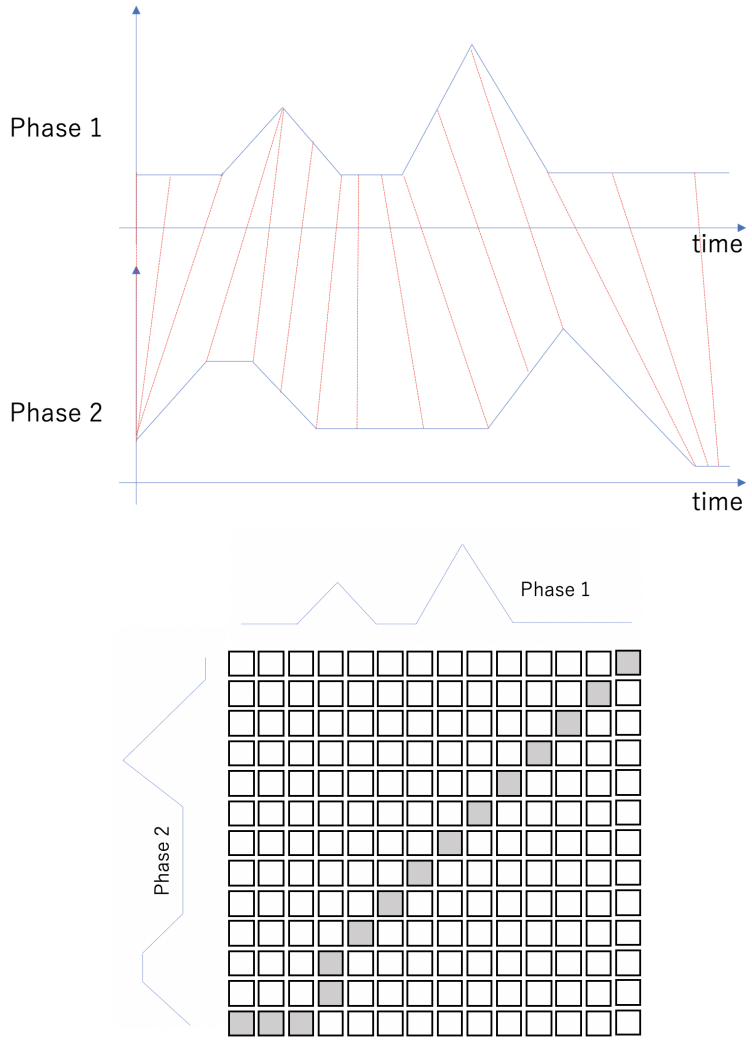


Figure 3.12: *Illustration of two different sequences and warping gird for Phase 1 and Phase 2.*

3.5.4 DTW-based SSGA for posture evaluation

Consider that in the standard SSGA, initialization for each time sequence is shown as:

$$\theta_{i,j,t+1} = \theta_{i,j,t} + \alpha(0,1) \tag{3.43}$$

where $\theta_{i,j,t+1}$ represents for the j_{th} variable of i_{th} agents at time $t+1$, and $\alpha(0,1)$ is for random variable. Considering about the task in this paper, since the user is asked make the posture be the instruction, the change of rotational variables has the large possibility of following the template. Based on this situation, I revise the initialization for each time sequence as:

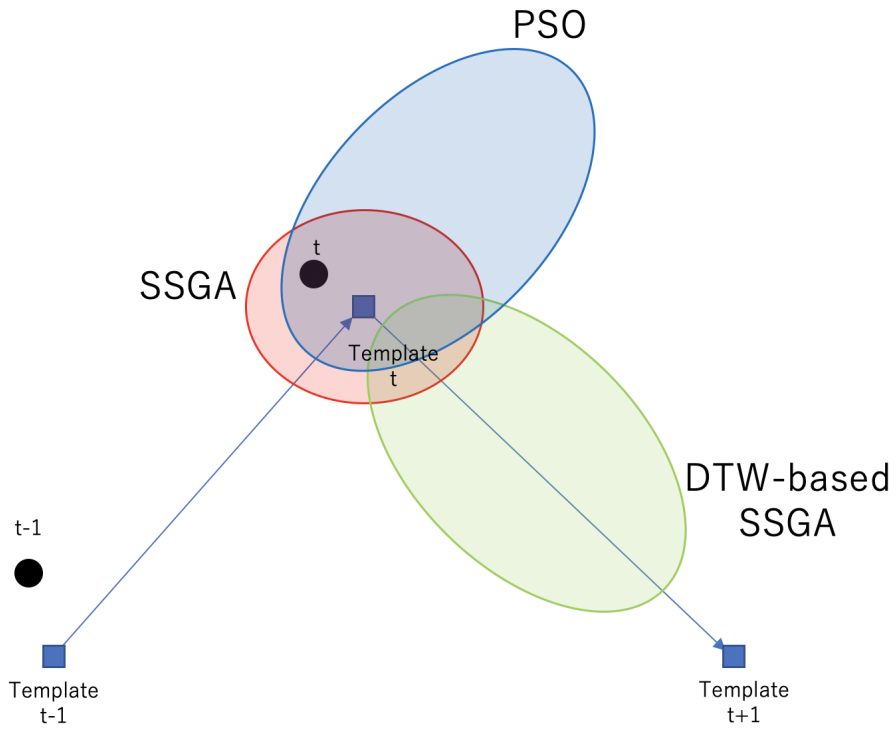


Figure 3.13: *Comparison between DTW-SSGA and other similar algorithms. The blue rectangles represent the template time series, where as black dot is for the predicted result. Red, green and blue ellipses represent for the initial range for SSGA, DTW-based SSGA and PSO respectively. It is obviously that the range of DTW-based SSGA would the most possible range for the time t+1.*

$$\theta_{i,j,t+1} = \theta_{i,j,t} + \alpha(0,1) + \beta_j(x_{i,n,t+1}^{Template} - x_{i,n,t}^{Template}) \tag{3.44}$$

where $x_{i,n,t+1}^{Template}$ and $x_{i,n,t}^{Template}$ represents for the possible adjacent value ranges of template corresponding to the $\theta_{i,j,t+1}$. By utilizing DTW-SSGA, initialization for each frame would be generated by the instruction of posture template. The illustration of DTW-SSGA is shown in Figure 3.13. When handling new frame, DTW-SSGA give the direction according to the instruction of DTW template. However, standard SSGA only initial around the previous best position. Moreover, other evolutionary algorithms such as Particle Swarm Optimization [60] initial the agents according to the direction of previous calculation. The description of DTW-based SSGA is shown in Algorithm. 1.

Algorithm 1 Dynamic Time Warping based Steady State Genetic Algorithm for evaluation of joint variables.

Input:

$\theta_t^{i,n}$: Joint variable agent, $1 < i < 4, n \in \{1, 2, \ldots N\}$

$x_{i,m}^{Template}$: DTW template value, $1 < i < 4, m \in \{1, 2, \ldots M\}$

t : time frame, $t \in \{1, 2, \ldots T\}$

START

for $t \leftarrow 1$ to T **do**

for $j \leftarrow 1$ to N **do**

for $i \leftarrow 1$ to 4 **do**

$$\theta_{i,j,t} = \theta_{i,j,t-1} + \alpha(0,1) + \beta_j(x_{i,m}^{Template} - x_{i,m-1}^{Template})$$

end for

end for

for $iteration \leftarrow 1$ to max iteration **do**

$\theta_{worst} \leftarrow \arg\max_m f(\theta_t^m)$

$\theta_{best} \leftarrow \arg\min_m f(\theta_t^m)$

$\theta_c \leftarrow Random(G)$

$\theta_{i,t}^{f^{worst}} \leftarrow \theta_{i,t}^{f^c} + (\alpha * \frac{\theta_{i,t}^{f^c} - \theta_{i,t}^{f^{best}}}{\theta_{i,t}^{f^{worst}} - \theta_{i,t}^{f^{best}}} + \beta) * N(0,1)$

Calculate fitness $f(\theta_{i,t}^{f^{worst}})$

end for

end for

Table 3.4: *Rotational range of joint variables for left and right arm.*

Left arm			Right arm		
variable	**min**	**max**	**variable**	**min**	**max**
θ_1	$-\frac{\pi}{2}$	$\frac{\pi}{2}$	θ_1	$-\frac{\pi}{2}$	$\frac{\pi}{2}$
θ_2	$-\frac{\pi}{2}$	$\frac{\pi}{2}$	θ_2	$-\frac{\pi}{2}$	$\frac{\pi}{2}$
θ_3	0	$\frac{3\pi}{4}$	θ_3	$-\frac{3\pi}{4}$	0
θ_4	0	$\frac{3\pi}{4}$	θ_4	$-\frac{3\pi}{4}$	0

3.6 Experimental result

In order to prove the performance of our proposed framework, I made the following experiment. In the first step, considering the possible moving range for human arms are limited, I set up the range of four rotational angles after experiment, and the range of each angles for two arms are shown in Table. 3.4.

In order to evaluate that whether the same postures have the similar change rate of rotational angles, I simulated several dynamic postures to test the performance artificially. Pose 1 represents for the T-posture with the rotation of upper arms, whereas Pose 2 is for moving arms straight forward. Pose 3 is for waving arms. Illustration of these poses is shown in Figure. 3.14, and the changing of joint variables is shown in the left column of Figure 3.15.

I first tested the performance of standard SSGA. Considering about the balance between run time cost and accuracy, in this experiment, I use 300 agents and 300 iterations for each step, and the predicting result for rotational variables θ_1, θ_2, θ_3 and θ_4 are shown in the bottom row of Figure 3.15. It should be noticed that not all of these four rotations play significant role for any postures. On the other hand, they show the limit effect for some special postures. For instances, in Pose3, rotational angle θ_2 played limited roles in this dynamic pose, which means that any value of rotation angle of θ_2 does not change the performance for Pose2. Therefore, the prediction for θ_2 is not similar with ground truth.

For DTW calculation, I predict θ_1, θ_2, θ_3 and θ_4 10 for each template poses that are explained above, and calculated the mean average angle values. The result is shown in Table. 3.5. To test the differences between each pair of postures, I evaluate the result of each prediction with each template respectively, and the result is shown as Table 3.6. It is obviously that each poses get the smallest DTW value when pairing with their referent templates.

However, it should be noticed that the performance above calculating the posture step by step, and the posture is simple. Considering about calculating time,it is necessary to sampling with a fixed time interval. In this case, the performance of standard SSGA and DTW-based

(a) *Pose 1*

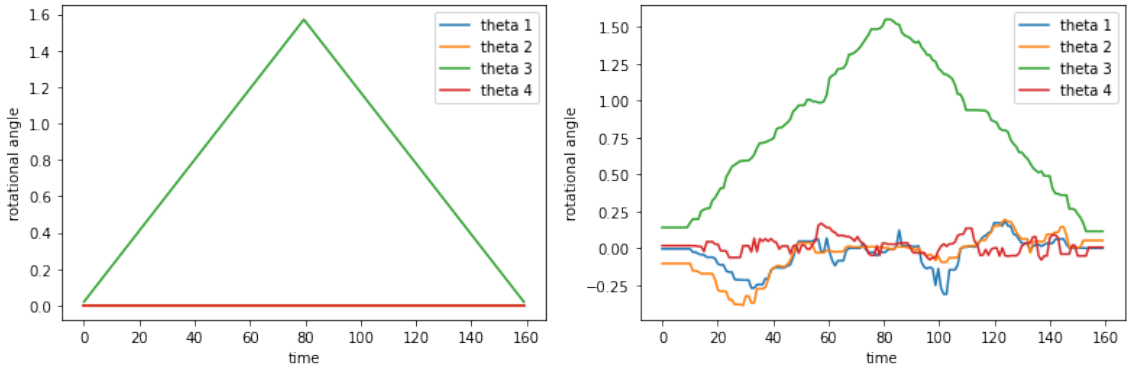
(b) *Pose 2*

(c) *Pose 3*

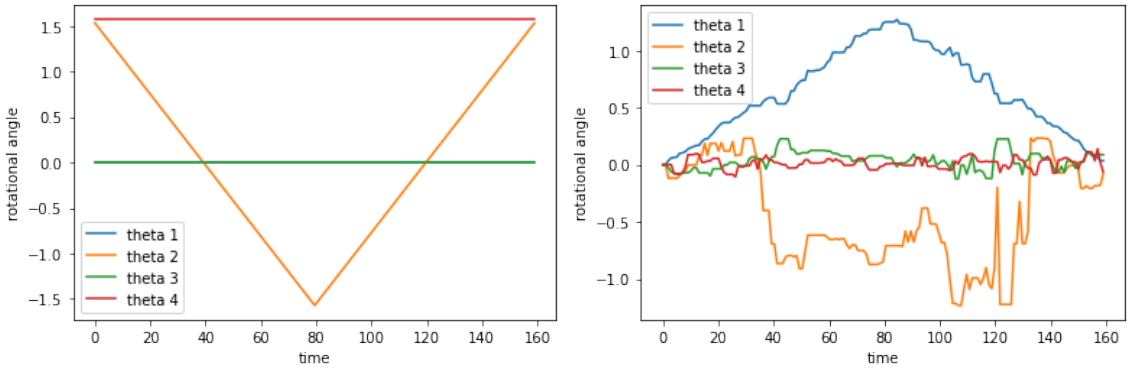
Figure 3.14: *Illustration of three template poses.*

Table 3.5: *Average DTW values of 10 times for 5 poses.*

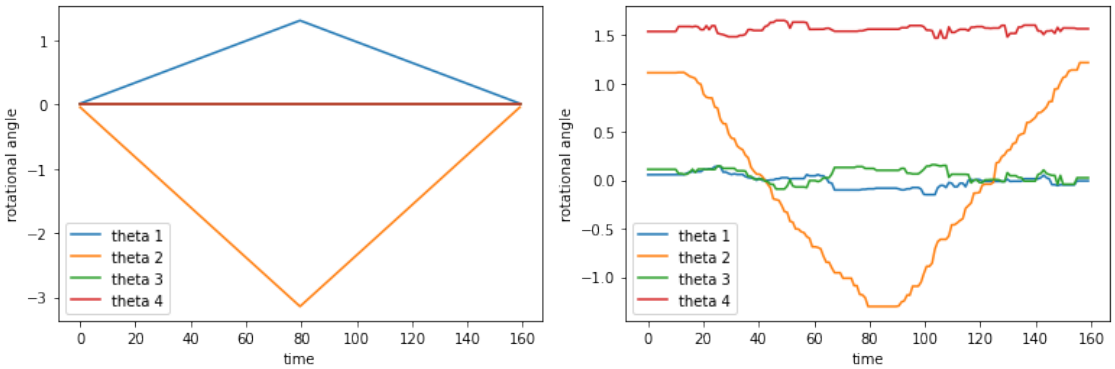
Pose	**Pose 1**	**Pose 2**	**Pose 3**
DTW	22.15 ± 11.99	20.28 ± 12.40	12.96 ± 3.48



(a) *Pose 1*



(b) *Pose 2*



(c) *Pose 3*

Figure 3.15: *Experiment result of several poses between simulation data and prediction result.*

Table 3.6: *DTW comparison between each poses.*

Prediction \ Template	Pose 1	Pose 2	Pose 3
Pose 1	21.20	205.36	106.38
Pose 2	198.79	15.73	179.14
Pose 3	95.06	180.64	14.28

SSGA is comparison in figure 3.16. It is obviously that the DTW-based SSGA performed much better than the former one.

Considering about the computation capability and cost, in this experiment I utilized the iPod touch 7th as the main device, The appearance is shown in and the technical specifications can be seen in Figure 3.17. It is capable of performing the whole system, including PoseNet deep neural network module with a FPS of 3 to 5.

The PoseNet is capably of recognizing human poses by detecting 17 key points of human joints, which is shown in the Figure 3.18 (a), and the performance is shown as Figure 3.18 (b). In this paper, I only test the exercises of upper body, therefore only six joints are under consideration, i.e. left shoulder, left elbow, left wrist, right shoulder, right elbow, right wrist.

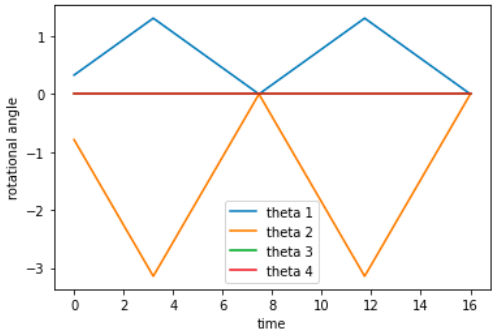
I made the volunteer to perform the same pose as the template, and the process of pose 2 is shown in Figure. 3.19. Comparing with the artificial template, there are several differences between it and real pose. And the differences of trajectory for the movement of arms can be seen in Figure. 3.20. In the comparison, the differences is mainly caused by several reasons: first is that it is because the volunteer is generally not possible to be as formal as requested; there are exits errors between real joint positions and estimated position by PoseNet.

The result of estimation is shown in Figure 3.21. It is obviously that the result shows the differences with the template of pose2 because of the reasons that are mentioned above. Nevertheless, the essential change of pose 2, i.e. the change of θ_2 predicted the same trajectory with the template values.

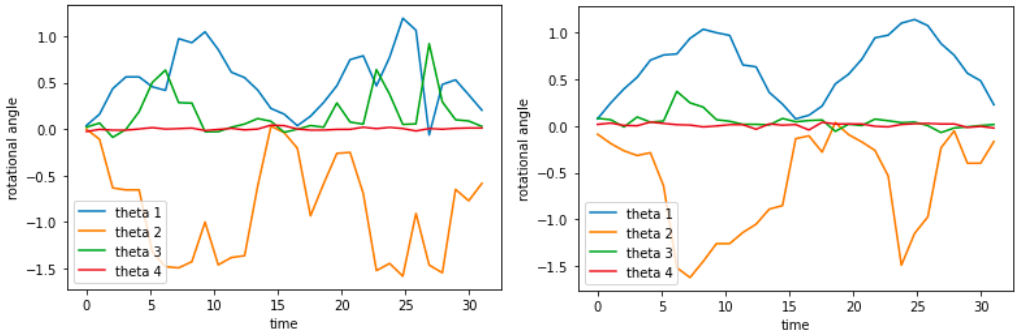
3.7 Summary

The strong relationship between physical exercise and cognitive performance especially in elderly people is shown in the current society, and physical exercise system is also getting more and more concentration.

Based on this situation, in this chapter, I proposed a dynamic posture evaluating and



(a) *Template*



(b) *SSGA* **(c)** *DTW-based SSGA*

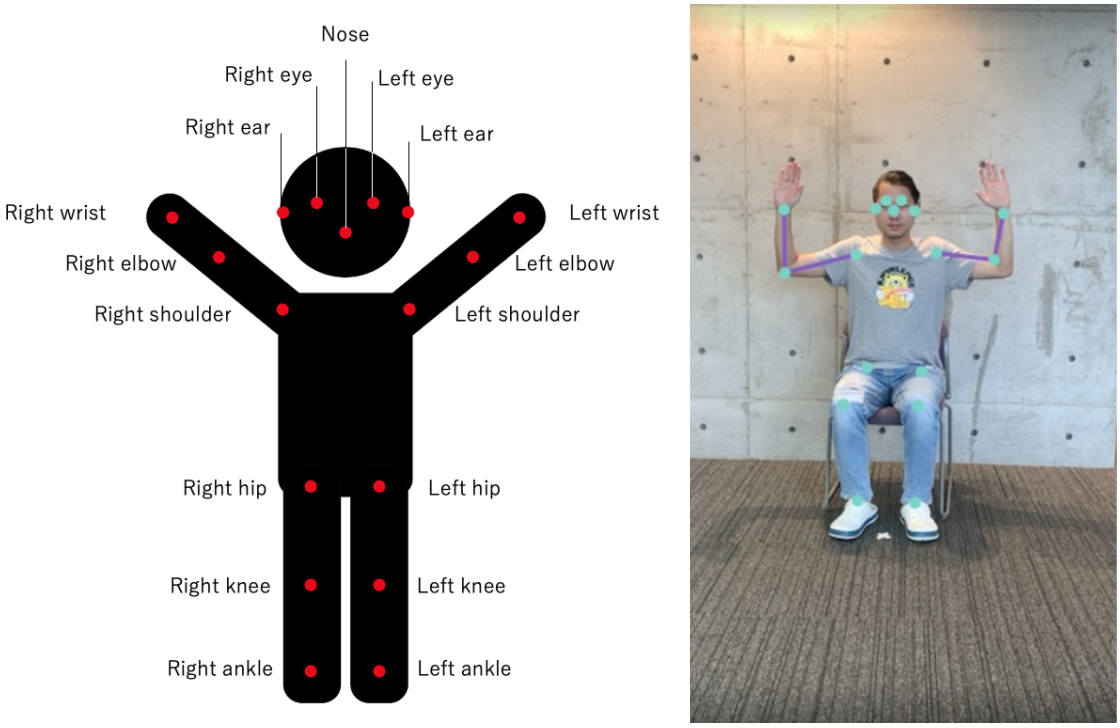
Figure 3.16: *Experiment result of several poses between standard SSGA and DTW-based SSGA.*

Feature	Detail
Released Date	May 28, 2019
Weight	88g
Processors	1(2 cores)
Processor Type	Apple A10 Fusion
Processor Speed	1.6GHz
Onboard RAM	2GB
Camera	8 Megapixels
Display resolution	1136*640

Figure 3.17: *Appearance and feature of iPod touch.*

matching framework which is constructed by DTW-based Steady State Genetic Algorithm based forward kinematics for rotational angle prediction, and Dynamic Time Warping for dynamic posture matching. Comparing with standard SSGA, DTW-based SSGA perform much better when handling the complected postures or longer time interval.

This make it possible that the system can be constructed by a simple device such smart phone and other mobile smart devices. In this paper, I only consider the evaluation of arms for exercise. In the future, I will also take into consideration of evaluation of legs, head and



(a) *17 Key points*

(b) *Performance when human is sitting.*

Figure 3.18: *Key points of PoseNet recognize.*

Figure 3.19: *Captured joints by PoseNet for template pose2.*

torso.

Despite these advantages, there are still some issues that I need to solve. First of all, the accuracy of the prediction is not as high as systems that calculate from three dimensional input data directly. In the future, I would focus on this issue by selecting more suitable algorithms if possible.

Another issue is that there is still some problems for estimating the real data because of the deviation between real values and estimated values. In the future I would also intend to

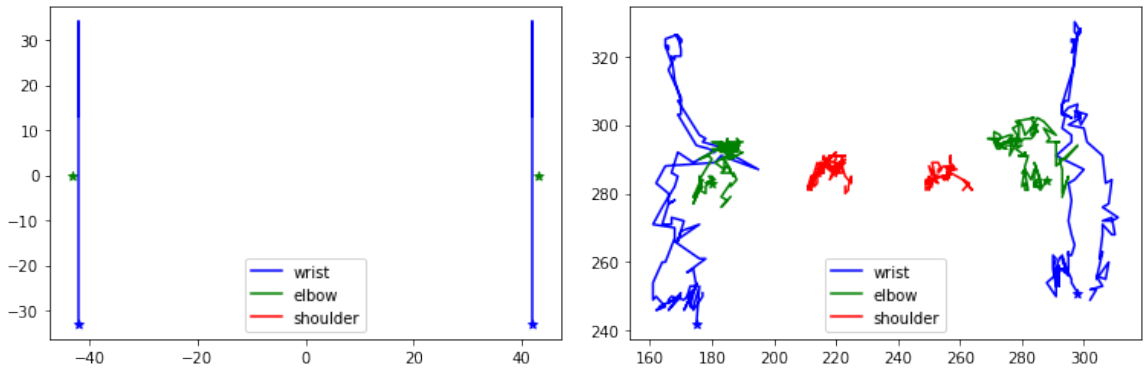


Figure 3.20: *Comparison of Trajectory of shoulders, elbows and wrist between simulated values and real values captured by camera.*

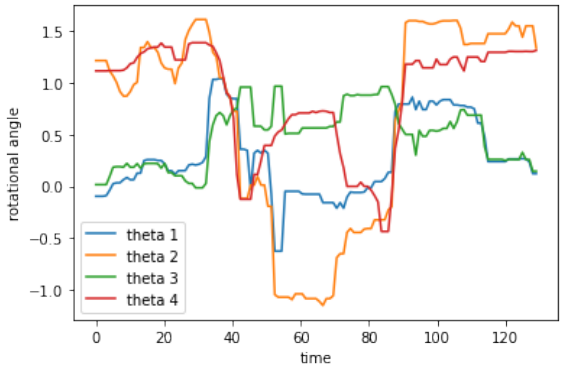


Figure 3.21: *Estimated joint variables of Pose 2.*

deal with the problems.

Chapter 4

Multi-view Evolutionary Robot Vision for Human Motion Estimation

4.1 Introduction

With the development of technology, humans live a much longer life than ever before. Nevertheless, this situation leads to another situation that the percentage of elderly people in the world population is getting larger and larger. Studies have shown that cognitive capability changes to less with the normal aging of humanity, and this situation would cause many more accidents and cognition-associated diseases [2]. On the other hand, physical exercise is considered to be one of the most effective solutions to solve this issue. The results of observational studies in [61] proved a strong relationship between physical exercise and cognitive capability, especially in elderly people. Elders who keep exercising would have less possibility to get cognitive diseases than people would are not.

Based on this truth, most physical exercises usually require the assistance of therapists during the practicing, whereas people who taking exercises at home have less capability of evaluating their performances correctly. And this leads to the rising requirement for the practice monitoring system which makes the elders able to practice anytime at home individually.

As the key part, the pose recognition system plays the core important role in these practice monitoring systems. Different from other fields, for the physical exercise system, therapists would evaluate human's states, according to the movement of his her arms or legs. On the other hand, most of referent researches focus on evaluating human's posture with high accuracy, rather than the convenience and affordable for ordinary users. In order to obtain result as accurate as possible, which either using specific cameras such as depth cameras, and the computation are always required to process on high-quality computation units.

For making this process more convenience and easy implementation, in this paper, I proposed a framework that applying evolutionary algorithm for optimizing three-dimensional human pose from the main view, which provides 2D image coordinates of human joints. This system is light-weighted and be implemented on low price smart devices, which is convenience for ordinary people to use at home. To reduce of error that is caused by fault human joints recognition, one more image view for fundamental matrix estimation is introduced for correction. Comparing with other previous researches, the proposed method is light weighted and can be implemented in a simple smart devices. On the other hand, there is no necessary for calibrating for acquiring camera parameters, which usually is necessary for most of referent researches.

The rest of this paper is organized as follows: Section 2 introduces the related works about human pose estimation, whereas Section 3 explains the detail of our proposed framework for estimating the 3D human poses. To solve the problem of error that is caused while the acquiring of 2D human pose, I also explain our strategy in this section. In Section 4, the experimental result is given for proving our method, and in the last section, the conclusion and future extension would be stated.

4.2 Related Work

Vision-based human pose estimation has become one of the most popular research topics since decades ago [62][63]. Methods of pose estimation can be roughly divided into two categories: model-based and model-free, which depends on whether prior knowledge is used[64].

In the early ages, researchers focused on recognizing 2D human poses because of the limitation of the hardware. In [65], human pose estimation is formulated as a jigsaw puzzle problem in which the body part tiles maximally cover the foreground region, match local image features, and satisfy body plan and color constraints. However, 2D human pose estimation sometimes is not sufficient because of the loss of spatial information.

With the improvement of devices such as the appearance of the depth camera, it has become possible to capture three-dimensional coordinates by the devices, therefore the 3D human pose estimation has also becoming possible[66]. In [67], the authors a method to quickly and accurately predict 3D positions of body joints from a single depth image by utilizing random forest algorithms. This system runs at as high as 200 frames per second on consumer hardware and shows high accuracy on both synthetic and real test sets.

Instead of acquiring three dimensional data directly, some researches estimating 3D human poses by utilizing multiple views. In [68], the authors estimated 3D pose estimation of

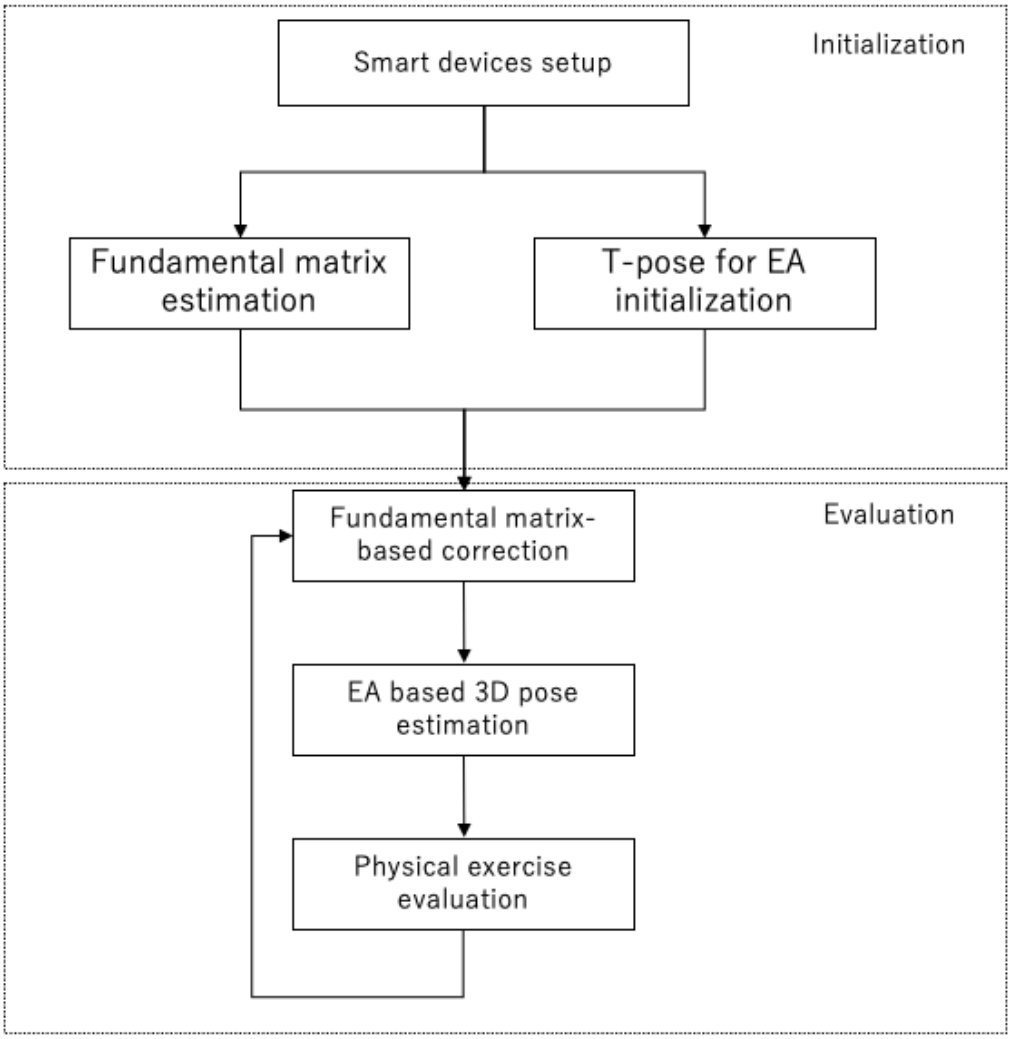


Figure 4.1: *Processing flow of proposed method*

multiple humans from multiple calibrated cameras. They first create a reduced state space by triangulation of corresponding body joints obtained from part detectors in pairs of camera views and then introduced a novel 3D pictorial structures (3DPS) model for inferring 3D human body configurations from the reduced state space.

4.3 Human pose estimation

Despite the excellent performances of the previous methods that are mentioned above, there are many problems that still exist, especially restrictions and the computation cost for estimation. Either specific devices such as depth cameras or more than three cameras are necessary. From conquering these restrictions, in this section, I explain our proposed method,

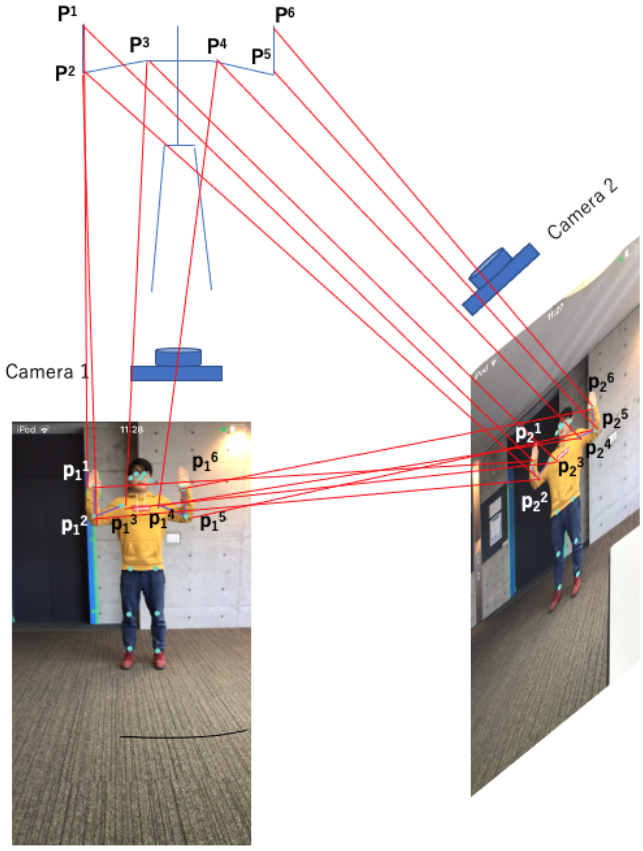


Figure 4.2: *Illustration of the setup for the experiment.*

which can be implemented in at most two smart devices. In the first step, length of upper & lower arms are initialized, which is captured by camera of main view, fundamental matrix between main camera view and sub camera view would be estimated by the captured joints coordinates, which is used for correcting fault human joints' coordinates. The process flow of our proposed method is shown in Fig. 4.1.

4.3.1 Acquiring of 2D human joint position

The first step for our proposed method is to acquire 2D human skeletons that are captured by smart devices. Even though there are many restrictions about 3D human pose recognition, 2D human pose estimation by deep neural network still keeps a high accuracy [69]. [70] proposed a robust and real-time monocular six degree of freedom relocalization convolutional neural network which is called PoseNet, and it is also capable of detecting the human body's joints in real-time [71] with low price devices.

I set up two smart deices, which is shown in Fig. 4.2. Each joint in 3D space $\mathbf{P^i}$ will be projected into two images $\mathbf{p_1^i}$ and $\mathbf{p_2^i}$ which refer to two image coordinates.

It should be noticed that it is necessary to define the length of a human's arms in the initial step of the system. In this step, users are instructed to take the T-pose in order to confirm l_1 and l_2 in the beginning, which represent the length of both lower and upper arms respectively. These two values would be used in the following steps.

4.3.2 Human joints correction with fundamental matrix

Despite the performance of the 2D skeleton joints estimation, there is still the information loss caused by projections from 3D to 2D. On the other hand, some joints are not detectable because of the accuracy. This could lead to a large error of prediction. To solve this problem, in this section, I explain our strategy by introducing one more view for improving the accuracy and reducing the instability at the same time.

As shown in Fig. 4.2 before, I denote $\mathbf{p_1}$ and $\mathbf{p_2}$ as the corresponding points that are the projection point of the arbitrary three dimensional joint P. Therefore the following equations are satisfied:

$$w_1\mathbf{p_1} = w_1 \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} = A[\mathbf{R_1}|\mathbf{t_1}]P \tag{4.1}$$

$$w_2\mathbf{p_2} = w_2 \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix} = A[\mathbf{R_2}|\mathbf{t_2}]P \tag{4.2}$$

Where $\mathbf{R}$ and $\mathbf{t}$ represent for the rotational matrix and transform vector respectively, whereas A is for projection matrix. The relationship between $\mathbf{P}$ and $\mathbf{p_1}$, $\mathbf{p_2}$ can be calculated as:

$$\mathbf{p_1}^T\mathbf{F}\mathbf{p_2} = 0 \tag{4.3}$$

Where $\mathbf{F}$ is the so-called fundamental matrix which includes 9 elements.

Consider the number of parameters, at least 8 corresponding points are required for estimating the previous formula[72]. Once the $\mathbf{F}$ between two camera views has been confirmed, the epipolar line in one view would be calculated by the known corresponding point in another view:

$$l_1 = \mathbf{F}^T\mathbf{p_2} \tag{4.4}$$

$$l_2 = \mathbf{F}\mathbf{p_1}^T \tag{4.5}$$

According to the feature of epipolar geometry, the corresponding point of p_l should lie on the epipolar line l_2, i.e.:

$$\|l_1,\ \ l_2\|^2 = \|\mathbf{F}^T\mathbf{p}_2,\ \ \mathbf{F}\mathbf{p_1}^T\|^2 = 0 \tag{4.6}$$

For handling the error, for a captured time series data of human pose, suppose that at time t, the image position of the joint at the main view p_1, the value would be set as:

$$\hat{\mathbf{p}}_\mathbf{1}^\mathbf{t} = \{ \begin{array}{cc} \mathbf{p}_\mathbf{1}^\mathbf{t} & if\ d_1\ or\ \ d_2 < \tau \\ \hat{\mathbf{p}}_\mathbf{1}^\mathbf{t-1} + \mathbf{v}_\mathbf{1}^\mathbf{t} & otherwise \end{array} \tag{4.7}$$

Where d_1 and d_2 represent for $\|\mathbf{F}^T\mathbf{p}_2, \mathbf{F}\mathbf{p_1}^T\|^2$ and $\|\mathbf{F}^T\hat{\mathbf{p}}_2, \mathbf{F}\mathbf{p_1}^T\|^2$ respectively. And for point in another view $\hat{p}_1^t$, it is updated as

$$\hat{\mathbf{p}}_\mathbf{2}^\mathbf{t} = \{ \begin{array}{cc} \mathbf{p}_\mathbf{2}^\mathbf{t} & if\ d_3\ or\ \ d_4 < \tau \\ \hat{\mathbf{p}}_\mathbf{2}^\mathbf{t-1} + \mathbf{v}_\mathbf{2}^\mathbf{t} & otherwise \end{array} \tag{4.8}$$

where d_3 and d_4 represent for $\|\mathbf{F}^T\mathbf{p}_2, \mathbf{F}\mathbf{p_1}^T\|^2$ and $\|\mathbf{F}^T\hat{\mathbf{p}}_2, \mathbf{F}(\mathbf{p_1} + \mathbf{v}_\mathbf{1}^\mathbf{t})^T\|^2$ respectively, and τ is the threshold of estimated Sampson distance, and $\mathbf{v}_\mathbf{1}^\mathbf{t}$ $\mathbf{v}_\mathbf{2}^\mathbf{t}$ are defined as:

$$\mathbf{v}_\mathbf{1}^\mathbf{t} = \hat{\mathbf{p}}_\mathbf{1}^\mathbf{t-1} - \hat{\mathbf{p}}_\mathbf{1}^\mathbf{t-2} \tag{4.9}$$

$$\mathbf{v}_\mathbf{2}^\mathbf{t} = \hat{\mathbf{p}}_\mathbf{2}^\mathbf{t-1} - \hat{\mathbf{p}}_\mathbf{2}^\mathbf{t-2} \tag{4.10}$$

4.3.3 Modeling of upper limbs

To explain our method clearly, I explain upper limbs as an example instead of the whole skeleton in the following of this paper.

As stated previously that I suppose the length of arms l_1 and l_2 are known and fixed, any posture for a single arm can be represented as:

$$g^* = \{\theta_1^*, \theta_2^*, \theta_3^*, \theta_4^*\} \tag{4.11}$$

Which is shown in Fig. 3.11 of previous chapter. In order to confirm the movement by previous rotational angles, it is necessary to calculate the position of joints (elbow and wrist in this paper). As the common structure from [73], in this paper, I utilized Denavit-

Hartenberg representation for calculating the position of elbow and wrist by modeling the forward kinematics of human upper limbs.

The Denavit-Hartenberg(DH) parameters are named after Jacques Denavit and Richard Hartenberg who introduced this representation in 1955 [**?**, 74]. It calculates the coordinate transformation frame by frame making a list of parameters, with four parameters for each transformation:

1. Rotation angle α about X axis
2. Translation a along X axis
3. Translation d along Z axis
4. Rotation angle β along Z axis.

Since that, all of the coordinate systems satisfy the constraint, all of the transformation can be represented by a set of quadruple parameters as

$$T_j = T_{\beta_j} T_{d_j} T_{a_j} T_{\alpha_j} \tag{4.12}$$

where $T_{\beta_j}, T_{d_j}, T_{a_j}, T_{\alpha_j}, (j \in m)$ represent the rotation matrix of the four steps listed above.

Since the length of upper limbs is known and fixed, the coordinate of elbow and wrist from the shoulder in real space can be estimated by DH representation as:

$$[P_{elbow}, P_{wrist}] = DH(g^*) \tag{4.13}$$

where P_{elbow} and P_{wrist} is constructed as:

$$P_{elbow} = [p_1^1, p_2^1, p_3^1]^T \tag{4.14}$$

$$P_{wrist} = [p_1^2, p_2^2, p_3^2]^T \tag{4.15}$$

which represent for ground truth three-dimensional coordinates of elbow and wrist respectively. And the DH parameters are shown in Table 4.1

4.3.4 Estimation of rotational joint variables

Since that the position of human joints is known, the typical solutions usually utilize backward kinematics to calculate the joint variables. However, in this paper, it is possible to get a two-dimensional image coordinate, which is not possible to calculate the joint variables directly. Therefore in this paper, I utilized the Steady-State Genetic Algorithm (SSGA),

Table 4.1: *DH representation for left arm*

index	α	**a**	**d**	β
1	θ_1	0	0	0
2	0	$\theta_2 + \frac{\pi}{2}$	0	0
3	θ_3	0	l_1	0
4	θ_4	0	l_2	0

which is one kind of genetic algorithm (GA) for the prediction of arm movement.

For genetic algorithm, the i_{th} candidate agent can be represented as:

$$g_i = (\theta_1^i, \theta_2^i, \theta_3^i, \theta_4^i) \tag{4.16}$$

According to the previous section, the position of elbow and wrist can be calculated according to the previous DH representations as:

$$[Q_{elbow}^i, Q_{wrist}^i] = DH(g^i) \tag{4.17}$$

where

$$Q_{elbow}^i = [q_1^{i,1}, q_2^{i,1}, q_3^{i,1}]^T \tag{4.18}$$

$$Q_{wrist}^i = [q_1^{i,2}, q_2^{i,2}, q_3^{i,2}]^T \tag{4.19}$$

represents for the prediction of the three-dimensional position of the elbow and wrist respectively.

Consider the complexity of the calculation and errors that can be ignored, I assume the projection matrix as:

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{4.20}$$

Therefore fitness value of i_{th} agent can be evaluated by fitness function:

$$\begin{aligned} f(g_i) &= \textstyle\sum_{m=1}^{2} w_m \cdot \|AP_m - AQ_m^i\|^2 \\ &= \textstyle\sum_{m=1}^{2} w_m \cdot \sum_{n=1}^{2} \|p_n^m - q_n^{i,m}\|^2 \end{aligned} \tag{4.21}$$

where w_m represents the weight for different joints. It should be noticed that Q_{elbow}^i, Q_{wrist}^i

only 2 defined values. It is obvious that when f^i is closed to 0, the more possibility of this agent would be.

Algorithm 2 Steady State Genetic Algorithm

```
initialize population
while termination criteria is reached do
    select best and worst agent
    randomly select agent
    worst agent crossover with probability
    worst agent mutation with probability
    fitness calculation
end while
```

Different from the standard GA that all the agents need to be updated, in SSGA, only the worst candidate is replaced with a candidate solution generated by the crossover and mutation, and the process is shown in Algorithm 2. It is an elitist crossover that an individual is selected randomly and a new individual is generated by combining genetic information between the selected individual and the best one. The worst individual is updated by:

$$\theta_i^{f^{worst}} \leftarrow \theta_i^{f^c} + (\alpha \cdot \frac{\theta_i^{f^c} - \theta_i^{f^{best}}}{\theta_i^{f^{worst}} - \theta_i^{f^{best}}} + \beta) \cdot N(0,1) \tag{4.22}$$

Where c represents the randomly selected candidate and $N(0,1)$ is the random value of Gaussian distribution with 0 for means and 1 for variance. According to the calculation of SSGA, the best agent would be selected for representing the current arm gesture after the iterations.

4.4 Experiment result

I first consider the possible moving range for human arms are limited, I set up the range of rotational angles for two arms as shown in Table. 4.2.

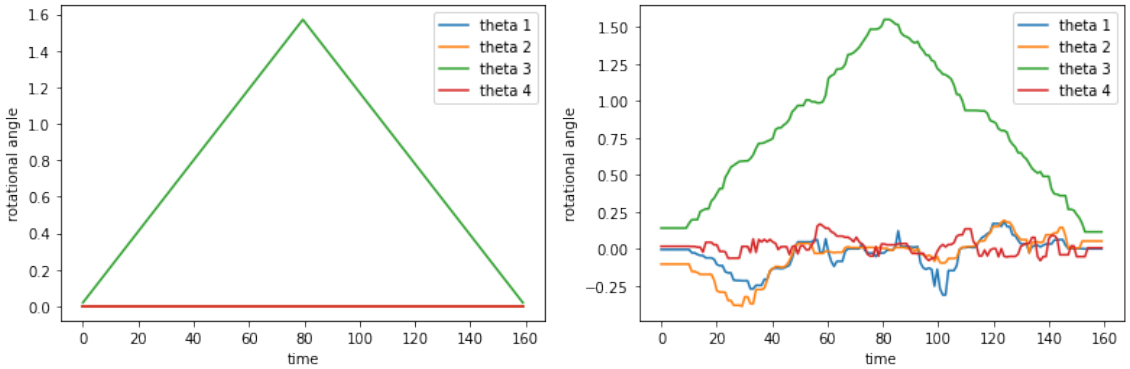
To evaluate that whether the same postures have a similar change rate of rotational angles, I manufacture several dynamic postures to test the performance artificially, which the intrinsic and extrinsic matrix as:

$$in = \begin{bmatrix} 7.0709e^2 & 0 & 6.0188e^2 \\ 0 & 7.0709e^2 & 6.0188e^2 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.23}$$

Table 4.2: *Rotational range of joint variables for left and right arm.*

Left arm			Right arm		
variable	**min**	**max**	**variable**	**min**	**max**
θ_1	$-\frac{\pi}{2}$	$\frac{\pi}{2}$	θ_1	$-\frac{\pi}{2}$	$\frac{\pi}{2}$
θ_2	$-\frac{\pi}{2}$	$\frac{\pi}{2}$	θ_2	$-\frac{\pi}{2}$	$\frac{\pi}{2}$
θ_3	0	$\frac{3\pi}{4}$	θ_3	$-\frac{3\pi}{4}$	0
θ_4	0	$\frac{3\pi}{4}$	θ_4	$-\frac{3\pi}{4}$	0

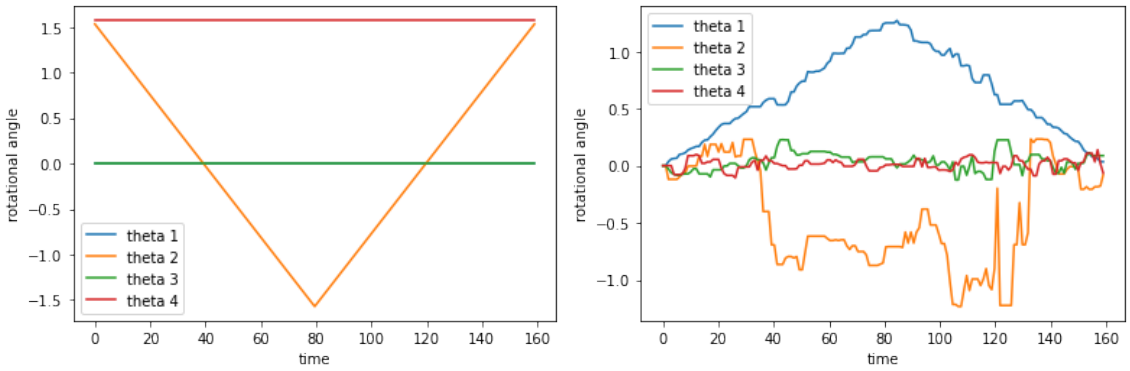
(a) *Pose 1*



(b) *Pose 1 parameters ground truth and estimated result*

Figure 4.3: *Experiment result of several poses between manufactured joint rotational angle and prediction result for pose 1.*

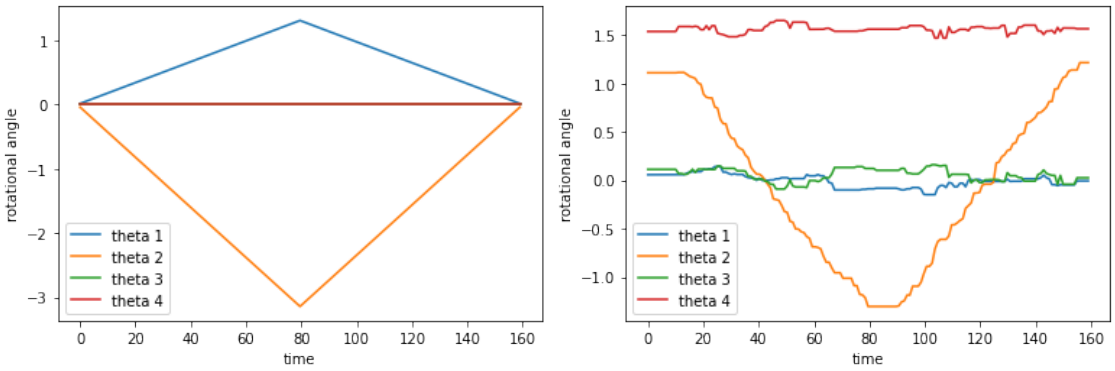
(a) *Pose 2*



(b) *Pose 2 parameters ground truth and estimated result*

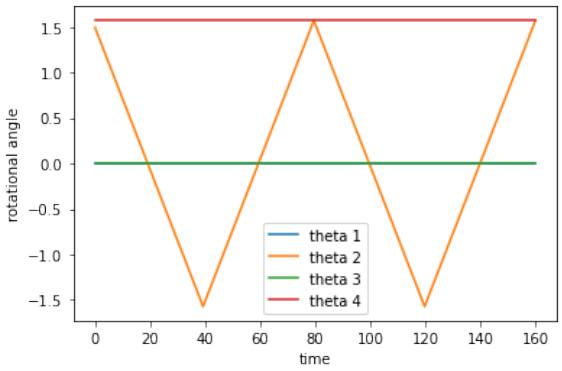
Figure 4.4: *Experiment result of several poses between manufactured joint rotational angle and prediction result for pose 2.*

(a) *Pose 3*

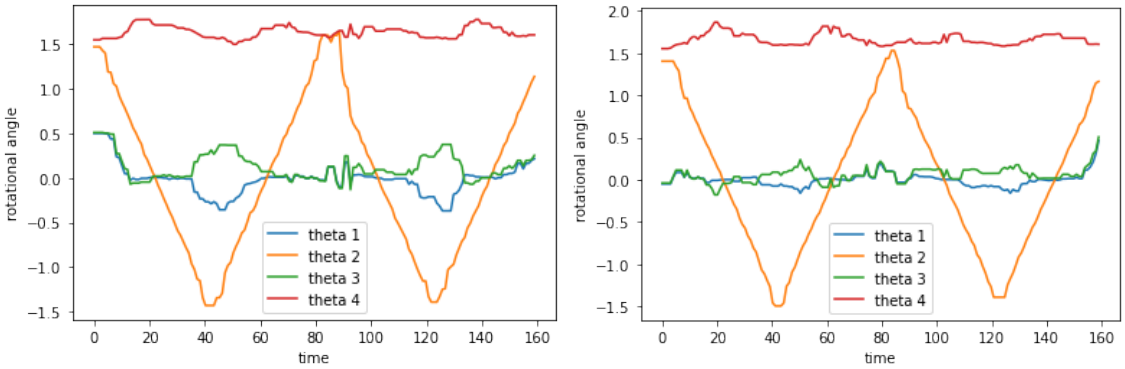


(b) *Pose 3 parameters ground truth and estimated result*

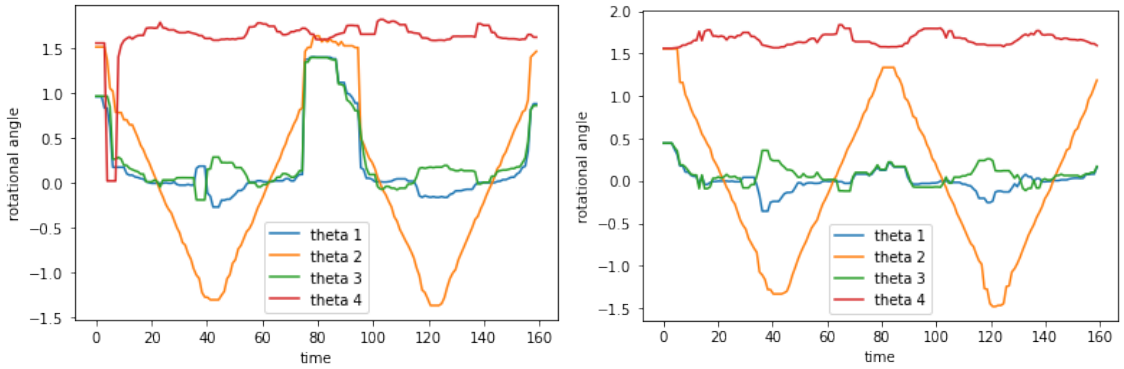
Figure 4.5: *Experiment result of several poses between manufactured joint rotational angle and prediction result for pose 3.*



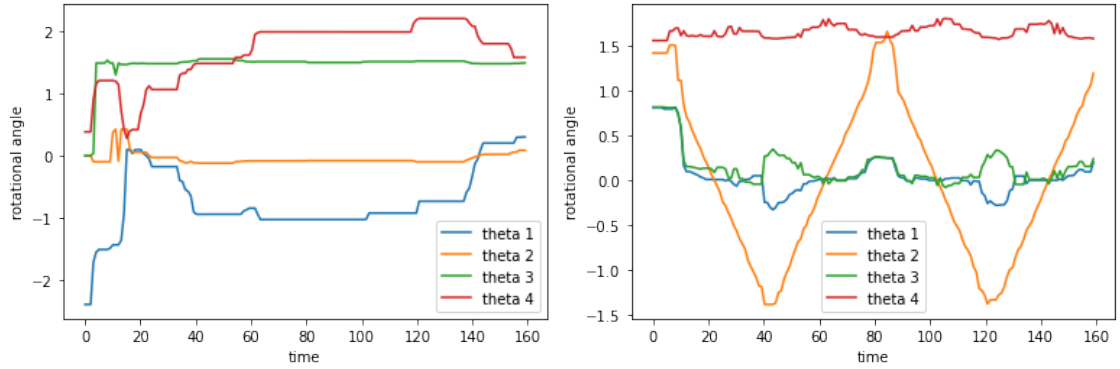
(a) *Ground truth*



(b) *0 error included*



(c) *with 10% error included*



(d) *with 20% error included*

Figure 4.6: *Experiment result of several poses between GA predicted only and fundamental matrix filtered.*

Table 4.3: *Comparison of error between SSGA and PSO.*

Algorithms \ Iterations	300	500	700	900
PSO	65.22	73.23	88.33	44.7
SSGA	15.41	13.15	6.69	8.66

$$ex = \begin{bmatrix} 1 & 0.5 & 0 & -700 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 100 \end{bmatrix} \tag{4.24}$$

Pose 1 represents the T-posture with the rotation of upper arms, whereas Pose 2 is for moving arms straight forward. Pose 3 is for waving arms. The changing of joint variables is shown in the left column of Figure 4.3, 4.4 and 4.5.

In order to choose the suitable optimal algorithms, I first made the comparison of performances between SSGA and the most wildly used Particle Swarm Optimization(PSO), and the result is shown in Table. 4.3. It is obviously that SSGA gave the better result because of its advantages on optimizing single target and continuous optimal issues.

I tested the performance of genetic algorithm under a single camera view. Considering the balance between run time cost and accuracy, in this experiment, I use 300 agents with 900 iterations for each step, and the predicting result for rotational variables θ_1, θ_2, θ_3 and θ_4 are shown in the bottom row of Figure 4.3 4.4 and 4.5. It should be noticed that not all of these four rotations play a significant role in any postures. On the other hand, they show the limit effect for some special postures. For instance, in Pose3, rotational angle θ_2 played limited roles in this dynamic pose, which means that any value of rotation angle of θ_2 does not change the performance for Pose2. Therefore, the prediction for θ_2 is not similar to ground truth.

To test the performance with the noise, I mixed noise with different amounts, and the result is shown in Fig .4.6. The left column shows the performance with GA prediction only, whereas the right column shows the fundamental matrix filtered. For evaluating the similarity between two poses, I introduced Dynamic Time Wrapping (DTW) as the evaluating algorithm. For two poses, the DTW score tends to 0 if these two poses are similar.

The comparison of DTW score is shown in Table .4.4. It is obvious that the two methods perform almost the same for the situation that no noise is included, but the performance will be larger since the noise is increased.

Table 4.4: *Comparison of the performance between with fundamental matrix correction and without case by DTW score.*

	no noise included	10% noise mixed	20% noise mixed
without fundamental matrix correction	21.12	41.70	188.08
with fundamental matrix correction	15.11	19.29	23.78

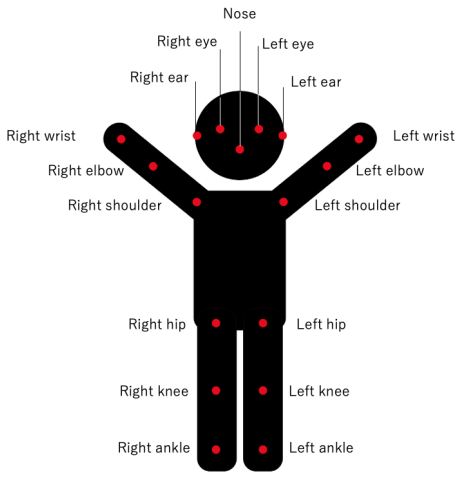


Figure 4.7: *Illustration of key points that can be detected.*

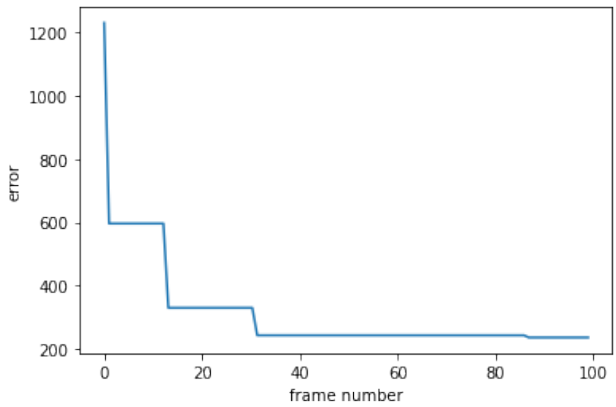


Figure 4.8: *Fundamental error with the frame number increase.*

In the next step, I tried to apply it to the real implementation. Considering the computation capability and cost, in this experiment I utilized the iPod touch 7_{th} as the main device.

The PoseNet is capable of recognizing human poses by detecting 17 key points of human

joints, which is shown in Fig. 4.7. In this paper, I only test the exercises of the upper body, therefore only six joints are under consideration, i.e. left shoulder, left elbow, left wrist, right shoulder, right elbow, right wrist. It is capable of performing the whole system, including the PoseNet deep neural network module with an FPS of 5 to 10.

I set up the environment as shown in Fig 4.2. In this experiment, I set the main iPod in front of users with the distance around 2 meters, and 1.5meters beyond the ground. The second iPod is paralleling to the main iPod, which is around 15 degrees rotated around ground normal. The corresponding joints are labeled and shown respectively, therefore it is possible to calculate the fundamental matrix directly. In this paper, I use 8-point algorithms for estimation. For connivance, I chose a fixed number of frames for estimating the fundamental matrix and chose the best one. The relationship between the error of the fundamental matrix and the number of frames for estimating is shown in Fig. 4.8. Consider the balance between cost and accuracy, it shows that 30-40 frames are enough for estimating the fundamental matrix for initial step.

I test our proposed method with one of the famous calisthenics for rehabilitation in Japan, which is named Arakawa-Koroban calisthenics. This exercise is used for elders to prevent deterioration of bodily functions. Fig. 4.9 shows sample postures of Arakawa-Koroban calisthenics and the evaluated result. In the 3D graphs, blue line represents for the left arm whereas green line represents for the right arm. For two or more users, I also showed the performance, which can be seen in Fig 4.10. It is obvious that the poses of these two users has been recognized with an ideal accuracy.

4.5 Summary

In this paper, I proposed a human pose estimating framework, which estimating three-dimensional positions of human joints without the known intrinsic parameters. It has the advantages that the pose estimation system can be easily established, and the devices of high performing ability are not necessary comparing with current popular deep neural network-based strategies.

On the other hand, different from our previous proposed method, which utilized a genetic algorithm only for estimating human pose, in this paper, I introduced one more camera view for improving the accuracy that is caused by the error of pose detection.

Despite these advantages, there are still some issues that need to be improved. For instance, in the future, I would focus on simplifying the initial step that the initial T-pose would not be necessary anymore, and accuracy is also able to be improved.

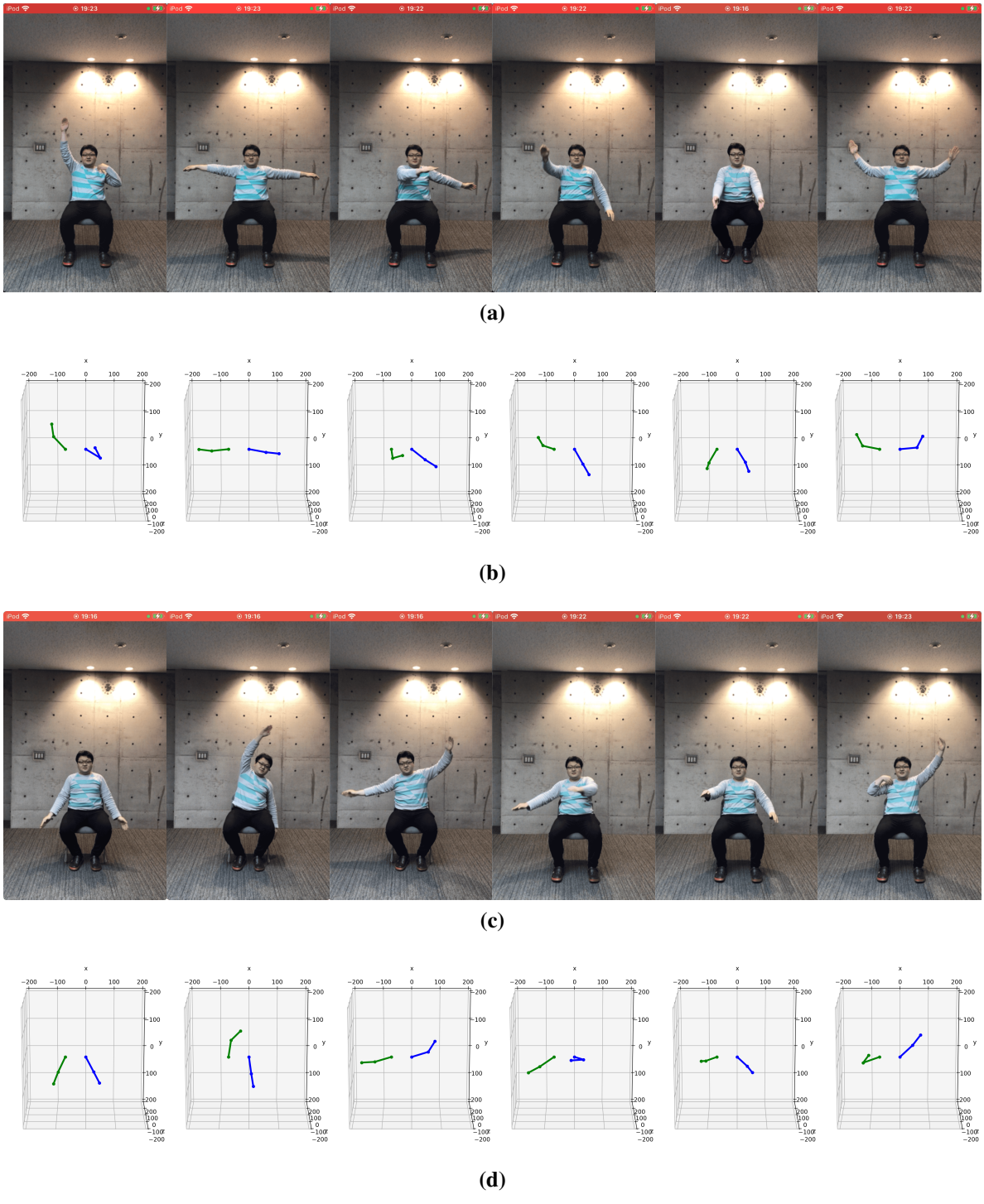
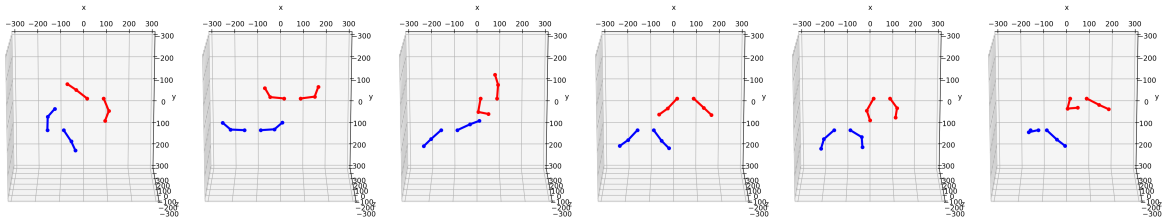


Figure 4.9: *Performance of our proposed method on Arakawa-Koroban calisthenics.*

(a)



(b)

Figure 4.10: *Performance of our proposed method on group Arakawa-Koroban calisthenics.*

Chapter 5

Implementations on Exertainment

In this chapter, we introduce several implementations for physical exercise, which is based on pose recognition proposed previously. These implementations combines physical exercise with rhythm, and the processing of gaming also make physical exercise no monotonous as typical ones. And we also introduced several other kinds of systems for exertainment or implementation, which is mainly based on evolutionary computation.

5.1 Calisthenics for elderly people

Kenko taiso, or healthy calisthenics, implies as the name implies, are exercises for promoting health. Generally, it is a gymnastics program designed for middle-aged to seniors who tend to get stiff, and it is composed mainly of improving endurance, muscular strength, and flexibility (stretching), which are the three major elements of physical fitness. There are no complicated movements or strenuous movements, and there are gymnastics that you can do while sitting in a chair and gymnastics only for the upper and lower body, so even those with back pain or those with bad legs can find a gymnastics program that suits them. Some of the songs, such as European music, Latin music, and trendy songs, are performed by incorporating music to improve the entertainment.

In these implementations, we selected 5 poses, which are main components for ”Arakawa koroban” calisthenics. These poses are shown in Figure 5.1. Users are required to control red ball in the center bottom of the screen, and also need to behave the referent posture in order to avoid the obstacles that come from top to down. These obstacles fall down with the fixed speed, which can be suit to the rhythm of calisthenics.

In order to test the performance of the gesture recognition, we tested the accuracy rate, which is shown in Table. 5.1.

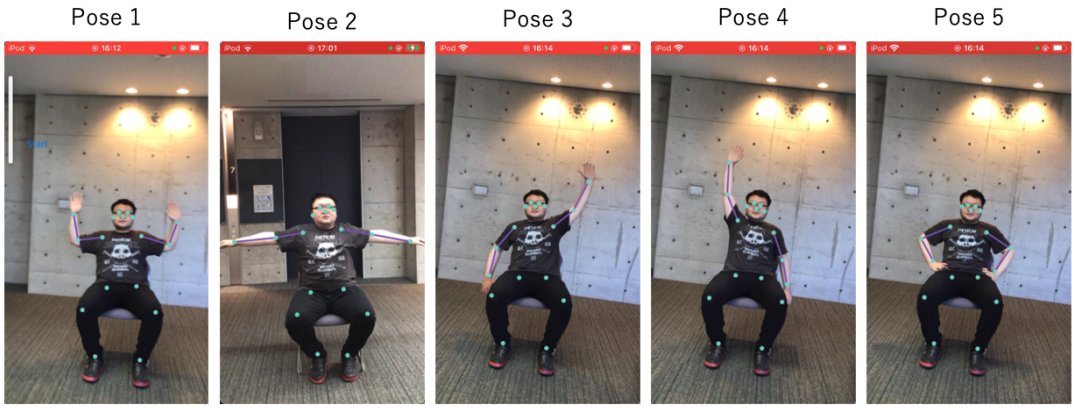


Figure 5.1: *5 sample postures which are selected in the implementation.*

Table 5.1: *Experimental result of pose recognition and classification.*

Ground true \ Prediction	Pose 1	Pose 2	Pose 3	Pose 4	Pose 5
Pose 1	0.94	0.04	0.0	0.0	0.02
Pose 2	0.13	0.87	0.0	0.0	0.0
Pose 3	0.0	0.03	0.74	0.0	0.24
Pose 4	0.0	0.0	0.08	0.82	0.10
Pose 5	0.0	0.0	0.0	0.08	0.92

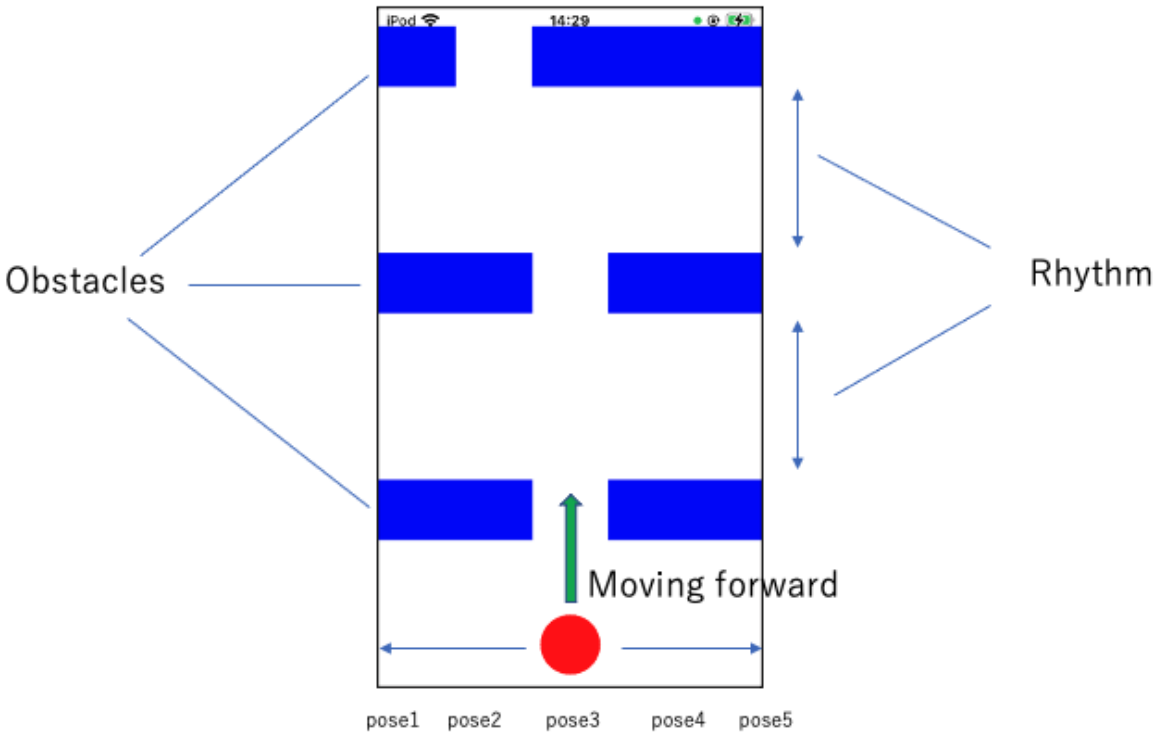


Figure 5.2: *Appearance of rhythm gaming for physical exercise.*

5.1.1 Rhythm implementation of physical exercise for single user

We first introduced our proposed implementation, which is for single users. It is shown as in Figure 5.2. In this implementation, users control the movement of red circle to avoid obstacles by performing corresponding postures. Obstacles fall down with fixed speed, which can be set with the same rhythm of physical exercises in necessary.

5.1.2 Implementation of physical exercise for multiple users

In this section, we introduced our proposed implementation for multiple users. As shown in Figure 5.4, the proposed implementation is similar with ”Whac-A-Mole” game. Whac-A-Mole is a popular arcade game and carnival game, originally known as Mogura Taiji (”Mole Buster”) in Japan. A typical Whac-A-Mole machine consists of a waist-level cabinet with a play area and display screen, and a large, soft, black mallet. Five holes in the play area top are filled with small plastic moles, which pop up at random. Points are scored by whacking each mole as it appears. The faster the reaction the higher the score.

Dementia affects the procedural memory, the part that stores long-term memories such as information on how to do things. Often times, people with dementia have difficulties

Figure 5.3: *Scene for multiple people of physical exercise.*

remembering how to do old tasks such as tying their shoelaces, but the results from the game showed that their procedural memory was still working.

Researchers decided to take their study to the next level by testing the subject's balance and stability. After playing whack-a-mole, players used the Biodex machine at Algonquin College which assigns them several balancing tasks.

It's difficult to track the rate of decline in a person with dementia. Frequent trips to the doctor unfortunately do not show how quickly it is developing. Playing a game such as whack-a-mole would allow the person to be monitored more frequently, even from their home.

In this game, two users are required to cooperate together, which is shown in Figure 5.3. First user controls the position of columns whereas second users controls the position of rows.

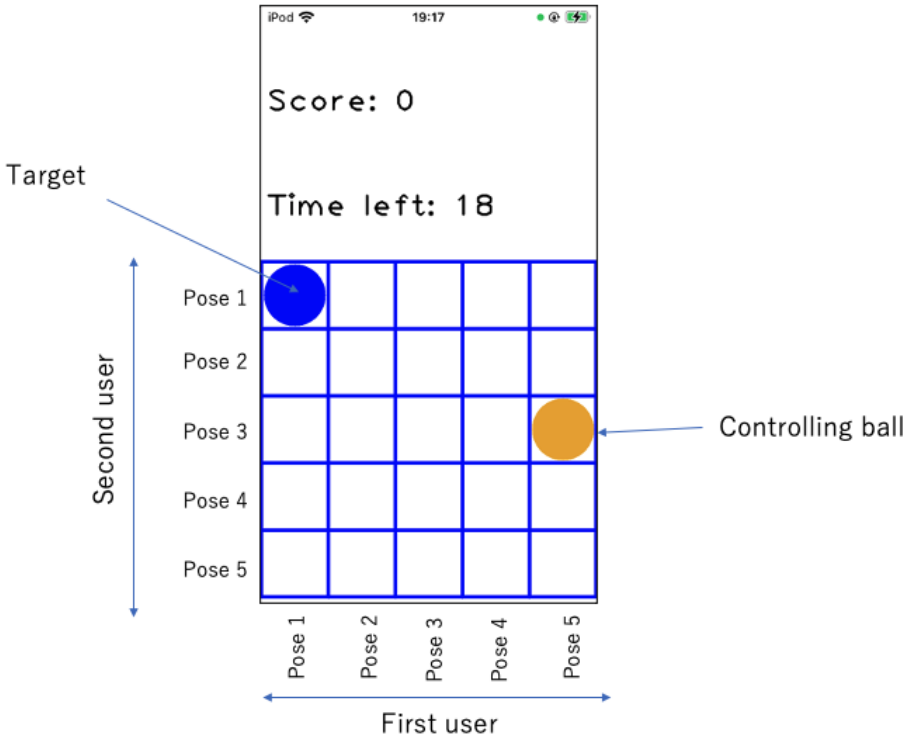


Figure 5.4: *Appearance of rhythm gaming for physical exercise.*

5.2 Implementation on multiple robotic balls tracking

In the past decades, mobile robots have been developed dramatically with the development of technology. Mobile robots are a certain kind of robots which have the capability to move around in their environment rather than being fixed to one physical location, and they have been applied into various kind of fields which from the aerospace industry to family entertainment.

Generally, mobile robots are not limited to land robots such as tracked robots, robot vehicles, and walking robots, but also flying robots. and there are also various kinds of appearances for mobile robots. Mobile robots can be turned to an autonomous mobile robot which means they are capable of navigating an uncontrolled environment by an autonomous controlling system, rather than guidance from the supervisor.

For certain fields such as industry and aerospace, autonomous controlling systems are required to increase precisely as high as possible. Therefore the cost of devices and system complexity would be ignored. On the other hand, personal robots are also playing a more and more important role in our daily life. A personal robot is one that enables an individual to automate the repetitive or menial part of a home or work life making them more productive. Similar to the way that the transition from mainframe computers to personal computers revo-

lutionized personal productivity, the transition from traditional robotics to personal robotics is changing productivity in home and work settings.

Based on this concept, in this paper, we proposed an autonomous controlling system for multiple robotic balls. The system is simply constructed by an infrared camera, a low- price compute unit, which is named raspberry PI, and Sphero SPRK robotic balls. The position of robotic balls are captured by a camera, which is based on the tracking-by-detection, then command of movement would be given by raspberry PI to control robots moving to the target positions. This system is capable of making a series of simulations such as golf, bowling for certain purposes such as entertainment, rehabilitation.

The remainder of this paper is organized as follows: Section 2 mainly introduces the related work for referent methods and technologies, whereas section 3 describes the framework of our proposed system. Section 4 shows the detail of proposed algorithms, and Section 5 gives the result of the experiment, last but not least, conclusion and future extension are given in the final section.

5.2.1 Related Work

Based on the architecture of system construction, systems can be divided into centralized systems or individual systems. A centralized system is a system in which an individual, a group of people or a corporate entity holds the entire control over the functionality of the system. In contrast, a distributed system is a system that consists of several servers, a cluster of servers be it backend, messaging or database running together to perform one single task.

On the other hand, the positioning system is a mechanism for determining the locating of an object (such as a robot) in space, and it plays an important role in the robot system. Exiting positioning systems can be mainly categorized into two groups: the outdoor positioning system and an indoor positioning system. The outdoor positioning system, which is often related to the global position system, is often utilized for localizing objects at the outside environment by satellite technologies, has shown its importance in various fields. The indoor positioning system, on the other hand, is generally defined as a network of devices for locating people or objects in an interior area. Comparing with the outdoor positioning system, it works in a smaller region, but more precise.

Indoor positioning has been researched for many years[75]. and it can be categorized according to different criteria. According to data from sensors, there are various kinds of sensors for exiting the indoor positioning system: IR signals, ultrasound waves, radio frequency, electromagnetic waves, vision-based analysis, etc. For example, in [76], the authors proposed an indoor positioning system based on incident angle measurement of infrared

lights has been suggested. This paper utilized three infrared emitters and incident angle sensor measures the angle differences between every two emitters.

The vision-based positioning system is also widely utilized[77]. It has many advantages, such as a low-cost camera can cover a large range of area compared with other sensors, and the setting of cameras is much more convenient than some embedded devices [**?**]. Nevertheless, the drawbacks of the current system are also obvious, such as the accuracy of current vision-based positioning systems are easily affected by the environment such as illumination.

Different from industrial robots that are chasing for accuracy as high as possible, a system for personal robots should also consider many other factors, such as cost on hardware and software, convenience for the setting.

Considering the scale of the system, in this chapter, we proposed a vision-based positioning & tracking system by a single infrared camera, which shows an ideal performance in the experiment. And we also proposed a semi-model-based tracking algorithm for multiple robotic tracking, which in the first level, rough positions of objects would be detected and attention regions would be generated, then in the second level each of previous tracking objects would be predicted and searched within the detected attention regions. At the same time, in order to fulfill the compute capability of our low-cost computer, we take a series of strategies to make it perform faster.

5.2.2 Construction of system of multiple robotic balls

The proposed system is mainly constructed with three parts: computation unit for computing and controlling, detection unit of infrared camera, and Sphero SPRK robotic balls. The detail is shown in Fig.5.5.

5.2.2.1 Computation core of Raspberry PI

In this chapter, we introduced Raspberry Pi4 Model B as the computation core. The Raspberry is a series of small single-board computers developed in the United Kingdom by the Raspberry Foundation to promote teaching of basic computer science in schools and in developing countries, and the features of Raspberry Pi4 Model B is described in Table5.2. Bluetooth module make it capable of communicating with robots.

5.2.2.2 Detecting sensor of infrared camera

In this paper, we utilized Kinect v1 as the infrared camera for detecting robots. The appearance and detail of Kinect sensor can be seen in Fig.5.5 and Table 5.3. The Kinect

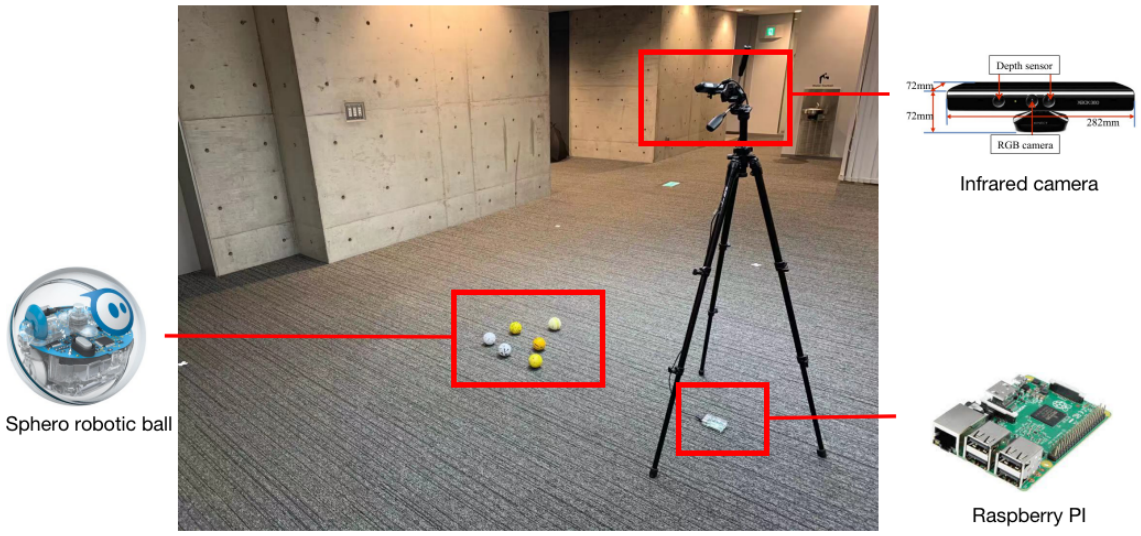


Figure 5.5: *Illustration of construction of system.*

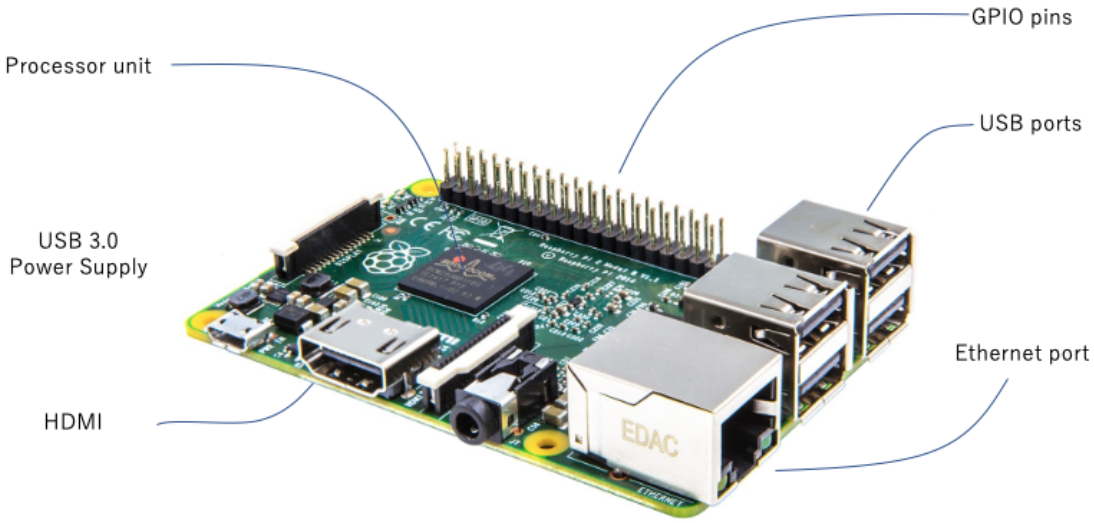


Figure 5.6: *Illustration of raspberry pi.*

Table 5.2: *Features of Raspberry Pi3 Model B.*

OS	Debian, Fedora, Arch Linux
Size	85.60mm x 53.98mm
CPU	Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC
Bluetooth	Bluetooth 4.2, BLE
LAN	2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless
Memory	1GB LPDDR2 SDRAM
USB ports	USB2.0 x 4

Table 5.3: *Features of Kinect v1.*

Depth type	Structured light
RGB resolution	640 x 480
Field of view for RGB camera	62^o x 48.6^o
Field of view for Depth camera	57^o x 43^o
Measurable range	0.8-4m

sensor bar contains two cameras, a special infrared light source, and four microphones. It also contains a stack of signal processing hardware that is able to make sense of all the data that the cameras, infrared light, and microphones can generate. We utilized Kinect as the sensor not only because of its structure of infrared, but also its ability for detecting human body [**?**]. With this function, it is possible to design commands to robots by human postures.

5.2.2.3 Robotic ball: Sphero SPRK

As an intelligent robotic ball, Sphero robot is small and excellent moving robot, which is shown in Fig. 5.7a. The moving mechanism is shown in Fig. 5.7b. There are two factors for controlling movement of robot: speed (range of $[0, 2m]$) for the velocity and the angle (range of $[0, 360^o)$) for moving direction.

Sphero is a robot ball with several features that can be controlled through mobile apps, including computer programs that the students build. There are several main features for Sphero robots: Rolling: The Sphero can roll at a given speed and heading for a given amount of time; Colors: The Sphero can light up in any color, which can be mixed by R G B color bases; Bluetooth: Sphero connects to devices such an iPad, iPhone, and Android devices through wireless Bluetooth connections. This allows the Sphero to be controlled by a number of apps.

5.2.3 Proposed Visual Tracking and Controlling Framework

Because of the feature of Sphero SPRK that it is constructed with transparent plastic, this make it a little difficult to be detected by camera strictly. Therefore we sought an alternative solution for solving this issue.

Because of the feature for the camera that we utilized in this paper, we consider using infrared camera to detect balls with reflective tape covered, shows in Fig. 5.7c, and the performance is shown is Fig.5.8. It is obviously that SPRK with inflictive tape covered is much easier to be detected in images that is captured by infrared camera.

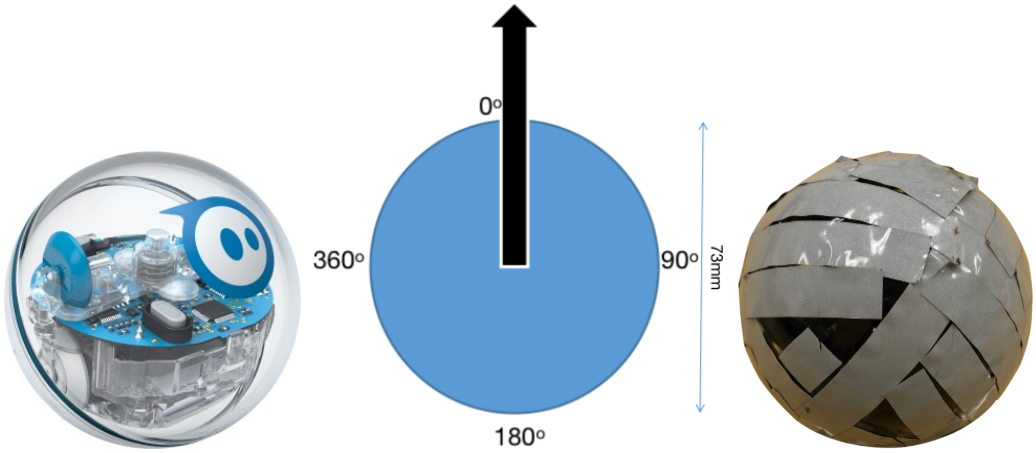


Figure 5.7: *Illustration of SPRK robot. (a) is the original appearance of Sphero SPRK robot; (b) is the feature of Sphero SPRK; (c) is the appearance of SPRK with reflective tape covered.*

5.2.3.1 Nearest Neighbor Evolutionary Algorithm for global searching

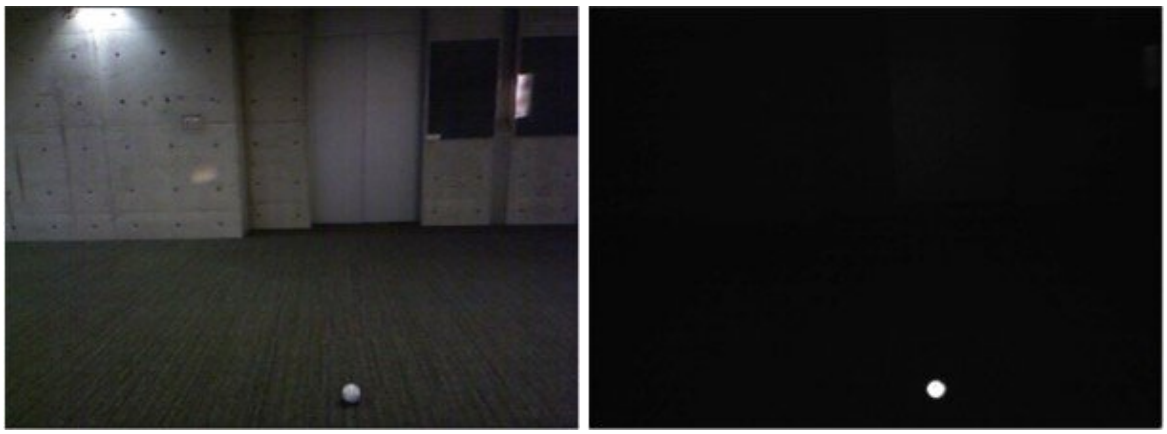
Considering the situation in this paper that the whole framework is built upon a low-cost raspberry PI computer, it is overwhelming important for considering the computation cost as well as the accuracy.

Considering about the computing capability of raspberry PI, it is necessary to reduce the computation cost as much as possible. However, currently evolutionary algorithms contains a large part of random selection and computation, which would cost too much if it is utilized for optimization in a large-size image. Therefore, in this paper, we proposed an image-pyramid-based two level evolutionary algorithm for multi-target tracking. The illustration of proposed algorithm is shown in Fig. 5.9.

Evolutionary computation [78] such as genetic algorithm (GA) is a filed of simulating evolution on a computer. Evolutionary optimization methods are fundamentally iterative generation and alternation processes of candidate solutions. [79] has proposed evolutionary robot vision for people tracking which shows efficient performance of multiple human face detection [80].

In the first step, global searching is proposed to find out attention regions. We introduced Nearest Neighbor Evolutionary Algorithm (NNEA) for this task [81], and a series of attention regions would be detected before precise search start. In order to reduce computation cost, global searching is proposed in the lower level of image pyramid.

An individual in NNEA is notated as $g_i = \{x_i, y_i, r_i\}$, which represent for coordinate of



(a) *Original image* **(b)** *Infrared image*

Figure 5.8: *Comparison of the appearance of SPRK taped between the normal image and infrared image.*

two dimension and radius respectively. Considering about the appearance of robots in Fig. 5.8, the fitness function for particle g_i can be formulated as:

$$f(g_i) = \sum_{(w-x_i)^2+(h-y_i)^2 \leq r_i^2} \lambda_1(I(w,h) - \lambda_2) \tag{5.1}$$

where $I(w,h)$ represent for gray scale value at the position (w,h) in the infrared image, and λ_1 and λ_2 are the constants weight and threshold.

The detail of NNEA for global searching is shown in Algorithm 3. Attention region $r_k^i = (x_{r_k^i}, y_{r_k^i}, w_{r_k^i}, h_{r_k^i}) \in R^i$ would be generated after NNEA is processed, where $R^i = \{r_1^i, r_2^i ... r_m^i\}$ represents the set of attention regions at time i.

5.2.3.2 Steady State Genetic Algorithm for local tracking

More precise detection and tracking is needed after attention regions have been selected by global searching. In this part, we introduced Steady State Genetic Algorithm(SSGA) for local searching & tracking within the attention regions [82] [83].

This GA is steady state meaning that there are no generations. Standard GA creates new offspring from the members of an old population using the genetic operators, and finished until the whole old population are replaced by the created offspring. Different from that, SSGA tournament selection does not replace the selected individuals in the population, and instead of adding the children of the selected parents into the next generation, the two best individuals out of the two parents and two children are added back into the population so that

Algorithm 3 Nearest Neighbor Evolutionary Algorithm for attention regions detection.

INPUT
$G = \{g_1, g_2...g_N\}$
θ as the threshold filter
OUTPUT
$R = \{r_1, r_2...r_m\}$ for the set of candidate regions
m is the number of candidate regions
START
for $iteration \leftarrow 1$ to max-iteration **do**
  for $j \leftarrow 1$ to number of genetic units **do**
    Calculate fitness $f(g_j)$
    Reproduction the offspring
    Perform crossover with a probability p_1
    Perform mutation with a probability p_2
    Replace g_j with a probability p_3
  end for
end for
for $j \leftarrow 1$ to N the number of genetic units **do**
  $G_c \leftarrow \{g_j\} \cup G_t$ if $f(g_j) \geq \theta$
end for
$R = \{g_1\}$ add g_1 to R for initialization
$m = 1$
for $i \leftarrow 2$ to number of particles in G_c **do**
  find the g_i in some cluster $g_j \in R$, i.e. $\|g_i, g_j\|_2^2$.
  if $\|g_i, g_j\|_2^2 < \theta$ **then**
    $R \leftarrow R \cup \{g_i\}$
    $m \leftarrow m + 1$;
  end if
end for

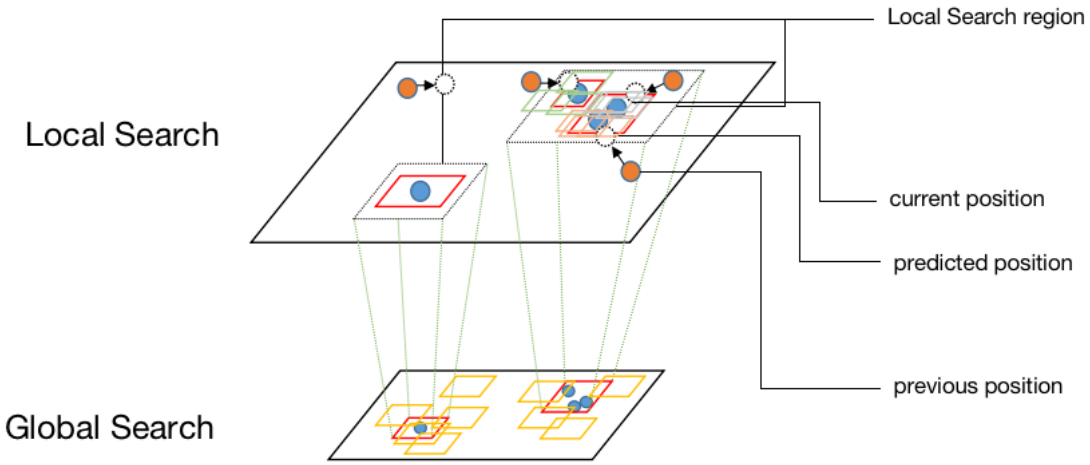


Figure 5.9: *Illustration of proposed combined Evolutionary Algorithm for tracking. In the global search, genetic particles search globally. Once the rough position of candidates are detected, more precise search in the image of higher level will be started locally around the candidates.*

the population size remains constant. Therefore the performance of population generation for SSGA would be more stable than standard GA.

We denote tracking target j at time i as $t_j^i = (\mathbf{x}_{t_j^i}, \mathbf{u}_{t_j^i} r_{t_j^i}) \in T^i$, where $T^i = \{t_1^i, t_2^i ... t_m^i\}$ are the tracked targets in time i, and $\mathbf{x}_{t_j^i}$, $\mathbf{u}_{t_j^i}$ and $r_{t_j^i}$ represent for vector of position, vector of velocities and radius of target j at frame i respectively. For a tracking target t_j^{i-1} at time $i-1$, it is necessary to predict its position $\hat{t}_j^i$ at time t by the following equations:

$$\hat{\mathbf{x}}_{t_j^i} \leftarrow \mathbf{x}_{t_j^{i-1}} + \mathbf{u}_{t_j^{i-1}} \cdot \Delta t + \eta_1 \tag{5.2}$$

$$\hat{\mathbf{u}}_{t_j^i} \leftarrow \mathbf{u}_{t_j^{i-1}} + \eta_2 \tag{5.3}$$

$$\hat{r}_{t_j^i} \leftarrow r_{t_j^{i-1}} + \eta_3 \tag{5.4}$$

Where η_1, η_2, η_3 denote the white noise. It still to have a necessary step of evaluating whether new target appeared or exiting target disappeared before tracking. For an exiting target t_j^i, it would be regarded as disappeared at time i if it satisfy:

$$\hat{t}_j^i \notin r_t^i \quad for \ \forall r_t^i \in R^i \tag{5.5}$$

Similarly, a new target would be regarded as new target if:

$$\hat{t}_j^i \notin r_t^i \quad for \ \forall \hat{t}_j^{i-1} \in T^{i-1} \tag{5.6}$$

The reason we selected Steady-state Genetic Algorithm (SSGA) as the tracking solution is that SSGA performs fast and stable under the situation of stable. It is capable of finding optimization with the shortest crossover and mutation iterations with the known prior experience. The detail of SSGA is shown in Algorithm 4:

Algorithm 4 Steady State Genetic Algorithm for multiple objects tracking.

INPUT
$G = \{g_1, g_2 ... g_N\}$
θ as the threshold filter
OUTPUT
$R = \{r_1, r_2 ... r_m\}$ for the set of candidate regions
m is the number of candidate regions
START
for $iteration \leftarrow 1$ to max iteration **do**
  Calculate fitness $f(g_j)$
  Select g_i with the smallest $f(g_i)$
  Perform crossover with g_k of largest $f(g_k)$ with a probability p_1
  Perform mutation with a probability p_2
end for
for $j \leftarrow 1$ to N the number of genetic units **do**
  $G_c \leftarrow \{g_j\} \cup G_t$ if $f(g_j) \geq \theta$
end for

The fitness value of selected particles g_k for each tracking target t_j^i is described as:

$$F_{t_j^i}(g_k) = \begin{cases} exp^{-\frac{\|g_k - \hat{t}_j^i\|_2^2}{\sigma^2}} \dot{f}(g_k) & if \ exp^{-\frac{\|g_k - \hat{t}_j^i\|_2^2}{\sigma^2}} > \tau \\ 0 & otherwise \end{cases} \tag{5.7}$$

Where σ and is the constant for controlling the spread of comparing, and τ as the threshold value.

5.2.3.3 Controlling of robot movement

The evaluation of velocity of robots are not so precise because of a series of errors during the evaluation in the previous procedures. According to the systematic of movement of SPRK which is shown in Fig. 5.7c, the essential of movement controlling for SPRK robots is the controlling of angle.

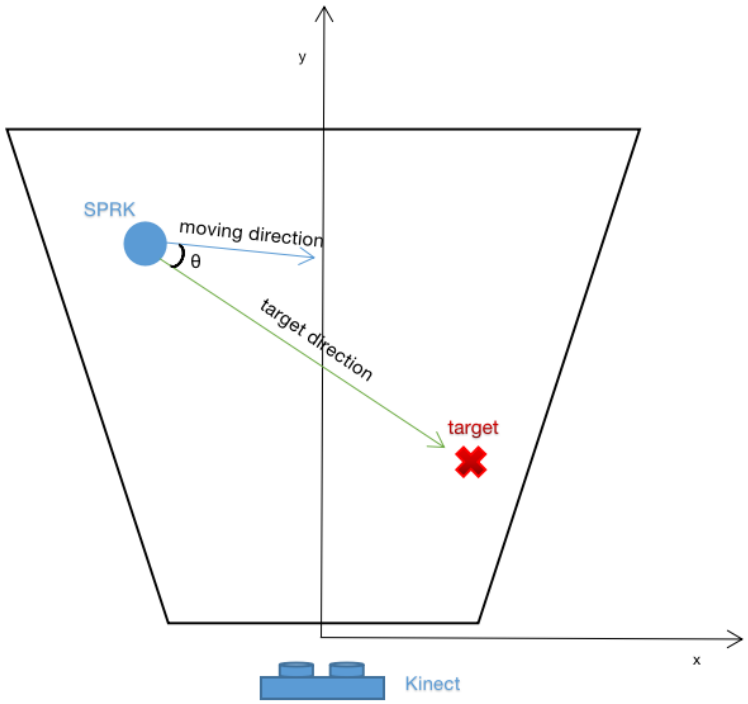


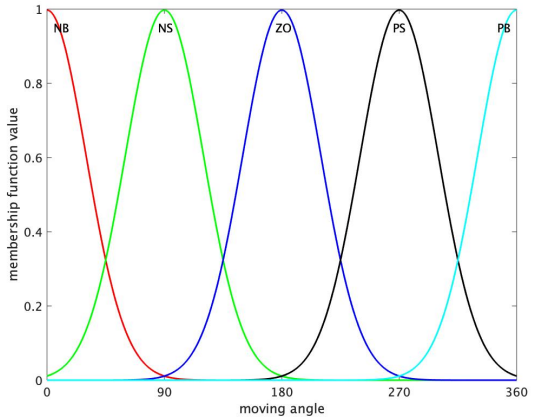
Figure 5.10: *Illustration for the controlling of movement.*

Nevertheless, as shown in Fig. 5.10, there is a difference between the angle of robot and the angle of target, and it would be a little complicated to compute directly. Based on this situation, in this paper, we introduced fuzzy theory to solve this issue.

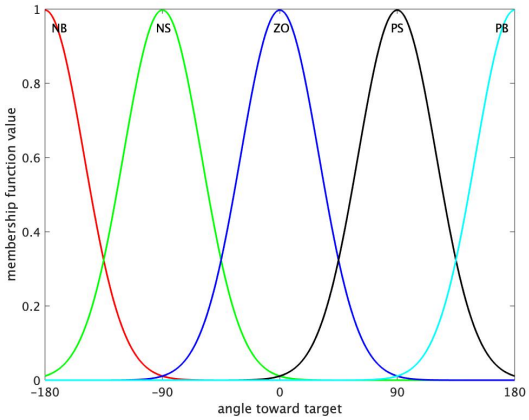
In this framework, we utilize fuzzy controller that give the output of shifting angle θ_{out} with the two input values, i.e. angle of real direction θ_{in1}, which is shown in Fig.5.7c, and angle of the direction to the targets θ_{in2}, shown in Fig.5.10. The range of them are $[0^o, 360^o)$, $[-180^o, 180^o)$ and $[-90^o, 90^o]$ respectively.

In this part, we utilized fuzzy controller to control the movement of robots. We introduce two inputs for the fuzzilization F_{in_1} and F_{in_2} which represent for θ and $\Delta\theta$ shown in Fig. 5.10 respectively, where $\Delta\theta$ means the change comparing with previous frame. And for inputs F_{in_1} and F_{in_2}, we fuzzilize them into five fuzzy sets within their domain respectively:NB, NS, ZO, PS, PB, where represent for negative big, negative small, zero, positive small and positive big respectively. The fuzzy sets for F_{in_1} and F_{in_2} are shown in the first row of Fig. 5.11a .

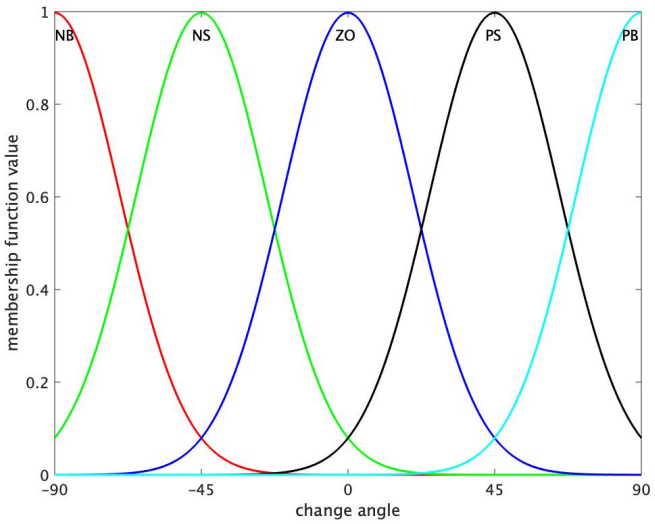
It is the same as in defuzzilization, and the output is described as $F_{out} = \{NB, NS, ZO, PS, PB\}$, and we utilized Gaussian function as the membership function. The distribution of membership functions are shown in Fig. 5.11.



(a) *Fuzzy membership function of input 1.*



(b) *Fuzzy membership function of input 2.*



(c) *Fuzzy membership function of output.*

Figure 5.11: *Fuzzy membership function for input and output angles.*

θ_{in1} \ θ_{in2}	NB	NS	ZO	PS	PB
NB	NM	NM	NS	NS	ZO
NS	NM	NS	NS	ZO	PS
ZO	NS	NS	ZO	PS	PS
PS	NS	ZO	PS	PS	PM
PB	ZO	PS	PS	PM	PM

Table 5.4: *Fuzzy rule base for movement controlling.*

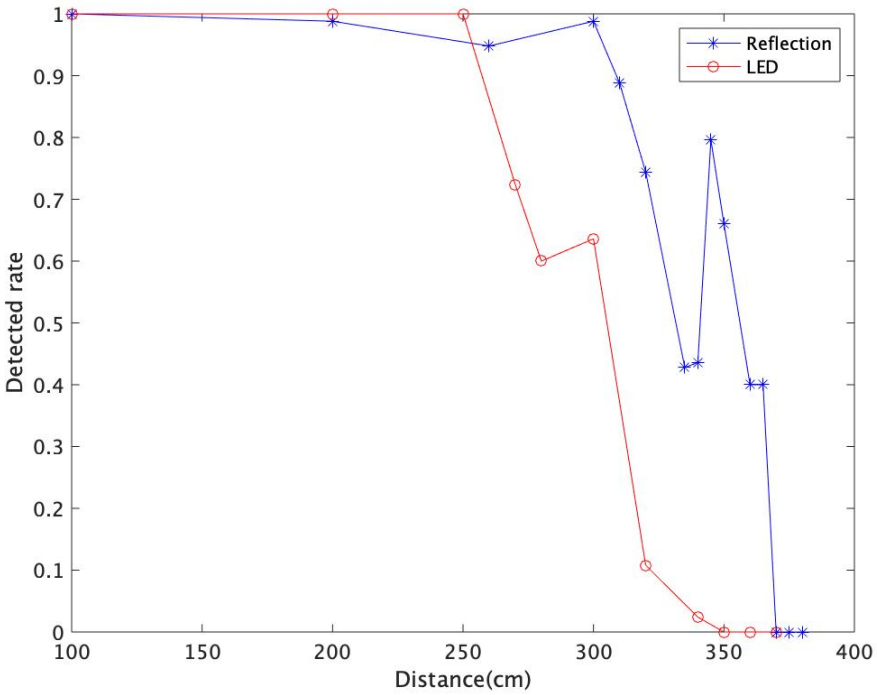


Figure 5.12: *Comparing of detection performances between with reflective tape and with LED light.*

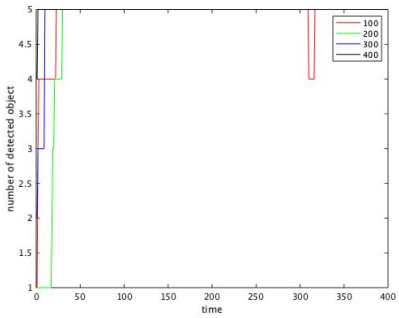
5.2.4 Experiment result

In order to prove the performance with reflective tape, we made the comparing experiment between the reflective tape and LED light. For the case of LED light, we detect the color for segmentation and locate the position of the robot. The result is shown in Fig. 5.12. It is obvious that at the range from 2.5 meters, the detectable rate with reflective tape is much larger than LED-based detection.

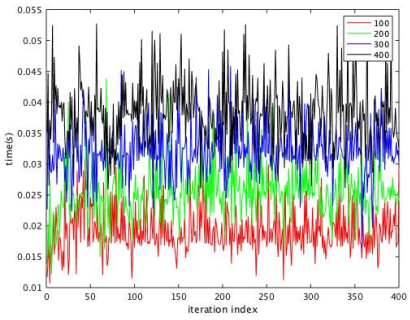
In the next step, we experimented to test the performance of the detection and tracking. In order to make a comparison, in this paper, we made four situations with a different number of robotic balls. Since that the process of global search would cost most of the computation time among the whole processing, we test the accuracy and time consuming with the different number of particles, which includes 100, 200, 300, 400 particles respectively.

The result is shown in Fig. 5.13. It is obvious that the more particles we utilized for global tracking, the more of targets can be detected, and the more accurate it would be. Nevertheless, the computation cost will also rise at the same time. In the following steps, we chose 300 particles in order to find an acceptable performance in both accuracy and time-consuming.

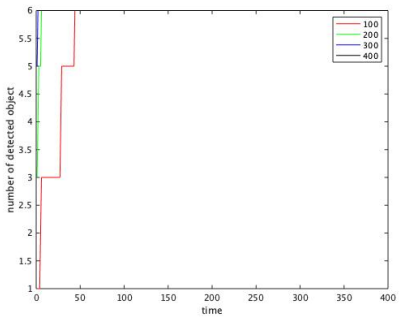
In the next step, we experimented with the performance of multiple robotic ball control-



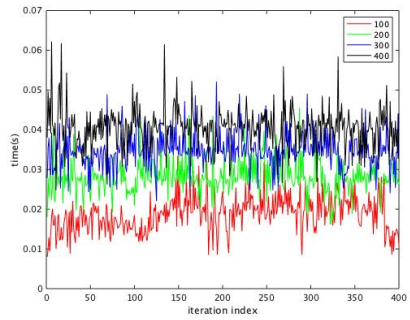
(a) *tracking result for 5 robots.*



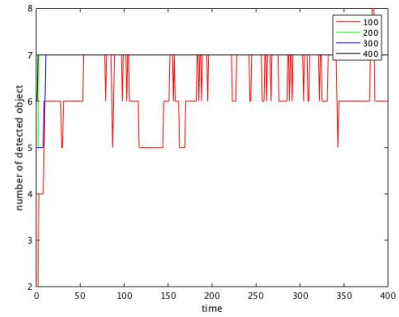
(b) *time consuming for 5 robots.*



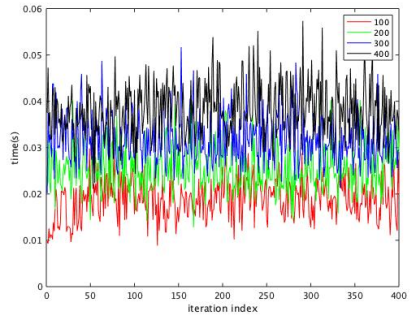
(c) *tracking result for 6 robots.*



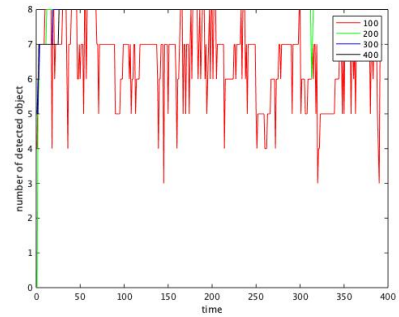
(d) *time consuming for 6 robots.*



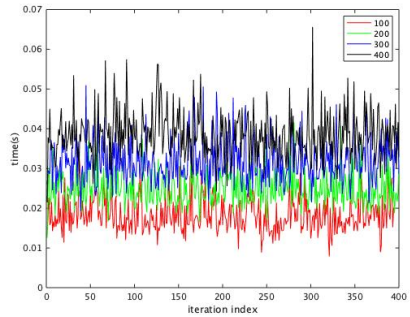
(e) *tracking result for 7 robots.*



(f) *time consuming for 7 robots.*

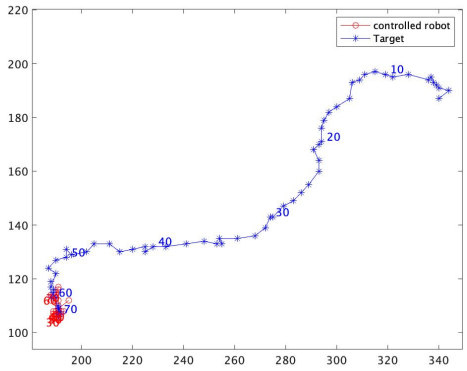


(g) *tracking result for 8 robots.*

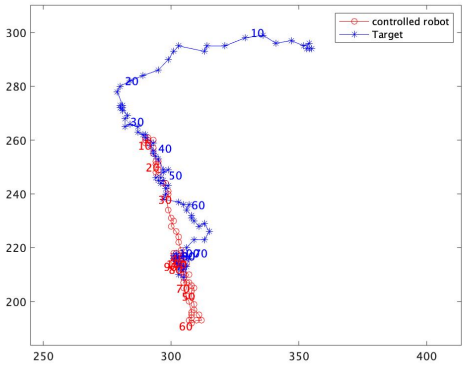


(h) *time consuming for 8 robots.*

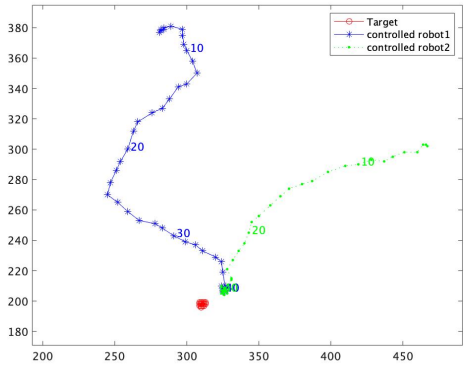
Figure 5.13: *Performance with different number of particles in four cases. The figures in up left column the accuracy with different robotic ball exits, whereas the right column shows the corresponding time cost.*



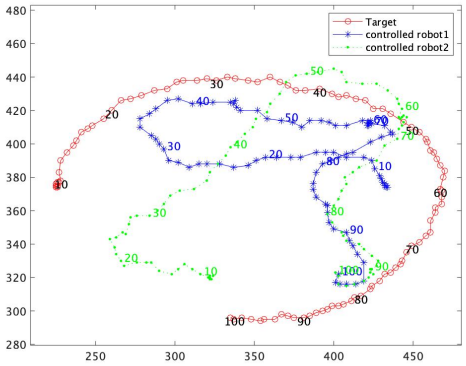
(a) *Single target robot which is static.*



(b) *Single target robot which moved straightly.*



(c) *Single target robot is static whereas two robots controlled.*



(d) *Single target robot moved roundly whereas two robots controlled.*

Figure 5.14: *Experiment result of robotic balls controlling.*

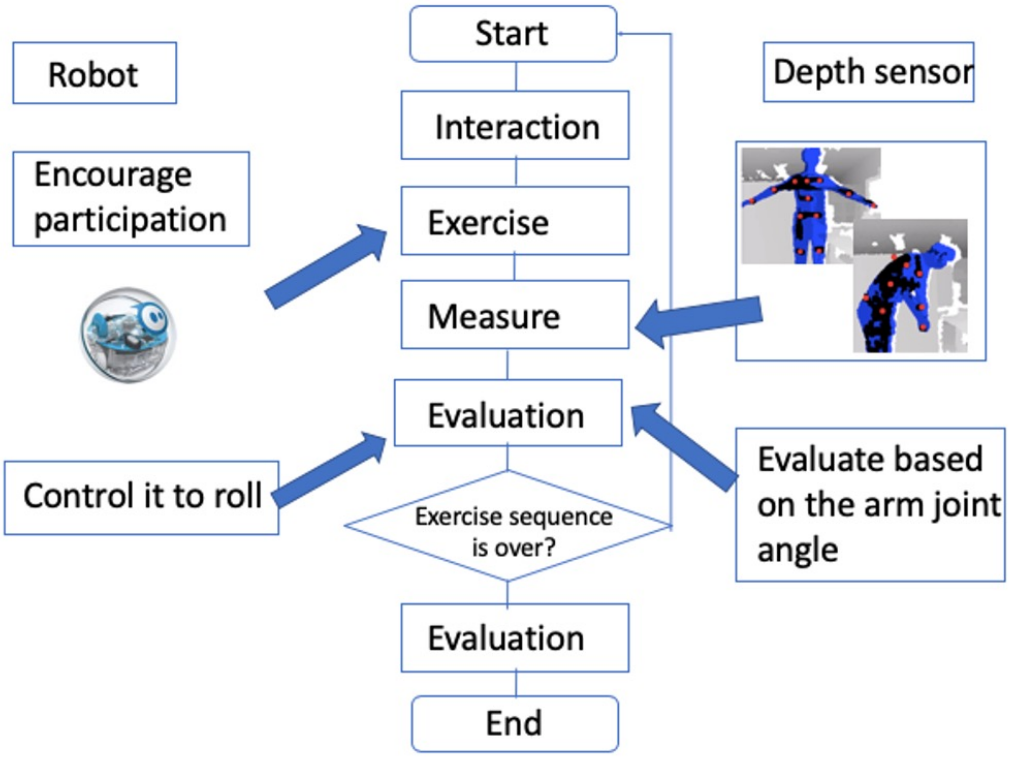


Figure 5.15: *Flowchart of evaluation process.*

ling. We designed a series of scenarios. In these scenarios, we regard one robot as the target and controlling other robots on the screen to move toward the target automatically.

The result is shown in Fig. 5.14. The red line and dots represents for the target robot's trail, whereas blue and green ones for the controlling robots, and the number shown in the image represent the frame index for position. In these experiment, controlled robots moved to target correctly in both dynamic and static situations.

5.2.5 Discussion

In this section, we proposed an intelligent autonomous control system for multiple robotic balls, which can be set and implemented conveniently. The system is simply constructed by an infrared camera, a low-price computer which is called raspberry PI, and multiple Sphero SPRK robotic balls. Both detecting and controlling the robot depending on the frames captured by the camera sensor. Therefore the system would be easy to be constructed and suitable in most environments.

On the other hand, this system provides not only the surveillance but also give motive

Figure 5.16: *Appearance of the airport robot. Left one is the prototype whereas right one is advanced designed shape.*

interactions. The Sphero robot would move towards the human that also detected by the depth camera, and is allowed to show special actions when given the command. Because of these, it can be applied as the robot partners for children and inspire their curiosity and interest.

5.3 Implementation on people tracking for navigating robot

In this section, we introduce another implementation of no-instructed evolutionary computation. It is an evolutionary people tracking system in a robot system that is used for giving the information service in not only airport but also public areas. And the robot has the appearance as figure 1 shows. The left one is the prototype of the robot and the right is well-designed type. This robot system has the ability to recognize the people in front of it and counting the number of them. For the people who might have some troubles, it would moves towards this people and then provides a information service automatically.

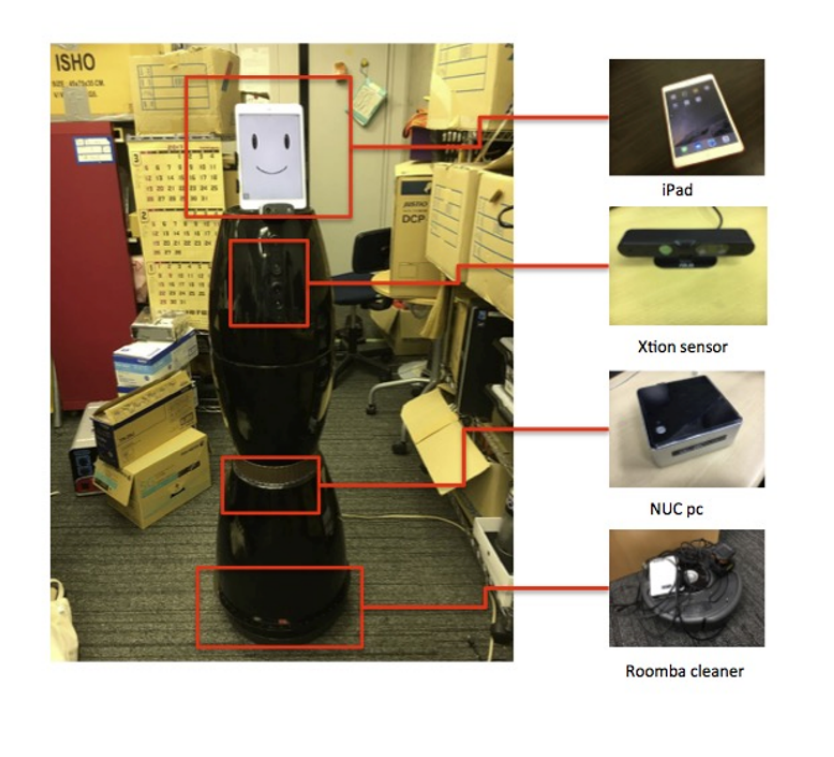


Figure 5.17: *Structure of the robot system.*

5.3.1 System Description

The whole system is constructed by four parts: surveillance sensor; computation core; moving robot and communication interface, which the appearance is shown in Figure 5.17.

Surveillance sensor performs like the ”eye” of the system and depth camera is usually used. We use ASUS Xtion live pro in this case. And it is deserved to be mentioned that we just rotate the Xtion to make it more suitable in the robot. Therefore the open source for referent applications such as toolkit for human skeleton extraction is not available in our case. We need to find a suitable solution that is less depends on other sources.

Computation core is the ”brain” of the robot. It receives the data from the Xtion sensor, and recognizes the human in front of it, and also keeps calculating the coordinate of the target during time the robot moving towards the target person. And it is also the server of the TCP protocol that sends the referent command to the clients according to the situation.

And for robot’s base, we used robot cleaner iRobot for robot moving component. It has lots of advantages such as that it provides its software development kit as the open source and we can control its moving at will.

Last but not least, good interface would leave a good impression to users. Therefore we developed an application under ios system and ran it on iPad for the communicating interface

between robot and users.

In this implementation, we focus on the part of robot vision, and our main target is to detect the person in front of the robot correctly. For more detail, that detect the number of the person and make a simple judgment that whether these person need help or not. After then, it will send command to the iPad interface for making different operations. The detail of this part will be given in the following section.

5.3.2 Human detection and recognition

Human detection and recognition is the first and also the most important part in this robot system. Accurate detecting is the premise for the following computations. In this case, we applied depth camera for the detecting sensor and all of the human detection and recognition will based on the RGB-D images gotten from it.

5.3.2.1 Coordinate transform

The images captured by depth camera have extra information of the distance comparing with other single cameras. Thus it is possible to transform the pixels in this image into 3 dimensional voxels if we got the parameters of the depth sensors once we got the RGB-D images. For a pixel $p(u, v, d)$ on the image, where u and v are the position of the pixel on camera screen coordinate and d is the depth of this pixel. We make the transform it into the 3 dimensional voxel that $p(u, v, d) \rightarrow v(x, y, z)$ under the camera coordinate system. And the coordinate system can be shown in Figure 5.18 (a).

5.3.2.2 Foreground detection

It is important to detect the foreground target. In our system, we only concern about the target that within a certain area and we call it region of interest. The foreground voxel sets V can be detected as

$$V = \{v_i(x_i, y_i, z_i, b_i)\} \tag{5.8}$$

Where x_i, y_i, z_i are the coordinate that calculated as previous subsection, and b_i is a Boolean value and will be 1 if v_i satisfy $x_i \in [0, t_x]$,$y_i \in [0, t_y]$ and $z_i \in [0, t_z]$ (t_x, t_y and t_z are the thresholds for region of interest at each direction respectively), and 0 otherwise.

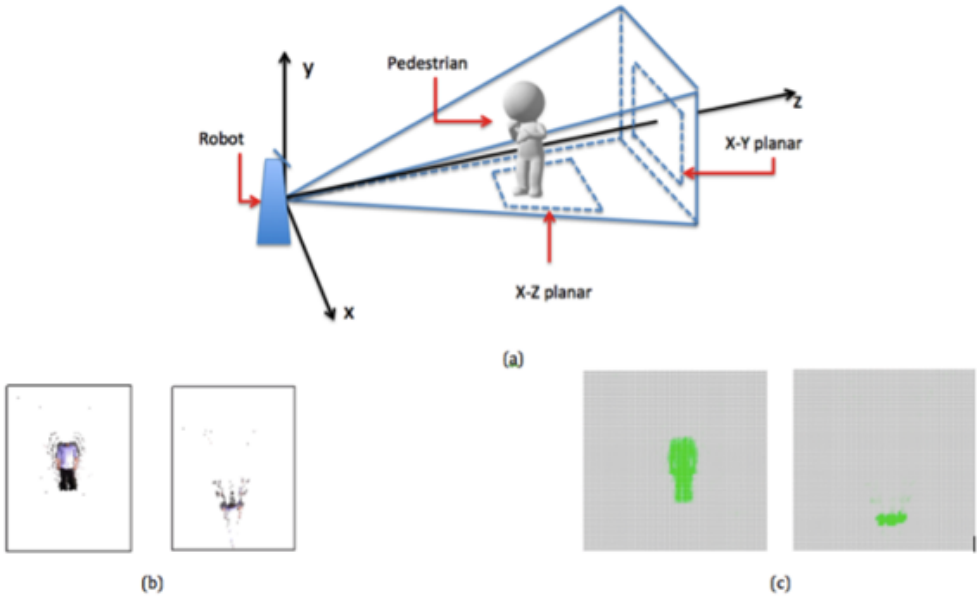


Figure 5.18: *rgbd images and the projections on x-y and x-z planar. (a) robot coordinate system; (b) the projection images on x-y and x-z planar respectively; (c) the discrete projection space of voxels on x-y and x-z planar respectively.*

5.3.2.3 Human candidate detection and nearest neighbor clustering

In this robot system, we applied genetic algorithm for human detection and nearest neighbor is applied for the calculation of human number.

It is a little slow to calculate in the coordinate space described above. For the purpose of real-time approach, we shrink down this continuous 3 dimensional space into a smaller discrete 3 dimensional space. And the shrinking size should depend on different situations. In our case, we shrieked all the region of interest into $M \times M \times M$ discrete cubes, and each cube represents the smallest unit in this area.

First proposed by John Henry Holland[84] Genetic algorithm is a random-based algorithm inspired by the process of natural selection that belongs to the larger class of evolutionary algorithms, and it is commonly used to generate high-quality solutions to optimization and search problems by relying on bio-inspired operators such as mutation, crossover and selection. We attribute all the genetic units into these cubs and let all the units reproduce, crossover and mutate for a given times of iteration, and choose the units with the highest fitness values as the detected targets.

Suppose there are a series of genetic units $G = \{g_j(x_{gj}, y_{gj}, z_{gj})\}$,where $j = 1, 2....N$,

and N is the number of genetic units. For initial part, we attribute all of the genetic units into region of interested averagely and with a little random shifting.

For each divide cube $c_i(l_{c_i}, m_{c_i}, n_{c_i})$ that satisfy $l_{c_i}, n_{c_i}, n_{c_i} = 1, 2....M$, its projection on x-y and x-z planar as 5.18 (c) shows.

For conventional, we project all these cubes into x-y planar and x-z planar and calculate the density on these two planar respectively.

And we choose the density that foreground voxel project into two planar as the fitness function of genetic units as fig 2(c) shows, the x-y planar and x-z planar have been compressed into a series of grids, and each grid represents for a position on its corresponding planar. Let G_{x-y} and G_{x-z} represent the grids of grids at x-y and x-z planar respectively, and the size of grids are $m \times m$, the value of each grid represents the number of voxels that project into this grid:

$$G_{x-y} = \{g_{x-y}(j,k) | 0 < j < m, 0 < k < m\} \tag{5.9}$$

And for the fitness value of each grid is

$$f_1(j,k) = \sum v_i \tag{5.10}$$

Where the coordinate of v_i satisfies $\frac{j*t_x}{M} < x_i < \frac{(j+1)*t_x}{M}$, $\frac{k*t_y}{M} < y_i < \frac{(k+1)*t_y}{M}$ and $b_i = 1$.

Similarity, we can also make the fitness value on x-z planar

$$G_{x-z} = \{g_{x-z}(j,l) | 0 < j < m, 0 < l < m\} \tag{5.11}$$

And the value for each grid on x-z planar calculated from

$$f_2(j,l) = \sum v_i \tag{5.12}$$

Where the coordinate of v_i satisfies $\frac{j*t_x}{M} < x_i < \frac{(j+1)*t_x}{M}$, $\frac{l*t_z}{M} < z_i < \frac{(l+1)*t_z}{M}$, and $b_i = 1$.

According to the formulas above, the fitness function for a genetic unit could described as:

$$f(g_i) = \lambda_1 f_1(j,k) + \lambda_2 f_2(j,l) \tag{5.13}$$

Where the position $c(l,m,n)$ for a genetic unit represents the position within the discrete space, and λ_1,λ_2 are fixed controlling parameters.

The set of candidate targets contains all the positions that might be the targets. nevertheless, there is a situation that some of these positions belong to one human because these

position probably point out part of these human. Thus it is necessary to use clustering algorithm to cluster these candidate into one cluster.

In some cases, different genetic units would have found the same targets, thus it is important to cluster these units into one cluster. Clustering algorithms can be categories into two kinds: sequence-based and distance-based. Typical algorithms for sequence-based like k-means clustering algorithm, and nearest neighbor can be a good sample for distance-based algorithm. Both of these algorithms have advantages and defects. For instance, we have to get the exact number of the clusters before calculation if we want to applied k-means algorithm, and in most cases it is hard to get such information beforehand. Thus this is one of the reasons that we chose nearest neighbor instead.

As mentioned before, it is necessary to cluster all of the genetic units that have the high fitness values. And we tried the Nearest Neighbor clustering algorithm for this issue.

Combined with genetic algorithm and nearest neighbor algorithm, the pseudo code can be described as Algorithm 5.

The radius would be a significant value in this system. It largely affects the result of the tracking. Thus a proper value would be the ideal. The experiment section will show the affection of the radius value.

5.3.2.4 Targets tracking

We have already recognized the number and positions of the targets by the process that described above, but it is still need to track the targets from different frames for a series of time. We also use the nearest neighbor for tracking in this part.

Suppose all the tracked targets at time $p-1$ are in the set $T^{p-1} = \{t_i^{p-1}\}, i = 1, 2 \ldots m$, where t_i^{p-1} is the tracked target at time $p-1$ and has the feature $t_i^{p-1} = (j_i, k_i, l_i, d_i)$, where j_i, k_i, l_i are the position of the target at 3 different direction in discrete space respectively and d_i is time decay value.

For the next time p, we have detected candidate targets $C^p = \{c_o^p\}, j = 1, 2 \ldots n$, and $c_o^p = \{j_o, k_o, l_o\}$ is the corresponding position. For each targets t_i^{p-1}, we try to find its current position among the candidate targets set C^p.

We try to find the best matching between current targets and candidate targets by bidirectional matching like

$$\theta_{t_i^{p-1}} = \underset{c_o^p \in C^p}{\arg\min}\ nn(t_i^{p-1}, c_o^p) \tag{5.14}$$

$$\delta_{c_o^p} = \underset{t_k^{p-1} \in T}{\arg\min}\ nn(c_o^p, t_k^{p-1}) \tag{5.15}$$

Algorithm 5 Combined evolutionary multiple detection algorithm

INPUT
$G = \{g_1, g_2...g_N\}$
θ as the threshold filter
OUTPUT
$G_t \subset G$ for the set of candidate targets
K is the number of detected people
START
for $i \leftarrow 2$ to t for the maximum iteration time **do**
  for $j \leftarrow 1$ to N the number of genetic units **do**
    Calculating the fitness value $f(g_j)$ for each genetic g_j
    Reproduction the offspring according to the fitness value
    Generate offspring from the crossover of two parents
    Mutate some generated offspring
  end for
end for
for $j \leftarrow 1$ to N the number of genetic units **do**
  $G_c \leftarrow \{g_j\} \cup G_t$ if $f(g_j) \geq \theta$
end for
$K_1 = \{t_1\}$ add K_1 to K for initialization
$k = 1$
for $i \leftarrow 2$ to n **do**
  find the t_i in some cluster K_m in K such that $dis(t_i, k_m)$ is the smallest.
  if $dis(t_i, t_m) < \theta$ **then**
    $K_m \leftarrow K_m \cup \{t_i\}$
  else
    $k \leftarrow k + 1$; $K_k = \{t_i\}$; add K_k to K
  end if
end for

Where the function $nn(\cdot)$ satisfies:

$$nn(a,b) = \begin{cases} \|a-b\| & if\ \|a-b\| \leq r \\ \infty & otherwise \end{cases} \tag{5.16}$$

For r is the threshold for the radius.

The target t_i^{p-1} will update to t_i^p and put into set T^p only if it has found its corresponding candidate targets c_o^p, i.e., $\theta_{t_i^{p-1}} = c_o^p$ and $\delta_{c_o^p} = t_i^{p-1}$. And the time decay value d_i will be refreshed to 0.

And if t_i^{p-1} cannot find its corresponding candidate target or $\theta_{t_i^{p-1}} = c_o^p$ but $\delta_{c_o^p} \neq t_i^{p-1}$, then it means that this target is occluded or disappear. d_i would refreshed as $d_i = d_i + 1$. And if d_i has risen to ti that is the threshold, and then this target will be deleted.

For the candidate target c_o^p, if it can not find it corresponding target, or $\delta_{c_o^p} = t_i^{p-1}$ but $\theta_{t_i^{p-1}} \neq o_c^p$, then it will be regarded as the potential target. And if it has become the potential target continuously for a long time, it will be regarded as a new target and put into T^p.

5.3.2.5 Robot moving control and human-robot communicating

The system should control the robot moving towards the target once it is been detected for a while. The power of movement for the robot is provided by the robot cleaner iRobot. Once it received the command from the system that a person that is interested in the robot appeared, it get the position of this target person, and keep moving towards to him/her until the distance to the target is less than a threshold.

It is also important to have a good user interface for communicating between users and robot. Travelers in the airport should get the necessary information as quick as possible once they consult to the robot.

Figure 5.19 shows an example for the navigation service. When the iPad showing the facial expression, press the button and then talk to it. Currently the system can recognize English, Japanese and Chinese languages. After talked by using one of these three languages, the menu for the same language will be shown to the user. And user start to chose the service. For security, the users have to scan their boarding pass before get service. And after then the service will be shown to the user, and the robot will be ready to provide the next service.

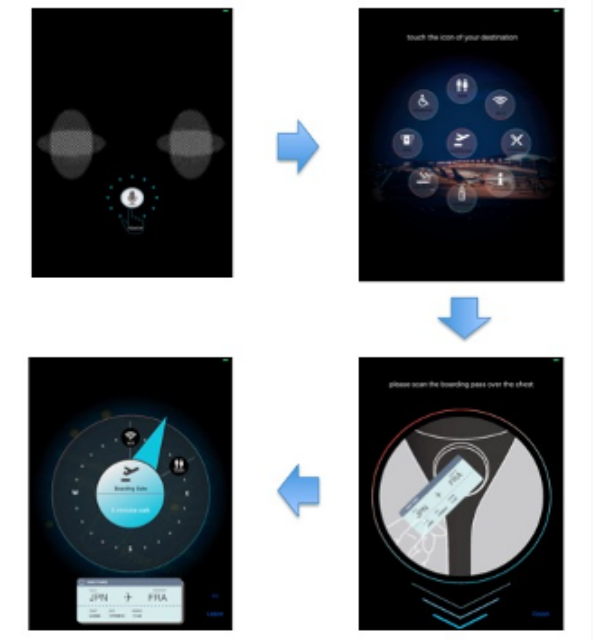


Figure 5.19: *Appearance of interface of the robot.*

5.3.3 Experiment result

5.3.3.1 Target detection

In order to make a comparison between different cases, we categorized the whole scene into case of single and multiple persons. For single case, the experiment shows a good performance. We tested the experiment with several different radius for nearest neighbor,

It is obvious to get the conclusion from the table I that for single case, the accuracy rate would be higher if the size of the radius becoming larger. This is because in single case, we do not need to consider the influence from other targets during clustering. And the error of under estimated might caused in the part of foreground abstraction.

To make a comparison with single case, we also made the experiment with the case that have two people in the scene. According to the distance between these two people, we divide it into 3 sub cases: near, middle and far. And the result shows in table II, III, IV.

It is much more complicating when there are two or more people appears in the scene at the same time. Radius size is not the only fact that affects the result anymore, and the distance between each target also has a strong for detecting. Different from the single case that larger radius has the more accuracy rate, for different distance between targets, and different radius would be the proper size. Thus the ideal radius size is no longer a fixed value any more.

Table 5.5: *Accuracy rate for single person with different radius.*

radius size	50	100	200	400
accuracy rate	86%	98%	99%	100%
over estimated rate	14%	0	0	0
under estimated rate	0	2%	1%	0

Table 5.6: *Accuracy rate for double person with the large distances between them.*

radius size	50	100	200	400
accuracy rate	83%	98%	99%	100%
over estimated rate	17%	1%	0	0
under estimated rate	0	1%	1%	0

Table 5.7: *Accuracy rate for double person with the middle distance between them.*

radius size	50	100	200	400
accuracy rate	86%	100%	98%	89%
over estimated rate	14%	0	0	0
under estimated rate	0	0	2%	11%

Table 5.8: *Accuracy rate for double person with the short distance between them*

radius size	50	100	200	400
accuracy rate	89%	95%	61%	0%
over estimated rate	11%	0	0	0
under estimated rate	0	5%	39%	100%

5.3.3.2 People tracking

Currently we introduce position as the only feature for people tracking. Therefore the radius is also significant in this stage.

Usually the travels pass through the screen by 5 to 10 frames according to the frames per second. And if there is a traveler that spends more time in the screen, he/she will be regarded as interested in this robot or having some problems. We made the experiment result of different time trigger.

In order to show the effect of the threshold value d_t, we made the experiment and the result can be seen in fig 6. In this case, the whole flow is divided into 3 status: none; people detected and interested people detected. We can see from the figure that smaller values for d_t will make the trigger for defining the human of interest earlier than usual whereas the larger values give the opposite affect. Therefore in this case, $d_t = 7$ is the suitable selection.

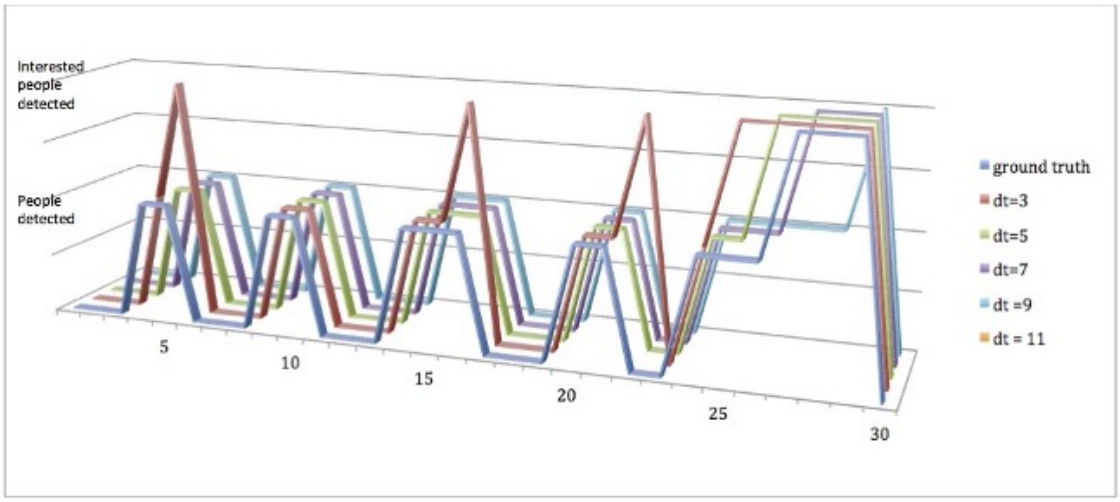


Figure 5.20: *Comparison for different value of threshold d_t.*

5.3.4 DiscuSssion

We constructed a robot navigating system for airport and other public areas and detecting interested pedestrian by heuristic and clustering algorithms. It shows a ideal performance on several cases, and the computation cost keeps in a low level so that this system can be applied into the platform which does not have high quality of the hardware device.

Currently the result is largely depending on the parameters setting such as the radius in nearest neighbor, and we set these parameters only by the experience. In the future we would focus on the generation of the self-adaptive parameters to reduce the affect from parameters setting as low as possible.

Nevertheless, we only considered the position of the target as the feature until now. It is not sufficient in some special cases. Later we will introduce other features to improve its robustness.

5.4 Summary

The tracking algorithms have been developed more and more precisely, especially since the development of deep learning algorithms. Nevertheless, these advanced algorithms improved the requirements for hardware and devices at the same time.

The strong relationship between physical exercise and cognitive performance especially in elderly people is shown in the current society, and physical exercise system is also getting more and more concentration.

Based on this situation, in this chapter, we proposed a dynamic posture evaluating and

matching framework which is constructed by DTW-based Steady State Genetic Algorithm based forward kinematics for rotational angle prediction, and Dynamic Time Warping for dynamic posture matching. Comparing with standard SSGA, DTW-based SSGA perform much better when handling the complected postures or longer time interval.

This make it possible that the system can be constructed by a simple device such smart phone and other mobile smart devices. In this paper, we only consider the evaluation of arms for exercise. In the future, we will also take into consideration of evaluation of legs, head and torso.

Despite these advantages, there are still some issues that we need to solve. First of all, the accuracy of the prediction is not as high as systems that calculate from three dimensional input data directly. In the future, we would focus on this issue by selecting more suitable algorithms if possible.

Another issue is that there is still some problems for estimating the real data because of the deviation between real values and estimated values. In the future we would also intend to deal with the problems.

Chapter 6

Conclusions

The strong relationship between physical exercise and cognitive performance especially in elderly people is shown in the current society, and physical exercise system is also getting more and more concentration. The exertainment has been promoted as a way to improve users' health through exercise, but few studies have been undertaken. In this dissertation, I focus on building up physical exercise systems for elderly people, which is fast and convenience.

I first proposed an unsupervised human posture recognition method that is different from most of the previous proposed methods. The proposed method contains a series of unsupervised learning algorithms, such as GNG, and PSO. Thus, no pre-training data is required, which is crucial for real world applications. By applying GNG, it deduces the run time cost dramatically compared with tackling the whole point cloud directly. In addition, the PSO made it possible to find the best simulated posture without any training.

However, considering the situation that depth cameras are quite rare for ordinary families, I proposed a monocular camera based dynamic posture estimation and matching framework which is constructed by DTW-based Steady State Genetic Algorithm based forward kinematics for rotational angle prediction, and Dynamic Time Warping for dynamic posture matching. Comparing with standard SSGA, DTW-based SSGA perform much better when handling the complected postures or longer time interval.

In the last step, we utilize these solutions into real implementations, which shows the fast and convenience and can be applied into inexpensive smart devices easily. In these implementations, elderly people are able to exercise with entertainment, which will improve the motivation of them for exercise comparing with the standard exercises.

References

[1] L. J. Baraff, R. Della Penna, N. Williams, and A. Sanders, "Practice guideline for the ed management of falls in community-dwelling elderly persons," Annals of emergency medicine, vol. 30, no. 4, pp. 480–492, 1997.

[2] D. L. Murman, "The impact of age on cognition," in Seminars in hearing, vol. 36, no. 3. Thieme Medical Publishers, 2015, p. 111.

[3] K. Wada and T. Shibata, "Living with seal robots—its sociopsychological and physiological influences on the elderly at a care house," IEEE transactions on robotics, vol. 23, no. 5, pp. 972–980, 2007.

[4] N. Kubotal and Y. Shimomura, "Human-friendly networked partner robots toward sophisticated services for a community," in 2006 SICE-ICASE International Joint Conference. IEEE, 2006, pp. 4861–4866.

[5] A. Lotfi, C. Langensiepen, and S. W. Yahaya, "Socially assistive robotics: Robot exercise trainer for older adults," Technologies, vol. 6, no. 1, p. 32, 2018.

[6] S. W. Yahaya, A. Lotfi, and M. Mahmud, "A framework for anomaly detection in activities of daily living using an assistive robot," 2019.

[7] D. Marr, "Early processing of visual information," Philosophical Transactions of the Royal Society of London. B, Biological Sciences, vol. 275, no. 942, pp. 483–519, 1976.

[8] S. J. Prince, Computer vision: models, learning, and inference. Cambridge University Press, 2012.

[9] D. Dumitrescu, B. Lazzerini, L. C. Jain, and A. Dumitrescu, Evolutionary computation. CRC press, 2000.

[10] L. J. Fogel, A. J. Owens, and M. J. Walsh, "Artificial intelligence through simulated evolution," 1966.

[11] K. A. De Jong, "Analysis of the behavior of a class of genetic adaptive systems," Tech. Rep., 1975.

[12] A. E. Eiben, J. E. Smith et al., Introduction to evolutionary computing. Springer, 2003, vol. 53.

[13] J. H. Holland et al., Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT press, 1992.

[14] Y. Zhang, S. Wang, and G. Ji, "A comprehensive survey on particle swarm optimization algorithm and its applications," Mathematical Problems in Engineering, vol. 2015, 2015.

[15] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in 1997 IEEE International conference on systems, man, and cybernetics. Computational cybernetics and simulation, vol. 5. IEEE, 1997, pp. 4104–4108.

[16] J. Kennedy and R. Eberhart, "Particle swarm optimization," in Proceedings of ICNN'95-international conference on neural networks, vol. 4. IEEE, 1995, pp. 1942–1948.

[17] A. Kleinsmith and N. Bianchi-Berthouze, "Affective body expression perception and recognition: A survey," IEEE Transactions on Affective Computing, vol. 4, no. 1, pp. 15–33, 2012.

[18] M. Hayase and S. Shimada, "Posture estimation of human body based on connection relations of 3d ellipsoidal models," JOURNAL OF ADVANCED COMPUTATIONAL INTELLIGENCE AND INTELLIGENT INFORMATICS, vol. 14, no. 6, pp. 638–644, 2010.

[19] J. Suarez and R. R. Murphy, "Hand gesture recognition with depth images: A review," in 2012 IEEE RO-MAN: the 21st IEEE international symposium on robot and human interactive communication. IEEE, 2012, pp. 411–417.

[20] J. K. Aggarwal and M. S. Ryoo, "Human activity analysis: A review," ACM Computing Surveys (CSUR), vol. 43, no. 3, pp. 1–43, 2011.

[21] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in CVPR 2011. Ieee, 2011, pp. 1297–1304.

[22] L. A. Schwarz, A. Mkhitaryan, D. Mateus, and N. Navab, "Human skeleton tracking from depth data using geodesic distances and optical flow," Image and Vision Computing, vol. 30, no. 3, pp. 217–226, 2012.

[23] M. Stommel, M. Beetz, and W. Xu, "Model-free detection, encoding, retrieval, and visualization of human poses from kinect data," IEEE/ASME Transactions on Mechatronics, vol. 20, no. 2, pp. 865–875, 2014.

[24] Y. Tang, H. A. Vu, P. Q. Le, D. Masano, O. Thet, C. Fatichah, Z. Liu, M. Yamaguchi, M. L. Tangel, F. Dong et al., "Multimodal gesture recognition for mascot robot system based on choquet integral using camera and 3d accelerometers fusion." JACIII, vol. 15, no. 5, pp. 563–572, 2011.

[25] J.-J. Cabibihan, W.-C. So, and S. Pramanik, "Human-recognizable robotic gestures," IEEE Transactions on Autonomous Mental Development, vol. 4, no. 4, pp. 305–314, 2012.

[26] T. Fong, I. Nourbakhsh, and K. Dautenhahn, "A survey of socially interactive robots," Robotics and autonomous systems, vol. 42, no. 3-4, pp. 143–166, 2003.

[27] Y. Tang, H. A. Vu, P. Q. Le, D. Masano, O. Thet, C. Fatichah, Z. Liu, M. Yamaguchi, M. L. Tangel, F. Dong et al., "Multimodal gesture recognition for mascot robot system based on choquet integral using camera and 3d accelerometers fusion." JACIII, vol. 15, no. 5, pp. 563–572, 2011.

[28] Y. Takahashi, K. Yoshida, F. Hibino, and Y. Maeda, "Human pointing navigation interface for mobile robot with spherical vision system," 2011.

[29] E. E. Stone and M. Skubic, "Fall detection in homes of older adults using the microsoft kinect," IEEE journal of biomedical and health informatics, vol. 19, no. 1, pp. 290–301, 2014.

[30] W. Song, M. Minami, and Y. Mae, "Evolutionary head pose measurement by improved stereo model matching," in 2006 SICE-ICASE International Joint Conference. IEEE, 2006, pp. 4234–4239.

[31] M. Ye, Q. Zhang, L. Wang, J. Zhu, R. Yang, and J. Gall, "A survey on human motion analysis from depth data," in Time-of-flight and depth imaging. sensors, algorithms, and applications. Springer, 2013, pp. 149–187.

[32] C. Lee, H. Song, B. Choi, and Y.-S. Ho, "3d scene capturing using stereoscopic cameras and a time-of-flight camera," IEEE Transactions on Consumer Electronics, vol. 57, no. 3, pp. 1370–1376, 2011.

[33] J. Jung, J.-Y. Lee, Y. Jeong, and I. S. Kweon, "Time-of-flight sensor calibration for a color and depth camera pair," IEEE transactions on pattern analysis and machine intelligence, vol. 37, no. 7, pp. 1501–1513, 2014.

[34] S. Foix, G. Alenya, and C. Torras, "Lock-in time-of-flight (tof) cameras: A survey," IEEE Sensors Journal, vol. 11, no. 9, pp. 1917–1926, 2011.

[35] R. S. Hartenberg and J. Denavit, "A kinematic notation for lower pair mechanisms based on matrices.(1955)," URL: https://datenpdf.com/download/a-kinematic-notation-for-lower-pair-mechanisms-basedon_pdf (visited on 14/04/2019), 1955.

[36] R. Hartenberg and J. Danavit, Kinematic synthesis of linkages. New York: McGraw-Hill, 1964.

[37] N. J. Kirk-Sanchez and E. L. McGough, "Physical exercise and cognitive performance in the elderly: current perspectives," Clinical interventions in aging, vol. 9, p. 51, 2014.

[38] A. Kleinsmith and N. Bianchi-Berthouze, "Affective body expression perception and recognition: A survey," IEEE Transactions on Affective Computing, vol. 4, no. 1, pp. 15–33, 2012.

[39] M. Hayase and S. Shimada, "Posture estimation of human body based on connection relations of 3d ellipsoidal models," JOURNAL OF ADVANCED COMPUTATIONAL INTELLIGENCE AND INTELLIGENT INFORMATICS, vol. 14, no. 6, pp. 638–644, 2010.

[40] J. Suarez and R. R. Murphy, "Hand gesture recognition with depth images: A review," in 2012 IEEE RO-MAN: the 21st IEEE international symposium on robot and human interactive communication. IEEE, 2012, pp. 411–417.

[41] J. K. Aggarwal and M. S. Ryoo, "Human activity analysis: A review," ACM Computing Surveys (CSUR), vol. 43, no. 3, pp. 1–43, 2011.

[42] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in CVPR 2011. Ieee, 2011, pp. 1297–1304.

[43] L. A. Schwarz, A. Mkhitaryan, D. Mateus, and N. Navab, "Human skeleton tracking from depth data using geodesic distances and optical flow," Image and Vision Computing, vol. 30, no. 3, pp. 217–226, 2012.

[44] M. Stommel, M. Beetz, and W. Xu, "Model-free detection, encoding, retrieval, and visualization of human poses from kinect data," IEEE/ASME Transactions on Mechatronics, vol. 20, no. 2, pp. 865–875, 2014.

[45] M. Mitchell and C. E. Taylor, "Evolutionary computation: an overview," Annual Review of Ecology and Systematics, vol. 30, no. 1, pp. 593–616, 1999.

[46] A. N. Sloss and S. Gustafson, "2019 evolutionary algorithms review," Genetic Programming Theory and Practice XVII, pp. 307–344, 2020.

[47] P. G. Espejo, S. Ventura, and F. Herrera, "A survey on the application of genetic programming to classification," IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 40, no. 2, pp. 121–144, 2009.

[48] P. J. Fleming and R. C. Purshouse, "Evolutionary algorithms in control systems engineering: a survey," Control engineering practice, vol. 10, no. 11, pp. 1223–1241, 2002.

[49] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," Swarm and Evolutionary Computation, vol. 1, no. 1, pp. 32–49, 2011.

[50] A. Lotfi and J. M. Garibaldi, Applications and science in soft computing. Springer Science & Business Media, 2013, vol. 24.

[51] S. Starke, N. Hendrich, D. Krupke, and J. Zhang, "Evolutionary multi-objective inverse kinematics on highly articulated and humanoid robots," in 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2017, pp. 6959–6966.

[52] S. Starke, N. Hendrich, and J. Zhang, "Memetic evolution for generic full-body inverse kinematics in robotics and animation," IEEE Transactions on Evolutionary Computation, vol. 23, no. 3, pp. 406–420, 2018.

[53] Y. Chen, Y. Tian, and M. He, "Monocular human pose estimation: A survey of deep learning-based methods," Computer Vision and Image Understanding, vol. 192, p. 102897, 2020.

[54] A. Kendall, M. Grimes, and R. Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," in Proceedings of the IEEE international conference on computer vision, 2015, pp. 2938–2946.

[55] Y. Chen, C. Shen, X.-S. Wei, L. Liu, and J. Yang, "Adversarial posenet: A structure-aware convolutional network for human pose estimation," in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1212–1221.

[56] W. Quan, J. Woo, Y. Toda, and N. Kubota, "Human posture recognition for estimation of human body condition," Journal of Advanced Computational Intelligence and Intelligent Informatics, vol. 23, no. 3, pp. 519–527, 2019.

[57] R. S. Hartenberg and J. Denavit, "A kinematic notation for lower pair mechanisms based on matrices.(1955)," URL: https://datenpdf.com/download/a-kinematic-notation-for-lower-pair-mechanisms-basedon_pdf (visited on 14/04/2019), 1955.

[58] E. C. Kinzel, J. P. Schmiedeler, and G. R. Pennock, "Kinematic synthesis for finitely separated positions using geometric constraint programming," 2006.

[59] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series." in KDD workshop, vol. 10, no. 16. Seattle, WA, USA:, 1994, pp. 359–370.

[60] J. Kennedy and R. Eberhart, "Particle swarm optimization," in Proceedings of ICNN'95-international conference on neural networks, vol. 4. IEEE, 1995, pp. 1942–1948.

[61] N. J. Kirk-Sanchez and E. L. McGough, "Physical exercise and cognitive performance in the elderly: current perspectives," Clinical interventions in aging, vol. 9, p. 51, 2014.

[62] R. Poppe, "A survey on vision-based human action recognition," Image and vision computing, vol. 28, no. 6, pp. 976–990, 2010.

[63] M. Eichner and V. Ferrari, "Human pose co-estimation and applications," IEEE transactions on pattern analysis and machine intelligence, vol. 34, no. 11, pp. 2282–2288, 2012.

[64] T. B. Moeslund and E. Granum, "A survey of computer vision-based human motion capture," Computer vision and image understanding, vol. 81, no. 3, pp. 231–268, 2001.

[65] H. Jiang, "Human pose estimation using consistent max covering," IEEE transactions on pattern analysis and machine intelligence, vol. 33, no. 9, pp. 1911–1918, 2011.

[66] L. L. Presti and M. La Cascia, "3d skeleton-based human action classification: A survey," Pattern Recognition, vol. 53, pp. 130–147, 2016.

[67] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in CVPR 2011. Ieee, 2011, pp. 1297–1304.

[68] V. Belagiannis, S. Amin, M. Andriluka, B. Schiele, N. Navab, and S. Ilic, "3d pictorial structures for multiple human pose estimation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1669–1676.

[69] Y. Chen, Y. Tian, and M. He, "Monocular human pose estimation: A survey of deep learning-based methods," Computer Vision and Image Understanding, vol. 192, p. 102897, 2020.

[70] A. Kendall, M. Grimes, and R. Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," in Proceedings of the IEEE international conference on computer vision, 2015, pp. 2938–2946.

[71] Y. Chen, C. Shen, X.-S. Wei, L. Liu, and J. Yang, "Adversarial posenet: A structure-aware convolutional network for human pose estimation," in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1212–1221.

[72] D. Metaxas and D. Terzopoulos, "Shape and nonrigid motion estimation through physics-based synthesis," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 15, no. 6, pp. 580–591, 1993.

[73] W. Quan, J. Woo, Y. Toda, and N. Kubota, "Human posture recognition for estimation of human body condition," Journal of Advanced Computational Intelligence and Intelligent Informatics, vol. 23, no. 3, pp. 519–527, 2019.

[74] E. C. Kinzel, J. P. Schmiedeler, and G. R. Pennock, "Kinematic synthesis for finitely separated positions using geometric constraint programming," 2006.

[75] D. Metaxas and D. Terzopoulos, "Shape and nonrigid motion estimation through physics-based synthesis," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 15, no. 6, pp. 580–591, 1993.

[76] C. Lee, Y. Chang, G. Park, J. Ryu, S.-G. Jeong, S. Park, J. W. Park, H. C. Lee, K.-s. Hong, and M. H. Lee, "Indoor positioning system based on incident angles of infrared emitters," in 30th Annual Conference of IEEE Industrial Electronics Society, 2004. IECON 2004, vol. 3. IEEE, 2004, pp. 2218–2222.

[77] A. W. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," IEEE transactions on pattern analysis and machine intelligence, vol. 36, no. 7, pp. 1442–1468, 2013.

[78] D. B. Fogel, Evolutionary computation: toward a new philosophy of machine intelligence. John Wiley & Sons, 2006, vol. 1.

[79] N. Kubota and I. A. Sulistijono, "Evolutionary robot vision for people tracking based on local clustering," in 2008 World Automation Congress. IEEE, 2008, pp. 1–6.

[80] A. Yorita and N. Kubota, "Multi-stage fuzzy evaluation in evolutionary robot vision for face detection," Evolutionary intelligence, vol. 3, no. 2, pp. 67–78, 2010.

[81] W. Quan and N. Kubota, "Evolutionary people tracking for robot partner of information service in public areas," in International Conference on Intelligent Robotics and Applications. Springer, 2017, pp. 703–714.

[82] G. Syswerda, "A study of reproduction in generational and steady-state genetic algorithms," in Foundations of genetic algorithms. Elsevier, 1991, vol. 1, pp. 94–101.

[83] D. Thierens and D. Goldberg, "Elitist recombination: An integrated selection recombination ga," in Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence. IEEE, 1994, pp. 508–512.

[84] J. H. Holland et al., Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT press, 1992.

Acknowledgement

Firstly, I would like to express my sincere gratitude to my advisor Prof. Naoyuki Kubota for the continuous support of my PhD study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my PhD study.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Naoyuki Takesue, Prof. Takao Fukui, and Prof. Chee Seng Chan, for their insightful comments and encouragement, but also for the hard question which incented me to widen my research from various perspectives.

I thank my fellow lab mates in for the stimulating discussions, for the sleepless nights we were working together before deadlines, and for all the fun we have had in the passed years.

In the end, I am grateful to my parents, friends and acquaintances who remembered me in their prayers for the ultimate success. I consider myself nothing without them. They gave me enough moral support, encouragement and motivation to accomplish the personal goals. My parents have always supported me financially so that I only pay attention to the studies and achieving my objective without any obstacle on the way.