

A Study of General-Purpose Parameter Optimization Algorithm in Three Diverse Applications

September, 2023

Huo Yuanzhi

Graduate School of
Natural Science and Technology

(Doctor's Course)
OKAYAMA UNIVERSITY

Dissertation submitted to
Graduate School of Natural Science and Technology
of
Okayama University
for
partial fulfillment of the requirements
for the degree of
Doctor of Philosophy.

Written under the supervision of

Professor Nobuo Funabiki

and co-supervised by

Professor Satoshi Denno

and

Professor Yasuyuki Nogami

OKAYAMA UNIVERSITY, September 2023.

TO WHOM IT MAY CONCERN

We hereby certify that this is a typical copy of the original doctor thesis of
Huo Yuanzhi

Signature of
the Supervisor

Seal of

Prof. Nobuo Funabiki

Graduate School of
Natural Science and Technology

Abstract

Parameter optimization is crucial in a lot of scientific researches, as it can greatly impact the performance and the accuracy of the implemented model and system. The process of a parameter optimization can be divided into several stages of defining the parameters, selecting an optimization method, defining a search space, evaluating the performance, and refining the parameter values. This process is often repeated many times. A parameter optimization algorithm should be adopted to automate the above-mentioned process and improve the result. A parameter optimization algorithm can be evaluated in efficiency, accuracy, robustness and generalizability.

Previously, we have proposed a general-purpose parameter optimization algorithm called *paraOpt*. *paraOpt* was designed on metaheuristic algorithms used for solving optimization problems. This algorithm starts with an initial solution that is generated randomly or is sequentially constructed based on heuristics. A neighborhood of the current solution is defined by generating a set of candidate solutions that are obtained by making small modifications to the current solution. Each candidate solution is evaluated using the objective function, and the best neighborhood is selected as the new current solution.

In this thesis, I study the application of this general-purpose parameter optimization algorithm *paraOpt* in three diverse applications.

The first application is the fingerprint-based indoor localization system using IEEE 802.15.4 devices called *FILS15.4* that can detect the location of a user in an indoor environment. The IEEE 802.15.4 standard defines the short-range and low-speed data communication using narrow-band and low-power wireless signals at 2.4 GHz frequencies. The transmitter can be small, light, and inexpensive, and can work for almost one year with a coin battery. The receiver can be small, portable, inexpensive, and equipped with USB connectivity. These features make IEEE 802.15.4 devices ideal for indoor location systems. The fingerprint-based approach is simple but dependable for indoor localization using wireless signals. The use of *FILS15.4* consists of two phases. In the calibration phase, the pattern of the typical received signal strength or *link quality indicator (LQI)* for every room is collected as the fingerprint by locating the transmitter there. In the detection phase, the location of the transmitter is continuously found by comparing the current LQI pattern with every fingerprint and selecting the closest one.

However, due to the narrow channel band and the low transmission power, LQI used for fingerprints easily fluctuates by human movements and other uncontrollable factors. To improve the localization accuracy, *FILS15.4* restricts the detection granularity to one room in the field, and adopts multiple fingerprints for one room, considering fluctuated signals. *paraOpt* is applied to optimize the parameter values of the multiple fingerprints. The experiment results show after the optimization the detection accuracy exceeds 98% in Okayama Engineering Building #2.

The second application is the human face contour approximation model that is described by a combination of half circles, line segments, and a quadratic curve. To approximate this model from the face image, *OpenPose* is used to extract the human face contour. *OpenPose* is the popular

open software that can jointly detect the coordinates of the *keypoints* in the human body, hands, face, and foot from a single image. A *keypoint* represents a feature point in them such as a joint, a fingertip, and a nostril.

However, *OpenPose* cannot extract the contour of the upper part of the face including the forehead due to the hair. For solving this limitation, we have proposed a simple face contour approximation model that consists of two half circles and line segments. The upper half circle will draw the forehead and the lower half circle will draw the chin. The two line segments that connect the ends of the two half circles will draw the edges of the face contour. *paraOpt* is applied to optimize the parameters of this model, including the center coordinates and the radiuses of the half circles, so that the model is well-matched with the *keypoints* by *OpenPose*. The experiment results show that the average face contour Euclidean distance difference of *OpenPose* from face contour approximation model is reduced to 17.98.

The third application is the computational fluid dynamic (*CFD*) simulation to estimate temperature changes in a room. A small-sized model room for experiments is assembled for evaluating the accuracy of the *CFD* simulation. In this model room, temperature-controlled air using an air conditioning unit is supplied. At the bottom of the model room, heaters are mounted to raise the temperature. To measure the temperature distribution of the room, 27 temperature sensors are installed at equal intervals in the room. To estimate or predict the distributions in a room together with sensors, the *CFD* simulation using *OpenFOAM* software. *paraOpt* is applied to optimize the parameters of this model, to fit well the simulation results with the corresponding measured ones. The experiment results show the thermal conductivity is optimized to make the average temperature difference between the estimated and measured 0.06°C .

In future works, I will improve the parameter optimization algorithm and evaluate it in other applications.

Acknowledgements

I would like to take this opportunity to express my deepest gratitude and appreciation to all the individuals that have contributed to the completion of my doctoral thesis. Their support, guidance, and encouragement have played a significant role in shaping my thesis.

First and foremost, I am indebted to my supervisor and advisor, Professor Nobuo Funabiki, for his unwavering dedication, invaluable expertise, and continuous guidance throughout the entire research process. His deep knowledge of the field and insightful feedback have been instrumental in shaping the direction and quality of this thesis and insightful suggestions and constructive criticism have greatly contributed to the improvement of my research work. I am truly grateful for his mentorship and support.

I am indebted to my PhD co-supervisors, Professor Satoshi Denno and Professor Yasuyuki Nogami, for taking their valuable time to give me advice, guidance, insightful comments, and proofreading of this thesis. Their valuable advice, expertise and knowledge in the field have undoubtedly shaped my understanding and expanded my academic horizons.

Then, I am deeply grateful to the participants who generously shared their time, experiences, and expertise for the purpose of this study. Especially to Associate Professor Minoru Kuribayashi and Associate Professor Kazuyuki Kojima who belong to Shonan Institute of Technology University for the valuable discussions during this work. Also I want to appreciate to Ms. Keiko Kawabata for the administrative support. Ms. Pradini Puspitaningayu, Mr. Kazushi Hamazaki, Mr. Yohanes Panduman, Mr Chenrui Shi who often worked together with me. Though it is difficult to mention every name, none of the enjoyable moments we shared would ever slip from my memory.

Over and above, to all of my friends who shared fate with me as students in Okayama, away from our far home in China. Having emotional support and continuous assurance from you gave me strength to a level I never knew I could have. It was truly a blessing to know all of you. I am also indebted to my family for their unwavering support, encouragement, and understanding throughout this challenging journey. Their love, patience, and belief in my abilities have been a constant source of motivation.

Last but not least, my utmost gratitude to everyone who gave me their unending and unconditional love. You are the reason I am here, and the reason I am coming home.

Huo Yuanzhi
Okayama, Japan
September 2023

List of Publications

Journal Papers

1. **Y. Huo**, P. Puspitaningayu, N. Funabiki, K. Hamazaki, M. Kuribayashi, and K. Kojima, “A proposal of the fingerprint optimization method for the fingerprint-based indoor localization system with IEEE 802.15.4 devices,” *Information*, vol. 13, no. 5, p. 211, 2022.
2. **Y. Huo**, P. Puspitaningayu, N. Funabiki, K. Hamazaki, M. Kuribayashi, Y. Zhao, and K. Kojima, “Three diverse applications of general-purpose parameter optimization algorithm,” *Algorithms*, vol. 16, no. 1, p. 45, 2023.

International Conference Papers

3. **Y. Huo**, P. Puspitaningayu, N. Funabiki, K. Hamazaki, M. Kuribayashi, and K. Kojima, “A parameter optimization method for fingerprint-based indoor localization system using IEEE 802.15.4 devices,” 2021 3rd International Conference on Computer Communication and the Internet (ICCCI), 2021.
4. **Y. Huo**, N. Funabiki, X. Xu, K. Kojima, Y. Zhao, and W. Kao, “Temperature Estimation Accuracy Improvement of Computational Fluid Dynamic Simulation by Optimizing Multiple Parameters,” 4th International Conference on Information Technology and Education Technology (ITET 2023), 2023.

Other Paper

5. **Y. Huo**, P. Puspitaningayu, K. Hamazaki, N. Funabiki, M. Kuribayashi, and K. Kojima, “A Study of Parameter Optimization Method for Fingerprint-based Indoor Localization System Using IEEE 802.15.4 Devices,” IEICE Life Intelligence and Office Information Systems Study Group (LOIS), 2021.

List of Figures

3.1	<i>FILS15.4</i> application overview.	11
3.2	Experiment field layout for fluctuations causes.	12
3.3	Measured <i>LQI</i> data for <i>D307-2</i>	12
3.4	Fluctuation <i>LQI</i> data at <i>0.5m</i>	15
3.5	Fluctuation <i>LQI</i> data at <i>1m</i>	16
3.6	Fluctuation <i>LQI</i> data at <i>1.5m</i>	17
3.7	Fluctuation <i>LQI</i> data at <i>1.8m</i>	18
3.8	Experiment field layout.	19
3.9	Detection accuracy with interval variation.	20
3.10	Number of fingerprints with interval variation.	20
3.11	Detection accuracy at 40 s interval.	21
3.12	Number of fingerprints at 40 s interval.	21
4.1	Face contour model and <i>Openpose</i> keypoints.	24
4.2	Face contour model generation procedure.	25
4.3	Example image and face contour.	25
4.4	Alternative model for chin.	26
4.5	images example.	27
4.6	Face contour results after optimization.	28
4.7	Score comparison between two models after optimization.	28
5.1	Model room for experiments.	31
5.2	Model–room for <i>CFD</i> simulation.	31
5.3	<i>OpenFOAM</i> simulation results.	32
5.4	Simulation result after optimization with <i>pattern1</i>	33
5.5	Simulation result after optimization with <i>pattern2</i>	34
5.6	Temperature results by four parameters optimization with constant value for training.	36
5.7	Temperature results by four parameters optimization with constant value for validation.	36
5.8	Temperature results by four parameters optimization with changed value for training.	37
5.9	Temperature results by four parameters optimization with changed value for validation.	37

List of Tables

3.1	Experiment scenarios for <i>LQI</i> fluctuation causes.	14
3.2	Fluctuation <i>LQI</i> data summary and detection accuracy at $0.5m$	14
3.3	Fluctuation <i>LQI</i> data summary and detection accuracy at $1m$	14
3.4	Fluctuation <i>LQI</i> data summary and detection accuracy at $1.5m$	15
3.5	Fluctuation <i>LQI</i> data summary and detection accuracy at $1.8m$	16
4.1	Parameter optimization results.	27
5.1	Boundary conditions.	32
5.2	Parameters values before proposal.	33
5.3	Parameters for boundary conditions.	34
5.4	Initial parameter values for <i>paraOpt</i>	35
5.5	Results by four parameters optimization with constant value.	36
5.6	Results by four parameters optimization with changed value.	36

Contents

Abstract	i
Acknowledgements	iii
List of Publications	v
List of Figures	vii
List of Tables	ix
1 Introduction	1
1.1 Background	1
1.2 Contribution	2
1.3 Contents of This Dissertation	3
2 Review of Parameter Optimization Algorithm	5
2.1 Symbols	5
2.2 Algorithm Procedure	5
2.2.1 Initialization Phase	6
2.2.2 Optimization Phase	6
2.3 Summary	7
3 Application to Fingerprint-Based Indoor Localization System	9
3.1 Background	9
3.2 Technologies	9
3.2.1 IEEE 802.15.4	9
3.2.2 Fingerprinting	10
3.2.3 Message-Queueing Telemetry Transport (MQTT)	10
3.3 Implemented System	10
3.4 Localization Logic and Parameters	10
3.5 Signal Fluctuation Problem	12
3.5.1 LQI Observations	13
3.5.2 Evaluation with Fluctuation Causes	13
3.6 Parameter Optimization Algorithm Application	13
3.7 Evaluations	19
3.7.1 Optimization of Detection Interval	19
3.7.2 Optimization of Fingerprints	20
3.7.3 Discussion	21

3.8	Summary	22
4	Application to Face Contour Approximation Model	23
4.1	Background	23
4.2	Supporting Technologies	23
4.2.1	Openpose	23
4.3	Proposed Model and Parameters	24
4.4	Model Generation Procedure	24
4.5	Initial Parameter Values	25
4.6	Example Initial Model	25
4.7	Alternative Model	26
4.8	Score Function	26
4.9	Evaluations	26
4.9.1	Face Images	27
4.9.2	Optimization Results	27
4.10	Summary	28
5	Application to CFD Simulation	29
5.1	Background	29
5.2	Technologies	29
5.2.1	CFD	29
5.2.2	OpenFOAM	30
5.2.3	Heat Flux	30
5.3	Model Room for Experiments	30
5.4	CFD Simulation Model and Parameters	31
5.5	Score Function	32
5.6	One Parameter Evaluations	33
5.6.1	Experiment Setup	33
5.6.2	Optimization Results	33
5.7	Four Parameters Evaluations	34
5.7.1	Experiment Setup	34
5.7.2	Optimization Results	35
5.8	Summary	38
6	Discussions	39
6.1	Performances in Three Applications	39
6.2	Complexities of Three Applications	39
6.3	Parametrizations in Three Applications	39
6.4	Summary	40
7	Related Works in Literature	41
8	Conclusion	45
	References	47

Chapter 1

Introduction

1.1 Background

Parameter optimization is a crucial task in a lot of scientific researches, since it may greatly impact the performance and the accuracy of the implemented model or system. The process of a parameter optimization can be divided into several stages. It defines the parameters, selects an optimization method, defines a search space, evaluates the performance, and refines the parameter values. This process is often repeated many times. A parameter optimization algorithm should be adopted to automate the above-mentioned process and improve the result. A parameter optimization algorithm can be evaluated in efficiency, accuracy, robustness and generalizability.

Currently, deep learning approaches such as the convolutional neural network (CNN) [1] and the recurrent neural network (RNN) [2] become popular. Although these approaches may give powerful solutions to some problems such as pattern recognitions and classifications, they are basically *black-box* approaches. The users cannot validate the correctness of the model and cannot modify it when mistakes or errors occur. Therefore, the *white-box* approach of optimizing the parameter values in the comprehensible logical model is essential for practical use.

With the advent of computers, parameter optimizations for models, algorithms, and logics have become important parts of computer-aided design activities. There are two distinct types of optimization algorithms widely used today, deterministic algorithms and stochastic algorithms. Both algorithms have been successfully applied to many problems. Deterministic algorithms use specific rules for moving one solution to another. Stochastic algorithms are in nature using probabilistic translation rules and may have many good advantages. Constraints are important in parameter optimizations. They represent functional relationships among the parameters that are described to satisfy certain physical phenomena or resource limitations.

Previously, we have proposed a general-purpose parameter optimization algorithm called *paraOpt*. *paraOpt* was designed on metaheuristic algorithms used for solving optimization problems. This algorithm starts with an initial solution that is generated randomly or is sequentially constructed based on heuristics. A neighborhood of the current solution is defined by generating a set of candidate solutions that are obtained by making small modifications to the current solution. Each candidate solution is evaluated using the objective function, and the best neighborhood is selected as the new current solution.

1.2 Contribution

In this thesis, I study the applications of the general-purpose parameter optimization algorithm *paraOpt* in three diverse applications.

The first application of *paraOpt* is the fingerprint-based indoor localization system using IEEE 802.15.4 devices called *FILS15.4* [3] that can detect the location of a user in an indoor environment. The IEEE 802.15.4 standard defines the short-range and low-speed data communication using narrow-band and low-power wireless signals at 2.4 GHz frequencies. The transmitter can be small, light, and inexpensive, and can work for almost one year with a coin battery. The receiver can be small, portable, inexpensive, and equipped with USB connectivity. These features make IEEE 802.15.4 devices ideal for indoor location systems. The fingerprint-based approach is simple but dependable for indoor localization using wireless signals. The use of *FILS15.4* consists of two phases. In the calibration phase, the pattern of the typical received signal strength or *link quality indicator (LQI)* for every room is collected as the fingerprint by locating the transmitter there. In the detection phase, the location of the transmitter is continuously found by comparing the current LQI pattern with every fingerprint and selecting the closest one.

However, due to the narrow channel band and the low transmission power, LQI used for fingerprints easily fluctuates by human movements and other uncontrollable factors. I conducted the experiments to evaluate the effects of them. They including the door open/close, Wi-Fi signals, human movements, transmitter directions and heights. All these factors can make LQI fluctuations.

To improve the localization accuracy, *FILS15.4* restricts the detection granularity to one room in the field, and adopts multiple fingerprints for one room, considering fluctuated signals. *paraOpt* is applied to optimize the parameter values of the multiple fingerprints. The experiment results show after the optimization the detection accuracy exceeds 98% in Okayama Engineering Building #2.

The second application of *paraOpt* is the human face contour approximation model that is described by a combination of half circles, line segments, and a quadratic curve. To approximate this model from the face image, *OpenPose* is used to extract the human face contour [4]. *OpenPose* is the popular open software that can jointly detect the coordinates of the *keypoints* in the human body, hands, face, and foot from a single image. A *keypoint* represents a feature point in them such as a joint, a fingertip, and a nostril.

However, *OpenPose* cannot extract the contour of the upper part of the face including the forehead due to the hair. For solving this limitation, we proposed a simple face contour approximation model that consists of two half circles and line segments. The upper half circle will draw the forehead and the lower half circle will draw the chin. The two line segments that connect the ends of the two half circles will draw the edges of the face contour.

paraOpt is applied to optimize the parameters of this model, including the center coordinates and the radiuses of the half circles, so that the model is well-matched with the *keypoints* by *OpenPose*. The experiment results show that the average face contour Euclidean distance difference of *OpenPose* from face contour approximation model is reduced to 17.98.

The third application of *paraOpt* is the computational fluid dynamic (CFD) simulation to estimate temperature changes in a room [5]. A small-sized model room for experiments is assembled for evaluating the accuracy of the CFD simulation. In this model room, temperature-controlled air using an air conditioning unit is supplied. At the bottom of the model room, heaters are mounted to raise the temperature. To measure the temperature distribution of the room, 27 temperature sensors are installed at equal intervals in the room.

To estimate or predict the temperature distributions in a room together with sensors, the CFD

simulation using *OpenFOAM* software. *paraOpt* is applied to optimize the parameters of this model, to fit well the simulation results with the corresponding measured ones. The experiment results show the thermal conductivity is optimized to make the average temperature difference between the estimated and measured 0.06°C .

1.3 Contents of This Dissertation

The remaining part of this thesis is organized as follows. Chapter 2 reviews the parameter optimization algorithm. Chapter 3 presents the *Fingerprint-Based Indoor Localization System application*. Chapter 4 presents the *Face Contour Approximation Model application*. Chapter 5 presents the *CFD Simulation model*. Chapter 6 discusses the results for three applications. Chapter 7 reviews relevant works in literature. Finally, Chapter 8 concludes this thesis with some future works.

Chapter 2

Review of Parameter Optimization Algorithm

This chapter reviews the parameter optimization algorithm *paraOpt* [6].

2.1 Symbols

First, define the symbols to describe the procedure of the parameter optimization algorithm. Among them, p_i^{init} , Δp_i , t_i , and $S(P)$ should be properly selected for the target algorithm/logic to achieve better result.

- P : the set of the n parameters for the algorithm/logic in the logic program whose values should be optimized.
- p_i : the value of the i th parameter in P ($1 \leq i \leq n$).
- p_i^{init} : the initial value of the i th parameter in P ($1 \leq i \leq n$).
- Δp_i : the change step for p_i .
- t_i : the tabu period for p_i in the tabu table.
- $S(P)$: the score of the algorithm/logic using P .
- P_{best} : the best set of the parameters.
- $S(P_{best})$: the score of the algorithm/logic where P_{best} is used.
- L : the log of the generated parameter values and their scores.

2.2 Algorithm Procedure

The following procedure describes the steps of the parameter optimization algorithm to find the parameter values of P to minimize the score $S(P)$:

2.2.1 Initialization Phase

First, the algorithm variables are initialized:

- (1) Clear the generated parameter log L .
- (2) Set the initial value in the parameter file for any p_i in P , set 0 for any tabu period t_i , and set a large value for $S(P_{best})$.

2.2.2 Optimization Phase

Then, the parameters are optimized iteratively:

- (3) Generate the neighborhood parameter value sets for P by:

- (a) Randomly selecting one parameter p_i for $t_i = 0$.
- (b) Calculate the parameter values of p_i^- and p_i^+ by:

$$\begin{aligned} p_i^- &= p_i - \Delta p_i, \text{ if } p_i > \text{lower limit,} \\ p_i^+ &= p_i + \Delta p_i, \text{ if } p_i < \text{upper limit.} \end{aligned} \quad (2.1)$$

- (c) Generate the neighborhood parameter value sets P^- and P^+ by replacing p_i by p_i^- or p_i^+ :

$$\begin{aligned} P^- &= \{p_1, p_2, \dots, p_i^-, \dots, p_n\} \\ P^+ &= \{p_1, p_2, \dots, p_i^+, \dots, p_n\} \end{aligned}$$

- (4) When P (P^- , P^+) exists in L , obtain $S(P)$ ($S(P^-)$, $S(P^+)$) from L . Otherwise, execute the logic program using P (P^- , P^+) to obtain $S(P)$ ($S(P^-)$, $S(P^+)$), and write P and $S(P)$ (P^- and $S(P^-)$, P^+ and $S(P^+)$) into L .
- (5) Compare $S(P)$, $S(P^-)$, and $S(P^+)$, and select the parameter value set that has the largest one among them.
- (6) Update the tabu period by:
 - (a) Decrement t_i by -1 if $t_i > 0$.
 - (b) Set the given constant tabu period TB for t_i if $S(P)$ is the largest at (5) and p_i is selected at (3)(a).
- (7) When $S(P)$ is continuously largest at (5) for the given constant times, go to (8). Otherwise, go to (3).
- (8) When the hill-climbing procedure in (9) is applied for the given constant times HT , go to (10) as the state is converged. Otherwise, go to (9).
- (9) Apply the hill-climbing procedure:
 - (a) If $S(P) < S(P_{best})$, update P_{best} and $S(P_{best})$ by P and $S(P)$.
 - (b) Reset P by P_{best} .
 - (c) Randomly select p_i in P , and randomly change the value of p_i within its range and go to (3).
- (10) Terminate the algorithm.

2.3 Summary

This chapter reviews the parameter optimization algorithm *paraOpt* proposed in our previous research and introduces the details of *paraOpt*. In the next chapter, I will present the *Fingerprint-Based Indoor Localization System application*.

Chapter 3

Application to Fingerprint-Based Indoor Localization System

3.1 Background

Various localization techniques have been applied in indoor and outdoor environments. In outdoor environments, the *global positioning system (GPS)* is available. However, it cannot cover indoor environments [7, 8]. Then, to successfully cover indoor environments, several wireless technologies have been explored for *indoor localization systems*.

Fingerprinting has obtained great interests due to the reasonable accuracy capability by adopting the *radio map pattern matching* [9]. Each location in the target field is assumed to have its own unique radio pattern called the *fingerprint*. The *fingerprint* value should be different from the one for other locations.

This method consists of the *calibration phase* and the *detection phase*. The *calibration phase* collects the *radio signal map* and generates the *fingerprint* for every location in the field, and stores it in the database. The *detection phase* compares the received radio signal with every *fingerprint* and selects the closest one as the current location. When considerable calibration efforts are made, this method can achieve robust detection capabilities [10].

Based on this method, we are currently studying the *fingerprint-based indoor localization system* using the *IEEE 802.15.4* protocol, called *FILS15.4* [3, 11]. *FILS15.4* uses the *IEEE 802.15.4* devices in *Mono Wireless* [12]. The small, light-weight transmitter is suitable for use to be worn by a user. It is inexpensive (USD 30), is small (2.5 mm × 2.5 mm), is light (0.93 g), and can work with a coin battery for a long time. The radio signal from the transmitter will be received at multiple receivers allocated in the field, and the *LQI (link quality indicator)* vector is compared with the *fingerprint* for each location.

3.2 Technologies

3.2.1 IEEE 802.15.4

The *IEEE 802.15.4* protocol defines the low-rate wireless network. The communication range is shorter than *IEEE 802.11* but longer than *Bluetooth*. The device is small and inexpensive, and will consume low energy, which makes possible use of a coin battery for a long time. *ZigBee* implements this protocol and has gained popularity because of its low-power, low-range, and low-

data transmission features. In [13], a ZigBee-based indoor localization system was proposed with the *radial basis function network (RBFN)* to determine the location with the fingerprinting method. In [14], the *nearest neighbor* and *Bayesian* were adopted, which promised less than or equal to the 0.81m accuracy. We believe that the *IEEE 802.15.4* protocol is a good candidate for indoor localization systems.

3.2.2 Fingerprinting

The *fingerprinting* method has gained the most attention due to its ability to achieve reasonable accuracy [9]. It adopts the *radio map pattern matching* where every location should have a unique signal pattern considering the transmitter and the receiver's location. In the *calibration phase*, the *radio map* in the target field is made by measuring the *RSS* [15]. In the *detection phase*, the current signal strengths are compared with every fingerprint in the *radio map*, and the closest one is selected as the current position [16].

3.2.3 Message-Queueing Telemetry Transport (MQTT)

The *message-queueing telemetry transport (MQTT)* protocol is one of the well-known transport protocols for device-to-device communications in *Internet of Things (IoT)* systems. It works with the publish/subscribe principle, where each device takes the role of the publisher, the subscriber, or both at transmitting data. In the middle of both sides, a broker acts to relay it.

3.3 Implemented System

Figure 3.1 illustrates the overview of *FILS15.4*. During location detections, the user always needs to wear the transmitter. The transmitter will send data with the 500 ms interval. The receivers allocated in the field will receive the data with *LQI*, and send them to the server through the *USB*-connected *Raspberry Pi* with the 30s interval, utilizing the *MQTT* protocol. The server detects the user's current room by comparing the received *LQI* with every stored fingerprint.

FILS15.4 adopts *Twelite 2525* in *Mono Wireless* [12] as the transmitter conforming the *IEEE 802.15.4* standard. The wireless signal is at the 2.4 GHz band, which can be interfered with *IEEE 802.11 Wi-Fi*. During detections, the user may wear it at the wrist.

Furthermore, *FILS15.4* adopts *Mono Stick* in the same company as the receiver. It is connected with *Raspberry Pi* through the *USB* interface. When a packet from a transmitter is received, the *link quality indication (LQI)* is also monitored. *Raspberry Pi* sends the received and *LQI* data to the server through the *MQTT protocol* [17].

In the *calibration phase*, the server stores the received data in the *SQLite* database, calculates the average *LQI* during every 30s, and combines the values from all the receivers into one vector to generate the fingerprint for the room. It is stored with the relevant location label. In the *detection phase*, the server calculates the Euclidean distance between the average measured *LQI* and every fingerprint to detect the current room at every 30s.

3.4 Localization Logic and Parameters

As the initial parameter values, one is used for the initial value of the number of fingerprints, and the average of all the measured *LQI* data at a receiver from a transmitter located in the target room

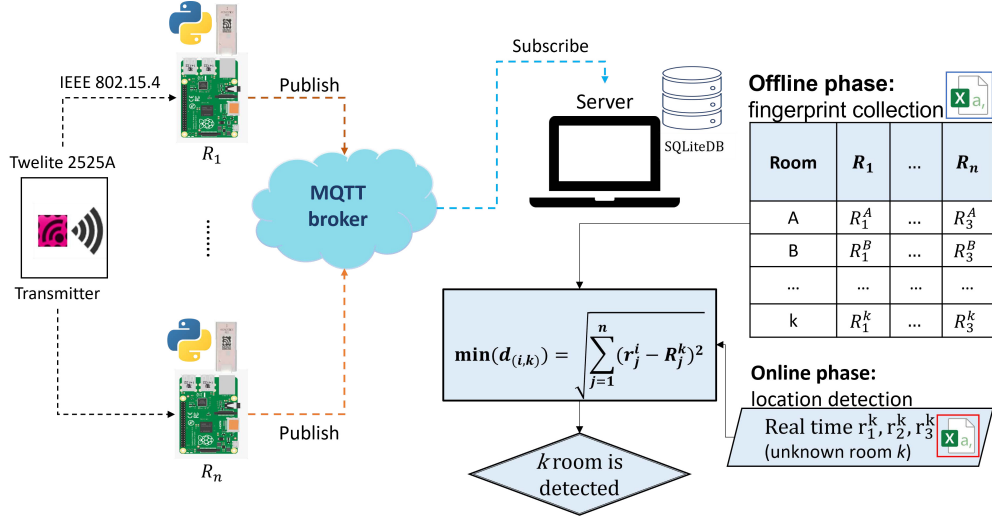


Figure 3.1: *FILS15.4* application overview.

is used for the initial value of the corresponding fingerprint value.

- (1) Properly locate the *Raspberry Pi* devices with the receivers in the target field.
- (2) Run the programs and create the connection to the *MQTT* broker.
- (3) Locate the transmitter at the specified location in the field. In our experiments, we selected several locations where we moved the transmitter from one place to another after measuring *LQI* for one minute by transmitting packets every 500 *ms*.
- (4) Receive and collect the packets from the transmitter at the *Raspberry Pi* device for 30 s.
- (5) Forward the collected data from the *Raspberry Pi* device to the server through the *MQTT* broker.
- (6) For each receiver, calculate the average *LQI* using the forwarded data from it after the last average *LQI* calculation.
- (7) Make the fingerprints at the server and store them in the *SQLite* database.

In the *detection phase*, the server detects the current room of the user by applying steps (1)–(6) in the procedure for the *calibration phase* periodically. Then, in step (7), after the vector of the average *LQI* values from all the receivers are obtained, the Euclidean distance is calculated against every pre-stored fingerprint by Equation (3.1), and the room whose fingerprint has the smallest distance is appointed as the detected room.

$$disF_i^k = \sqrt{\sum_{j=1}^n (r_j^i - R_j^k)^2} \quad (3.1)$$

where

- $disF_i^k$ represents the Euclidean distance between the i -th measured average *LQI* and the fingerprint for room k ;

- r_j^i does the i -th measured average LQI at receiver j ; and
- R_j^k does a fingerprint for room k at receiver j .

3.5 Signal Fluctuation Problem

In our preliminary experiments, we collected LQI data for one hour using five receivers on the third floor of the no. 2 Engineering Building at Okayama University in Figure 3.2 and observed the signal fluctuation problem.

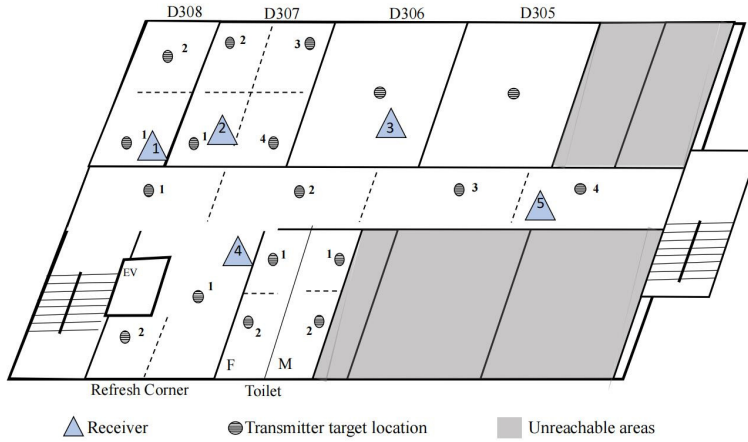


Figure 3.2: Experiment field layout for fluctuations causes.

Figure 3.3 shows the measured LQI data at the five receivers, $LQI1$ - $LQI5$, when the transmitter was located at $D307$ -2. Any data always fluctuated. Sometimes no data was received at the four receivers except $LQI2$ due to the *connection loss*, where $LQI = 5$ indicates no data reception. It could be caused by human movements in the field, where someone in the field blocked the signal path, or closed the door of the room.

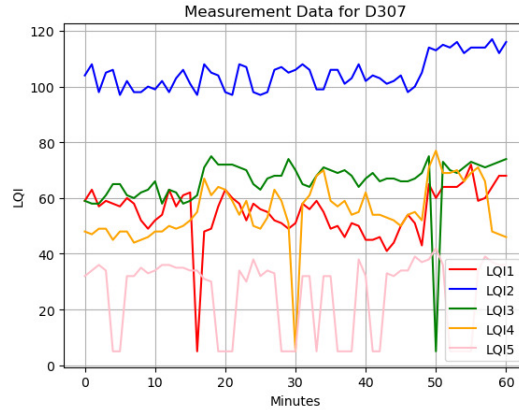


Figure 3.3: Measured LQI data for $D307$ -2.

3.5.1 LQI Observations

Let us discuss observations of each *LQI* data in Figure 3.3.

- At *LQI2*, which comes from the nearest receiver from the transmitter, no *connection loss* appeared, and two different *LQI* levels can be observed.
- At *LQI1*, *LQI3*, and *LQI4*, one *connection loss* appeared, and two-three different *LQI* levels can be observed.
- At *LQI5*, *connection loss* often appeared, whereas the *LQI* level is almost constant.

3.5.2 Evaluation with Fluctuation Causes

We listed the six causes for *LQI* data fluctuations and conducted the experiments to evaluate the effects by them using the scenarios in Table 3.1. During the experiments, the transmitter was located at *D307-4* in Figure 3.2. For *door open/close*, the door of *D307* was opened and closed. For *Wi-Fi*, the Wi-Fi interface of a smartphone was turned on and off in *D307*. For *human movement*, one, two, or three persons moved around in *D307*. For *transmitter direction*, the face of the transmitter was directed to eastward, westward, northward, southward, upward, and downward directions. For *transmitter movement*, the transmitter location was moved in the five rooms. For *transmitter height*, the height of the transmitter from the floor was changed.

We show the results by each transmitter height as follows. Figures 3.4–3.7 show the measured *LQI* data, when the transmitter height was 0.5 m, 1 m, 1.5 m, and 1.8 m, respectively. Tables 3.2–3.5 summarize the average and standard deviation (SD) of the *LQI* data and the room detection accuracy of the proposed *FILS15.4* for each transmitter height.

These results indicate that the measured *LQI* data are frequently fluctuating at any case of the six fluctuation causes. Nevertheless, the room detection accuracy of *FILS15.4* is sufficiently high for any case of the fluctuation causes when the transmitter location is fixed (no *transmitter movement*). Even for *transmitter movement*, the accuracy reached 94% when the transmitter height was 1.8 m. As the transmitter height increases, the obstacles between the transmitter and the receivers are reduced. Thus, stronger and more stable signals can be detected at the receivers, which reduces the *LQI* data fluctuations and improves the detection accuracy.

To accomplish the high detection accuracy by solving the signal fluctuation problem on *IEEE 802.15.4* devices, we limit the detection granularity of *FILS15.4* to one room in the field. Furthermore, we make multiple fingerprints with distinct values for each room. As a result, the optimization of the number of fingerprints and their values for each room becomes an important issue in determining the detection accuracy of *FILS15.4*, which will be very difficult to achieve manually.

3.6 Parameter Optimization Algorithm Application

FILS15.4 has several parameters whose values should be optimized. The following procedure describes the calculation of the score $S(P)$:

- (1) Calculate the Euclidean distance $disF_i^k$ between the i -th average measured *LQI* and the k -th current fingerprint.
- (2) Find $disF^{OK}$ that represents the minimum Euclidean distance against a fingerprint representing the correct room.

Table 3.1: Experiment scenarios for LQI fluctuation causes.

fluctuation cause	experiment scenario
door open/close	<ul style="list-style-type: none"> • 0-20 min: open • 20-40 min: close • 40-60 min: frequently open/close
Wi-Fi on/off	<ul style="list-style-type: none"> • 0-30 min: on • 30-35 min: off • 35-60 min: on
human movement	<ul style="list-style-type: none"> • 0-20 min: three persons • 20-40 min: two persons • 40-60 min: one person
transmitter direction	<ul style="list-style-type: none"> • 0-10 min: east • 10-20 min: west • 20-30 min: north • 30-40 min: south • 40-50 min: up • 50-60 min: down
transmitter movement	<ul style="list-style-type: none"> • 0-10 min: D306 • 10-20 min: Refresh Corner • 20-30 min: D307 • 30-40 min: Corridor • 40-50 min: D308
transmitter height	<ul style="list-style-type: none"> • 0.5m • 1m • 1.5m • 1.8m

Table 3.2: Fluctuation LQI data summary and detection accuracy at 0.5m.

fluctuation cause	value	LQI1	LQI2	LQI3	LQI4	LQI5	accuracy
door open/close	AVE	59.24	141.24	48.71	85.29	40.46	96.4%
	SD	10.89	26.71	18.93	15.33	7.1	
Wi-Fi on/off	AVE	57.56	130.85	52.14	93.3	49.03	96.7%
	SD	14.88	23.96	9.19	17.2	11.22	
human movement	AVE	64.41	134.01	60.54	56.05	21.59	96.2%
	SD	17.46	31.67	3.68	8.22	13.56	
transmitter direction	AVE	79.63	128.18	65	71.87	37.31	96.6%
	SD	13.42	28.63	12.69	11.76	16.73	
transmitter movement	AVE	38.36	49.09	46.71	64.82	47.96	82%
	SD	43.16	35.58	27.04	47.5	30.65	

Table 3.3: Fluctuation LQI data summary and detection accuracy at 1m.

fluctuation cause	value	LQI1	LQI2	LQI3	LQI4	LQI5	accuracy
door open/close	AVE	90.3	146.41	53.39	64.67	40.44	100%
	SD	2.97	1.6	5.59	14.71	6.89	
Wi-Fi on/off	AVE	97.27	157.27	67.05	75.73	24.3	100%
	SD	1.31	0.67	1.87	2.0	10.09	
human movement	AVE	64.89	137.53	61.0	76.21	36.36	100%
	SD	2.38	1.45	1.88	1.12	1.13	
transmitter direction	AVE	78.5	131.35	48.13	73.16	38.88	100%
	SD	14.84	6.25	17.02	10.27	15.48	
transmitter movement	AVE	71.55	66.34	54.53	57.31	43.18	82%
	SD	39.86	30.54	32.21	27.81	29.29	

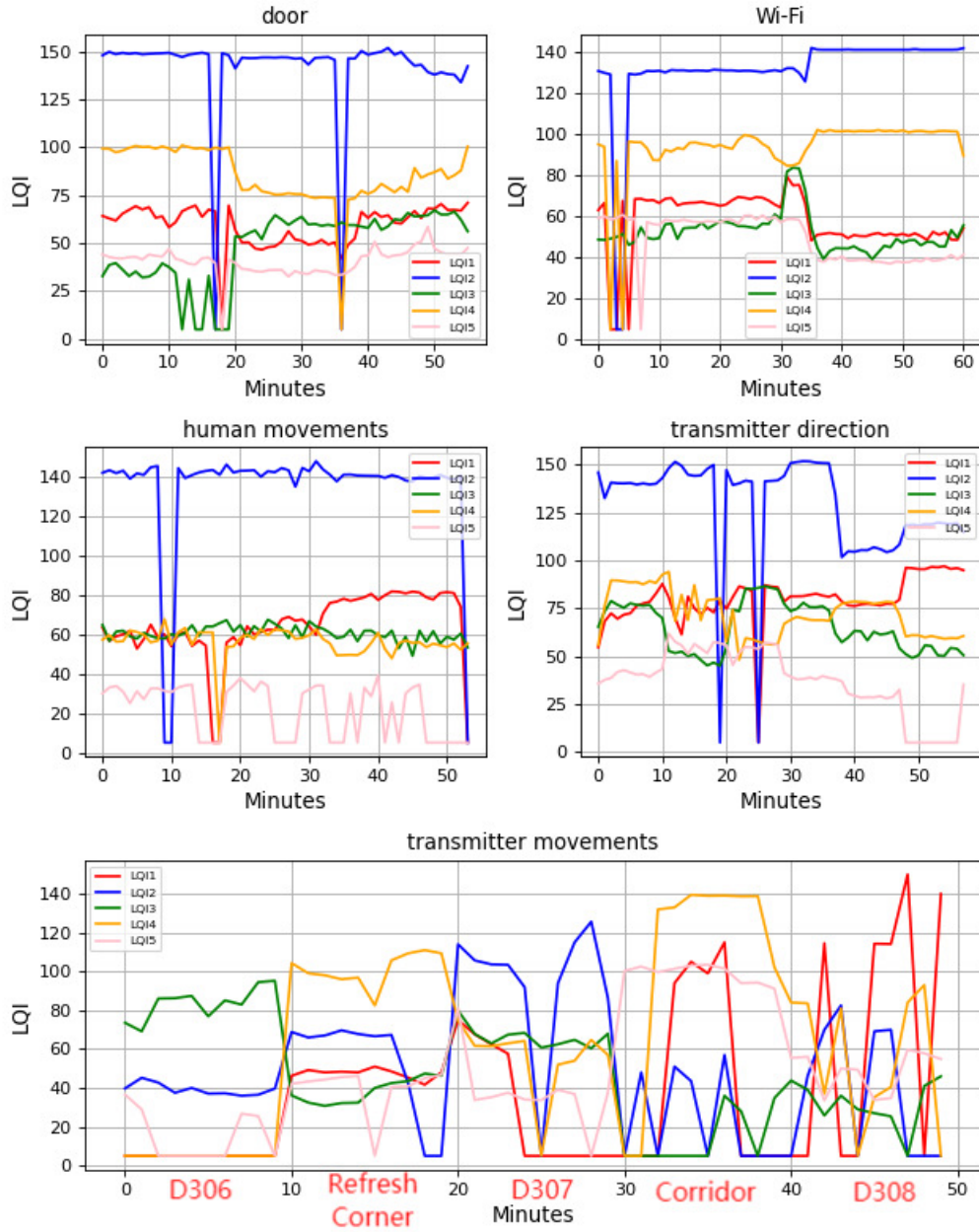


Figure 3.4: Fluctuation LQI data at $0.5m$.

Table 3.4: Fluctuation LQI data summary and detection accuracy at $1.5m$.

fluctuation cause	value	LQI1	LQI2	LQI3	LQI4	LQI5	accuracy
door open/close	AVE	85.63	126.69	50.25	70.37	28.8	93.1%
	SD	25.01	33.53	8.43	23.51	10.63	
Wi-Fi on/off	AVE	85.21	136.65	70.4	69.62	47.8	96.8%
	SD	14.85	24.23	2.8	9.5	1.06	
human movement	AVE	72.89	122.55	73.04	48.04	7.79	98.1%
	SD	9.51	16.38	1.25	9.0	8.06	
transmitter direction	AVE	78.11	116.35	68.27	64.08	42.19	94.7%
	SD	24.08	30.07	20.79	12.87	9.53	
transmitter movement	AVE	81.74	71.03	65.3	65.25	45.46	88%
	SD	31.28	23.79	30.57	24.85	15.89	

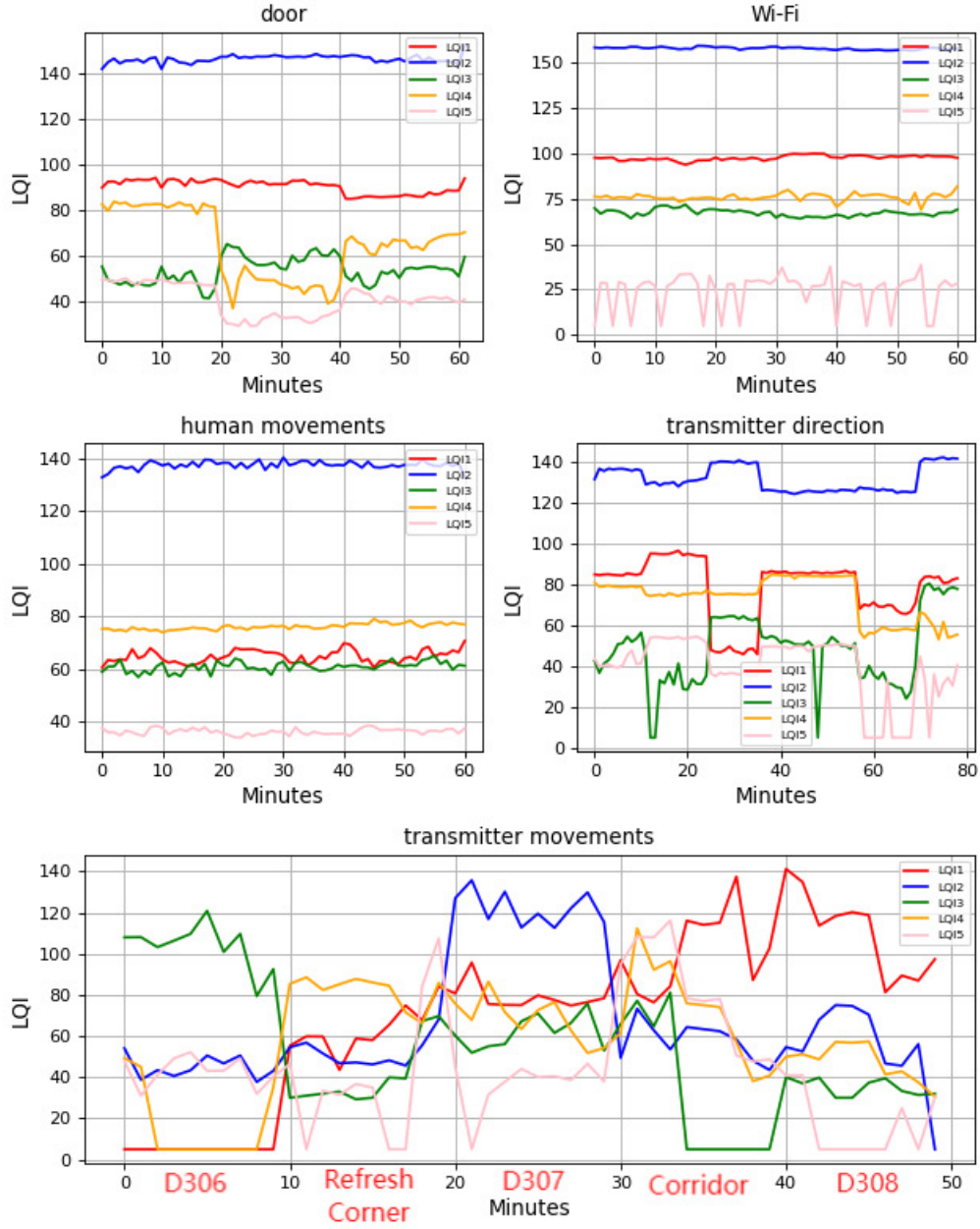


Figure 3.5: Fluctuation LQI data at 1m.

Table 3.5: Fluctuation LQI data summary and detection accuracy at 1.8m.

fluctuation cause	value	LQI1	LQI2	LQI3	LQI4	LQI5	accuracy
door open/close	AVE	72.02	150.97	74.91	90.01	51.42	100%
	SD	7.03	10.27	5.52	7.42	6.3	
Wi-Fi on/off	AVE	92.03	114.56	74.37	82.61	44.53	100%
	SD	1.02	3.2	1.79	1.08	1.45	
human movement	AVE	95.28	112.67	77.65	83.63	41.81	100%
	SD	2.73	5.19	2.39	1.23	2.82	
transmitter direction	AVE	84.09	120.46	88.66	74.95	45.62	100%
	SD	14.88	18.24	19.51	15.14	11.86	
transmitter movement	AVE	74.79	75.47	69.73	67.42	62.31	94%
	SD	26.96	25.81	32.11	27.85	24.37	

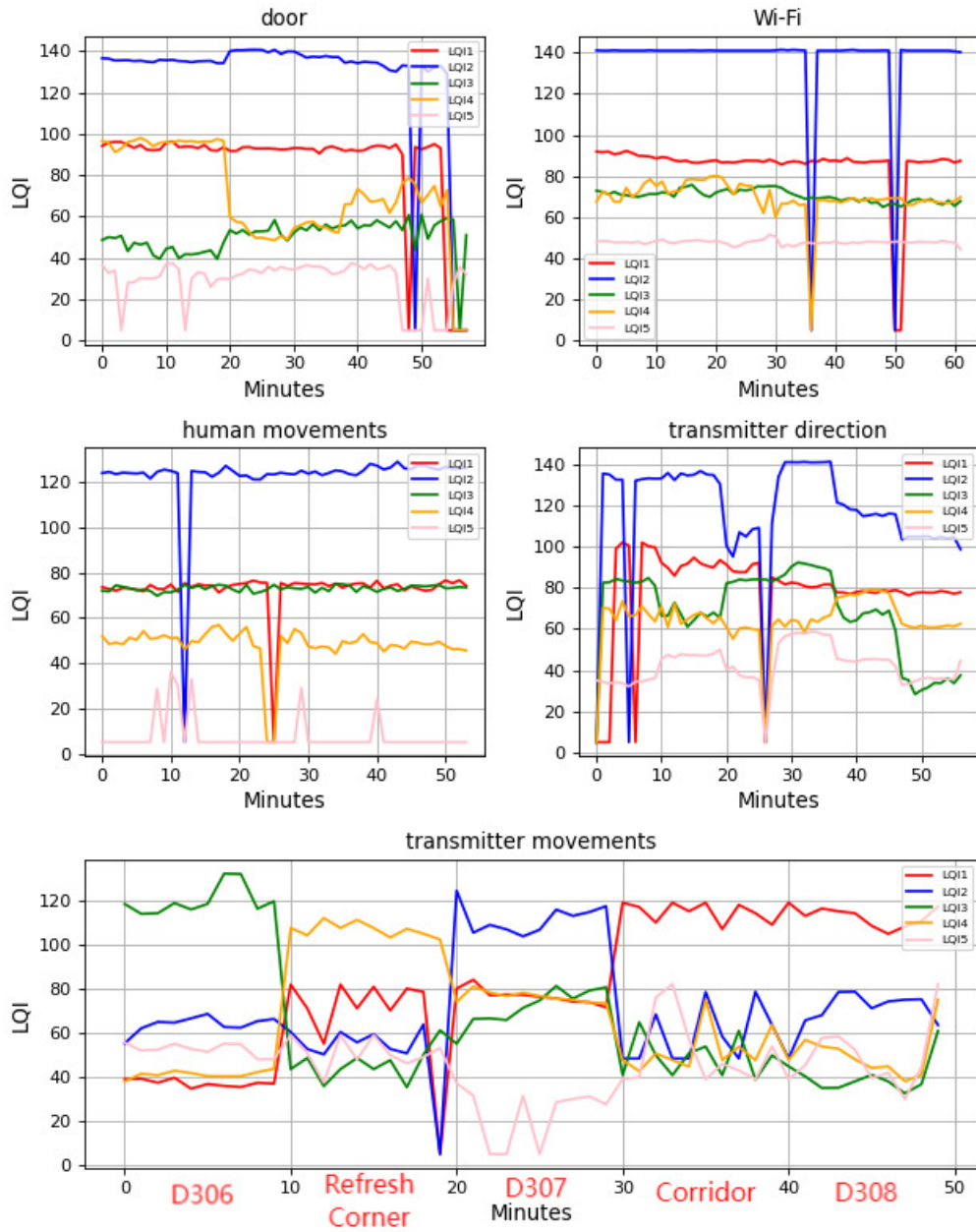


Figure 3.6: Fluctuation LQI data at $1.5m$.

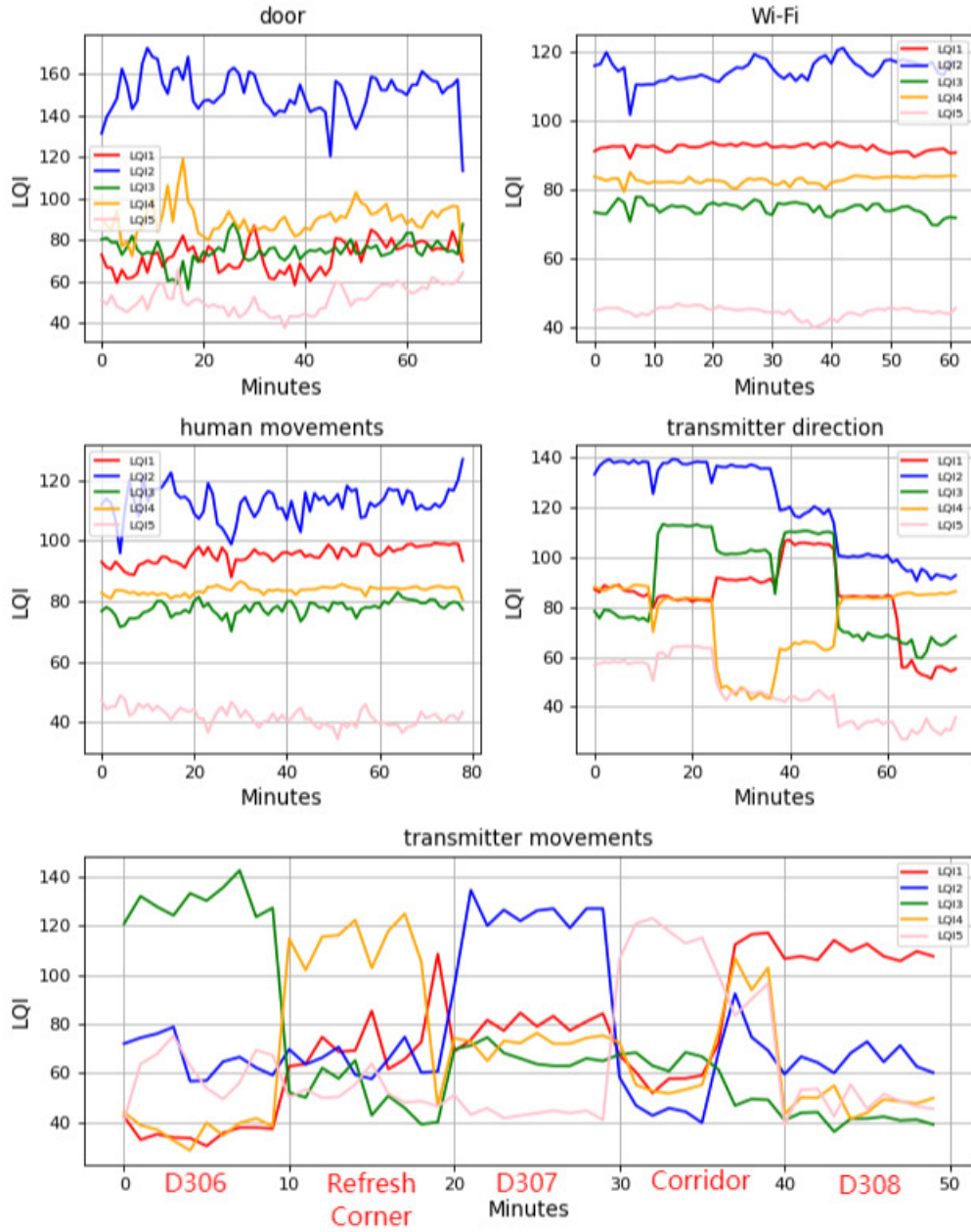


Figure 3.7: Fluctuation LQI data at $1.8m$.

- (3) Find $disF^{NG}$ that represents the minimum Euclidean distance against a fingerprint representing the incorrect room.
- (4) Calculate $S(P)$ by:

$$S(P) = A \sum_{i=1}^N true(disF^{OK} - disF^{NG}) + B \sum_{i=1}^N \frac{disF^{NG}}{disF^{OK}} + C \sum_{k=1}^M \min_{b \neq c} |F_b^k - F_c^k| \quad (3.2)$$

where A and B represent constant coefficients ($A = 10$, $B = 1$ and $C = 1$ in this paper), N is the number of the average measured LQI for the optimization, the function $true(x)$ returns 1 if $x > 0$ and 0 otherwise. The C-term represents the sum of the minimum Euclidean distance between two different fingerprints for the same room. It intends to generate different fingerprint values for the same room.

Moreover, as the important parameters in *paraOpt*, $t_i = 10$ for the tabu period, $\Delta p_i = 5$ for the detection interval, and $\Delta p_i = 1$ for the fingerprint are adopted.

3.7 Evaluations

The field layout in Figure 3.8 is used in experiments. Among the parameters in *FILS15.4*, the detection interval and the fingerprint values can most influence the detection accuracy. Thus, their optimizations are discussed.

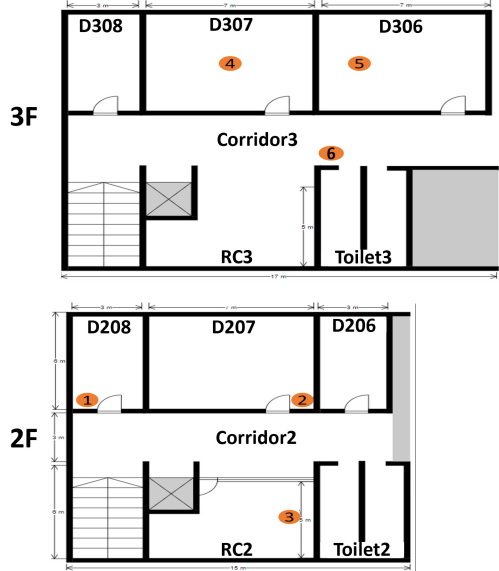


Figure 3.8: Experiment field layout.

3.7.1 Optimization of Detection Interval

First, the detection interval is optimized. Figures 3.9 and 3.10 show the detection accuracy and the number of fingerprints for each interval respectively. From 0s to 30s, both the detection and the number of fingerprints gradually increase. Then, both are saturated. The best detection accuracy is obtained when the interval is 40s, where the total number of fingerprints is 109.

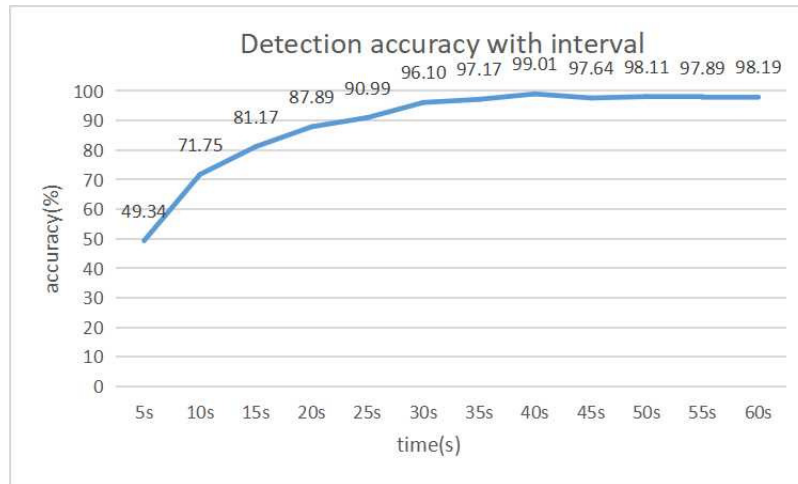


Figure 3.9: Detection accuracy with interval variation.

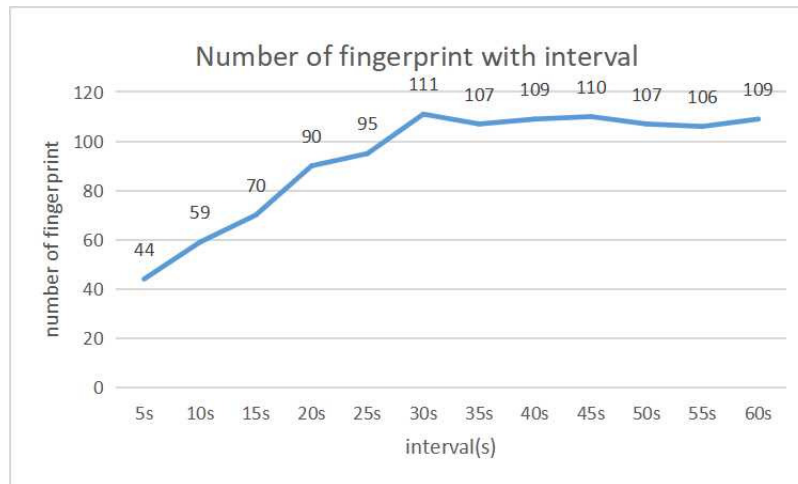


Figure 3.10: Number of fingerprints with interval variation.

3.7.2 Optimization of Fingerprints

Figures 3.11 and 3.12 show the detection accuracy and the number of fingerprints for each room respectively when the detection interval is 40s. The largest number of fingerprints is 13 for *RC2*, *D307* and *Toilet2*. The least number of fingerprints is four for *D208* and *D308*.

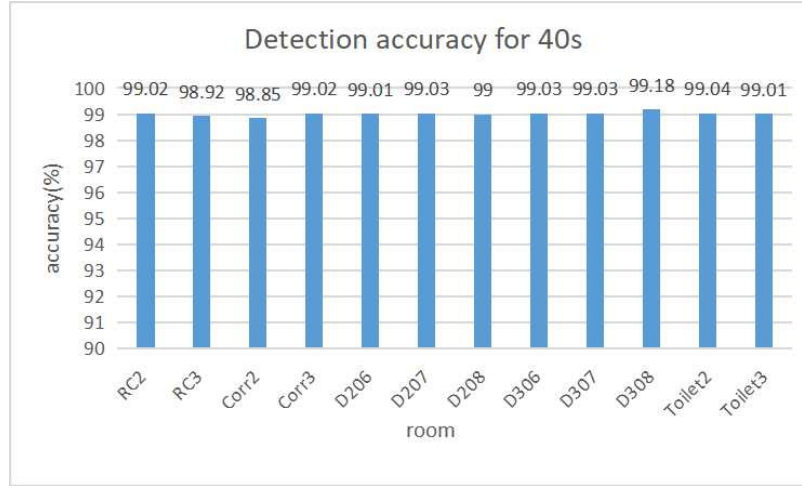


Figure 3.11: Detection accuracy at 40 s interval.

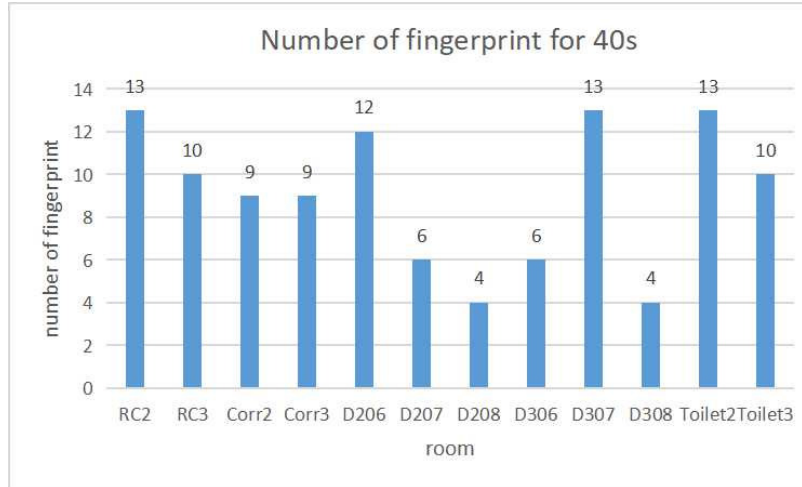


Figure 3.12: Number of fingerprints at 40 s interval.

3.7.3 Discussion

Since the measured LQI is frequently fluctuating, the moving average should be used instead of the instantaneous value to reduce misdetections. Then, the time period of the average, or the detection interval should be optimized to maximize the detection accuracy.

Then, the results in Figures 3.9 and 3.10 indicate that the detection accuracy is improved until the interval becomes 40s. After that, the accuracy is saturated, where the number of fingerprints is also saturated. Thus, 40s is selected as the best detection interval.

It is noted that the results were obtained when all the transmitters were stationary or not moving. The detection interval should be optimized when transmitters are sometimes moving in the field. However, variations of transmitter/user movements are much more diverse, including source/destination locations, moving speeds, and paces. A lot of experiments will be necessary to find the optimal interval. Thus, it will be in future works.

With the fixed detection interval of 40s, Figure 3.11 shows a sufficiently high detection accuracy of higher than 98% for any room. Figure 3.12 shows the number of fingerprints generated by

the proposal. For *D208* and *D308*, only four fingerprints are necessary and are smaller than the other rooms. The reason is that both rooms are located at the end of each floor in the two-floor field and are isolated from the other rooms. It will cause less confusion with other rooms. On the other hand, the other rooms are surrounded by several rooms and need many fingerprints to reduce confusion among them.

3.8 Summary

This chapter presents the implementation of *paraOpt* for *FILS 15.4*. Experiments were done to analyze six factors that can cause *LQI* fluctuations. Meanwhile, detection accuracy and fingerprint are optimized. Results show optimal detection interval is 40s and accuracy over 98%. In the next chapter, I will present the *Face Contour Approximation Model application*.

Chapter 4

Application to Face Contour Approximation Model

4.1 Background

Face drawing has been a longstanding and distinct art. It typically uses a sparse set of continuous graphical elements such as lines to capture the distinctive appearance of a human. It will be done in the presence of a person or his/her face image, and will rely on the holistic approach of observation, analysis, and experience [18].

The traditional technology to draw a human face contour may include four steps [19]. The first step is to draw a circle and a cross to represent the top portion of the head. The second step is to draw a square within the circle to represent the edges of the face. The third step is to draw the chin from each side of the square. The last step is to locate the hair and eyes by using lines.

For beginners, the traditional technology can be hard to learn by themselves. Therefore, an application system should be developed to assist them to learn the drawing of the face contour. A lot of technologies can help draw the face contour, including AI technology [20,21].

4.2 Supporting Technologies

4.2.1 Openpose

OpenPose is a popular computer vision library and framework that enables the real-time multi-person keypoint detection and the pose estimation from videos, images, or live camera feeds. It was developed by researchers at the Carnegie Mellon University and became open-sourced by the company CMU Perceptual Computing Lab.

OpenPose works by analyzing the input data, such as video frames or images, and processing them through a multi-stage pipeline. Initially, the framework performs the body part localization by predicting keypoint heatmaps, which represent the likelihood of each body joint's presence in the image. It then estimates the locations of individual body parts and associates them to form complete skeletal poses. The resulting pose estimation can include keypoints of various body parts, including the head, neck, shoulders, elbows, wrists, hips, knees, and ankles.

4.3 Proposed Model and Parameters

OpenPose is used to assist in drawing the human face contour by beginners. *OpenPose* is the popular open software that can jointly detect the coordinates of the *keypoints* in the human body, hands, face, and foot from a single image [4]. A *keypoint* represents a feature point in them such as a joint, a fingertip, and a nostril. Since *OpenPose* will extract the contour of the chin, it can help extract the face contour. However, *OpenPose* cannot extract the contour of the upper part of the face including the forehead due to the hair.

For solving this limitation, we propose a *simple face contour approximation model* that consists of two half circles and quadratic curve. The upper half circle will draw the forehead and the lower half circle will draw the chin. The two line segments that connect the ends of the two half circles will draw the edges of the face contour. Then, the parameters of this model including the center coordinates and the radii of the half circles should be properly selected so that the resulting model is well matched with the keypoints found by *OpenPose*. *paraOpt* is applied to the optimization of the parameters. Figure 4.1 illustrates the face contour approximation model and the related keypoints by *OpenPose*, which is obtained from [22].

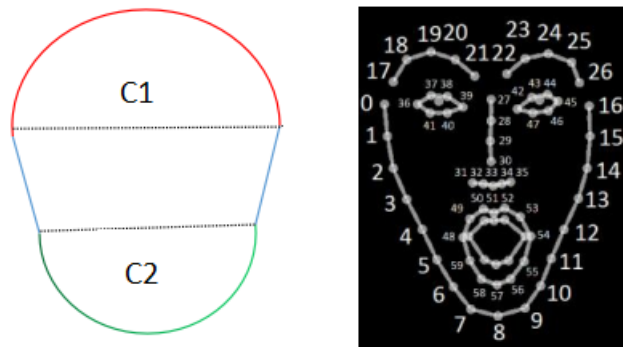


Figure 4.1: Face contour model and *Openpose* keypoints.

4.4 Model Generation Procedure

Figure 4.2 shows the procedure of generating the *face contour approximation model*. It is noted that the image in this figure was generated by using the online deep learning model [23]. First, the user prepares the face image to be drawn. Second, by applying the image to *OpenPose*, the keypoints of the face are extracted from the image and saved into the *Json* file. Finally, our Python program for *paraOpt* reads the keypoints and optimizes the parameters of the model.

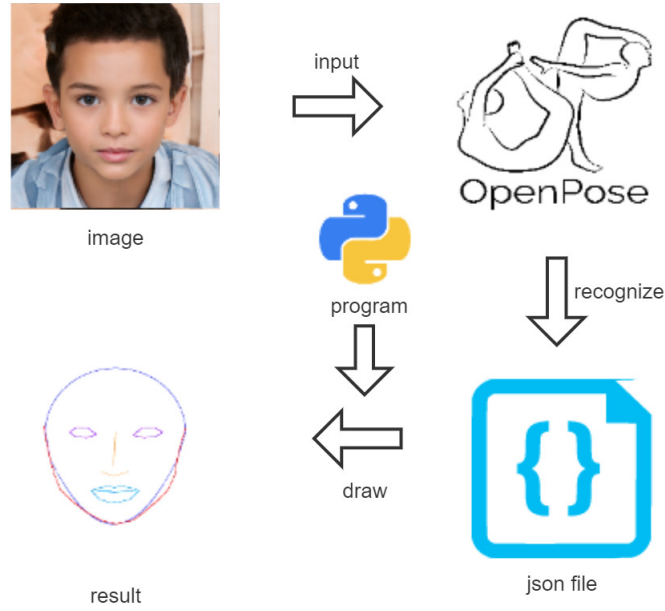


Figure 4.2: Face contour model generation procedure.

4.5 Initial Parameter Values

The initial values of the parameters are obtained from the related keypoint coordinates. For the upper half circle $C1$, keypoint 27 of *Openpose* is used for the center, and the Euclidean distance between two keypoints 27 and 16 is used for the radius. For the lower half circle $C2$, keypoint 33 is used for the center, and the distance between keypoints 33 and 8 is used for the radius.

4.6 Example Initial Model

Figure 4.3 shows the example image and the face contour model using the initial parameter values. This image was also generated by using the same online deep learning model [23]. The red line represents the contour that is given by tracing the keypoints by *OpenPose*, and the green line represents the model. Some differences can be recognized between them. Thus, the parameters of the model should be optimized.

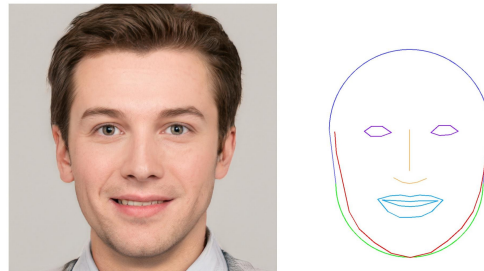
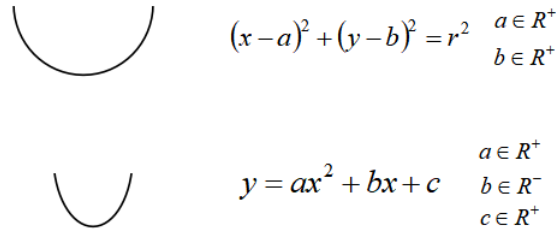


Figure 4.3: Example image and face contour.

4.7 Alternative Model

It has been observed that the chins of some persons are not round but rather sharp. For such faces, an alternative model is proposed. Here, instead of the lower half circle, a quadratic curve shown in Figure 4.4 is used, which was originally drawn in this paper. The initial values of the three coefficients, a , b , and c , are calculated by solving the equations that will be introduced by assuming that this quadratic curve will cover the three keypoints 2, 8, and 14.



$$(x-a)^2 + (y-b)^2 = r^2 \quad \begin{matrix} a \in R^+ \\ b \in R^+ \end{matrix}$$

$$y = ax^2 + bx + c \quad \begin{matrix} a \in R^+ \\ b \in R^- \\ c \in R^+ \end{matrix}$$

Figure 4.4: Alternative model for chin.

4.8 Score Function

To optimize the parameters for the human face contour approximation, the score function $S(P)$ is calculated by the following procedure:

- (1) Calculate the Euclidean distance between each of the 17 keypoints (0~16) and keypoint 33 in the *OpenPose* result in Figure 4.1 respectively.
- (2) Find the corresponding coordinate on the function of the proposed model to each of the 17 keypoints by calculating the y coordinate on the function that has the same x coordinate.
- (3) Calculate the Euclidean distance between each corresponding point to the 17 keypoints and the keypoint 33 respectively.
- (4) Calculate the score function $S(P)$ by:

$$S(P) = \sum_{i=0}^{16} |E_i^k - E_i^s| \quad (4.1)$$

where E_i^k represents the Euclidean distance between keypoint i for $i = 0 \sim 16$, and keypoint 33, and E_i^s denotes the Euclidean distance between the corresponding coordinate on the model function and keypoint 33.

In the parameter optimization algorithm, tabu $t_i = 10$, and $\Delta p_i = 1$ are used.

4.9 Evaluations

Here, we evaluate the proposal for the *human face contour approximation model*.

4.9.1 Face Images

For evaluations, 200 face images with 1024×1024 pixels are collected from an online site. They are artificially generated using the *deep learning model*, including both genders, and various ages and races. Figure 4.5 shows some of them that were generated by the online deep learning model [23].



Figure 4.5: images example.

4.9.2 Optimization Results

Table 4.1 shows the number of images that selects each model, and the average score results before and after applying *paraOpt* for all the images. The results suggest that most chin shapes can be approximated by a quadratic curve, where the score is smaller than that for the half circle.

Ideally, the score should be zero where all the 17 keypoints are on the model function. However, it is not realistic, because the adopted model functions may not well represent the face contour, and *OpenPose* usually make some errors on keypoints. It is necessary to find and define proper model functions that will reduce the scores depending on human faces. It will be in future works.

Table 4.1: Parameter optimization results.

model	#	before optimization	after optimization
half circle	21 (10.5%)	473.51	448.80
quadratic curve	179 (89.5%)	274.31	263.05

Figures 4.6 depicts the results of the face contour approximation models and the keypoints in faces by *OpenPose* for the nine face images in Figure 4.5. Figure 4.7 compares the score results between the two models. The half circle model is better for only three images of 2, 3, and 4. The score difference between the scores is larger for the images where the quadratic curve model is better.

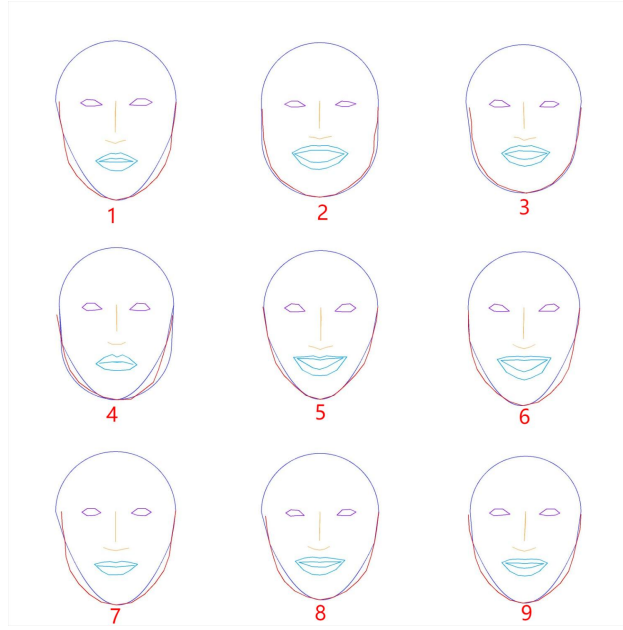


Figure 4.6: Face contour results after optimization.

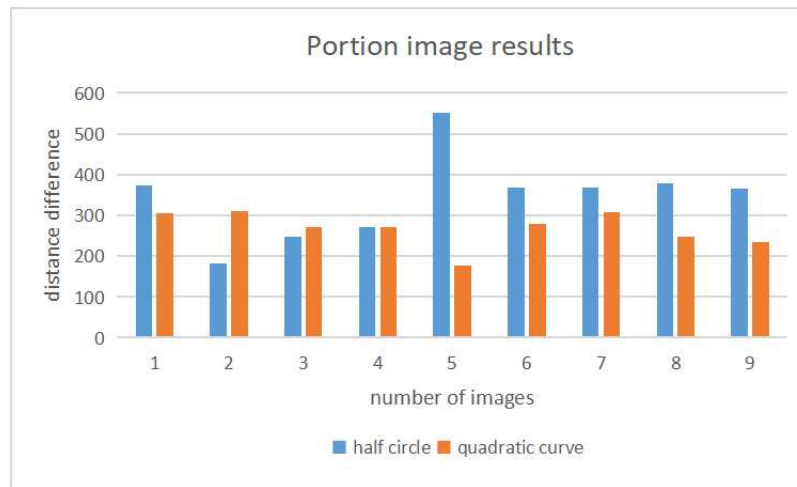


Figure 4.7: Score comparison between two models after optimization.

4.10 Summary

This chapter presents the implementation of *paraOpt* for *face contour approximation model*. Half circles and quadratic curves are used to approximate the human face contour. Results show the optimal approximation line is the quadratic curve. In the next chapter, I will present the *CFD Simulation model*.

Chapter 5

Application to CFD Simulation

5.1 Background

Nowadays, *air conditioners (ACs)* are equipped in many rooms in houses, schools, factories, and offices to offer comfortable environments for humans and machines. On the other hand, global warming has been escalated due to over consumptions of fossil fuels. Therefore, the proper use of ACs has become more important around the world.

Then, the estimation or prediction of the distributions of the temperature or humidity in a room using a simulation model will be useful to properly control ACs. By estimating the room environment changes under various actions, it will be possible to decide when ACs be turned on or off. Even, the timing to open or close windows in the room can be selected.

ACs rely on a limited number of sensors for measuring the temperature and humidity in the room. Therefore, to obtain the distribution of the temperature or humidity in a whole room, additional sensors should be used together by externally allocated in the room, which is not practical. Moreover, the sensors cannot predict future changes of them.

To estimate or predict the distributions in a room together with sensors, we are investigating the *Computational Fluid Dynamics (CFD)* simulation using the *Open Field Operation and Manipulation (OpenFOAM)* software [24]. Then, the optimization of the parameters in *OpenFOAM* is critical in order to fit well the simulation results with the corresponding measured ones.

5.2 Technologies

5.2.1 CFD

CFD is a branch of fluid mechanics that uses numerical methods and algorithms to solve and analyze fluid flow problems. *CFD* involves the simulation of the fluid flow, the heat transfer, and the related phenomena using computational techniques.

CFD can be applied to a wide range of industrial applications. It has been commonly used in engineering design processes to analyze and optimize the performances of fluid systems and components. For example, it is used in the aerospace industry to study the aerodynamics of aircrafts, in the automotive industry to analyze airflows around vehicles, and in the energy sector to optimize the design of turbines and heat exchangers.

5.2.2 OpenFOAM

OpenFOAM is a popular open-source computational fluid dynamics for *CFD* software package. It provides a comprehensive suite of tools and libraries for solving a wide range of fluid flow and heat transfer problems using numerical methods. *OpenFOAM* was initially developed in the late 1980s at Imperial College London, and has grown into a robust and versatile *CFD* platform. It is written primarily in C++ and follows an object-oriented approach, allowing users to create custom solvers, models, and utilities.

5.2.3 Heat Flux

Heat flux refers to the rate of heat transfer per unit area. It is a measure of the amount of heat energy that flows through a given surface per unit time. Heat flux is typically represented by the symbol " q " and is measured in units of watts per square meter (W/m^2) or other appropriate units of power per unit area. Mathematically, heat flux is defined as:

$$q = \lambda \frac{\Delta T}{\Delta x} \quad (5.1)$$

where

- q represents the heat flux, where the unit is W/m^2 .
- λ represents the thermal conductivity through a specified material, which is expressed as the amount of heat that flows per unit of time through a unit area with a temperature gradient of one degree per unit distance.
- ΔT represents the difference between the outside and inside temperatures of the wall, where the unit is *Kelvin*(K).
- Δx represents the thickness of the wall, where the unit is *meter*(m).

5.3 Model Room for Experiments

In a real room, it is very difficult or impossible to control the temperature or humidity to be the required one for the experiment under various weathers or seasons. To solve this problem, a small-sized model room for experiments in Figure 5.1 was assembled for this study. The size of this model room is $1m \times 1m \times 1m$, and is covered by the outer box whose size is $2m \times 2m \times 1.5m$. The walls of this box are insulated with the 30mm thick insulations. In the model room, the temperature-controlled air using an air conditioning unit can be supplied. Besides, at the bottom of the model room, the heaters are mounted to raise the temperature. To measure the temperature distribution of the room, 27 temperature sensors are installed at equal intervals in the room.

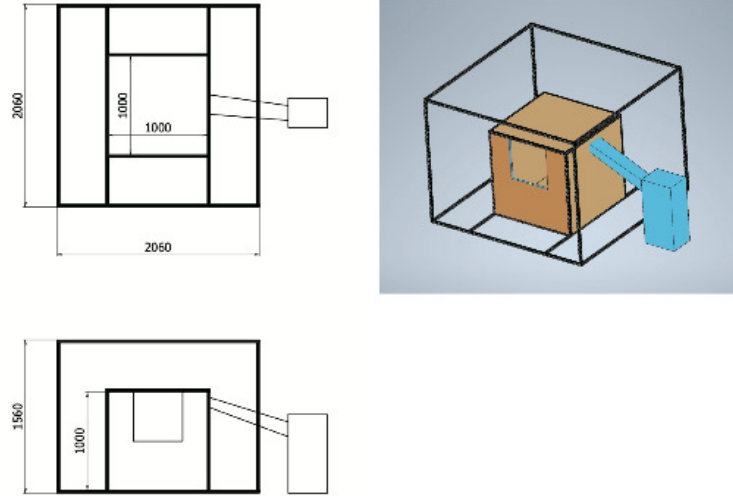


Figure 5.1: Model room for experiments.

5.4 *CFD* Simulation Model and Parameters

To estimate the temperature distribution of the model room, the *CAD* model for *OpenFOAM* in Figure 5.2 is made to represent the room structure. The dimension of the *CAD* model is the same as the real one.

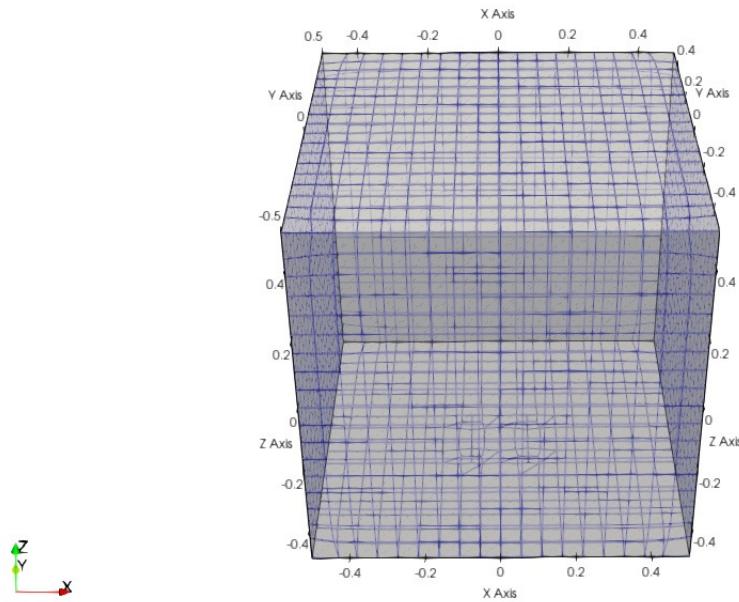


Figure 5.2: Model-room for *CFD* simulation.

Before starting the *CFD* simulation using *OpenFOAM*, the boundary conditions for the walls

and the heater need to be set properly, since they strongly influence the simulation results. Table 5.1 shows some examples of them. The *zeroGradient* represents the adiabatic condition and *fixedValue* represents the wall having the fixed temperature. The boundary condition of the heater is given by *heat flux* that will be presented later. The origin coordinate (0, 0, 0) in the *CAD* model is selected as the monitoring point where the sensor is mounted. Figure 5.3 shows some simulation results.

Table 5.1: Boundary conditions.

wall	zeroGradient	zeroGradient	zeroGradient	fixedValue
heater	500	550	600	600

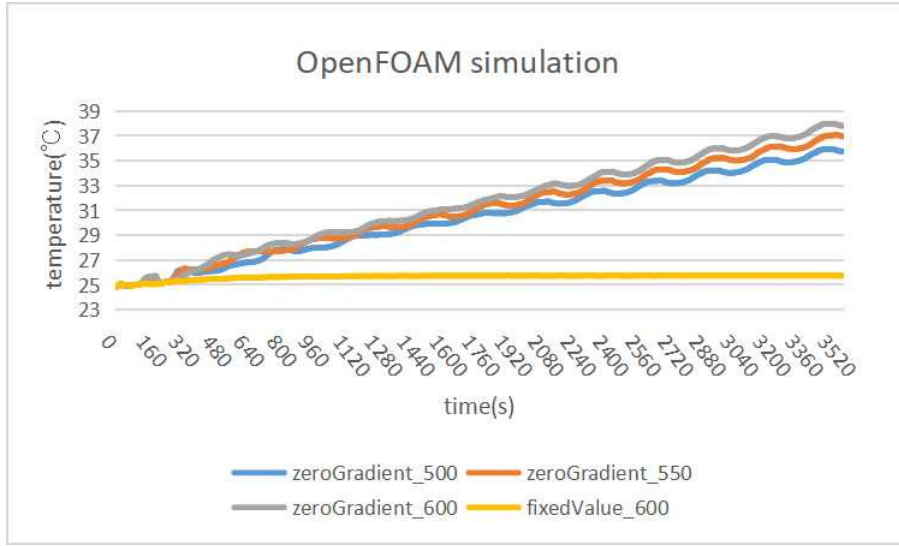


Figure 5.3: *OpenFOAM* simulation results.

5.5 Score Function

The boundary conditions of the walls have large influences on the temperature changes in the room. To accurately predict the temperature changes, the values of the boundary condition parameters in *OpenFOAM* should be optimized. The score function $S(P)$ is calculated from the given simulation heat flux values P and the measured temperatures by the following procedure:

- (1) Record the simulation temperature every five seconds for one hour.
- (2) Calculate the absolute value of difference simulation temperature between measurement actual temperature.
- (3) Calculate $S(P)$ by:

$$S(P) = \sum_{i=0}^N |T_s^i - T_m^i| \quad (5.2)$$

where T_s^i does the i -th simulated temperature at every five seconds, T_m^i does the i -th measured temperature saved at every five seconds, and N does the total number of temperatures.

5.6 One Parameter Evaluations

5.6.1 Experiment Setup

In experiments, the initial boundary conditions in Table 5.2 are used. *zeroGradient* represents the adiabatic condition of the wall. *fixedValue* represents that the outside of the wall has a fixed temperature. The initial temperature of the room including the inside and outside of the wall is 24.85 °C. As the critical boundary condition parameter, the value of *heat flux* is optimized by *paraOpt*, where the values in Table 5.2 are used as the initial values.

Table 5.2: Parameters values before proposal.

number	mesh	heater	boundary condition of wall
pattern1	10,000	heat flux 500	zeroGradient
pattern2	10,000	heat flux 500	fixedValue

5.6.2 Optimization Results

Figures 5.4 and 5.5 show the *CFD* simulation results after optimizing the parameters using *paraOpt* with *pattern1* and *pattern2*, respectively. When the two results are compared, *pattern2* is better.

In *pattern1*, *paraOpt* finds 100 W/m² for the optimal *heat flux* value. Figure 5.4 compares the measurement and simulation temperatures. Although the *heat flux* is relatively small, the room temperature increases rapidly, and continues to increase. Here, due to the adiabatic condition, no heat is dissipated to the outside of the room. However, the measurement temperature is saturated and the heat is dissipated outside the room, which suggests that the walls are not adiabatic.

In *pattern2*, *paraOpt* finds 1390 W/m² for the optimal *heat flux* value. Figure 5.5 shows that the measurement and simulation temperatures are similar, where the temperature difference is only 0.22 °C. *paraOpt* can find the proper parameter value with the proper assumption of the simulation model.

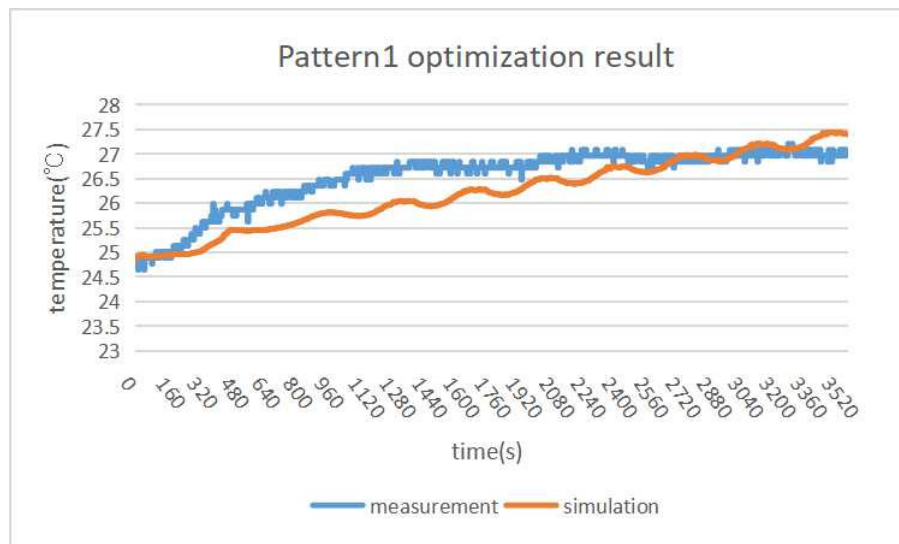


Figure 5.4: Simulation result after optimization with *pattern1*.

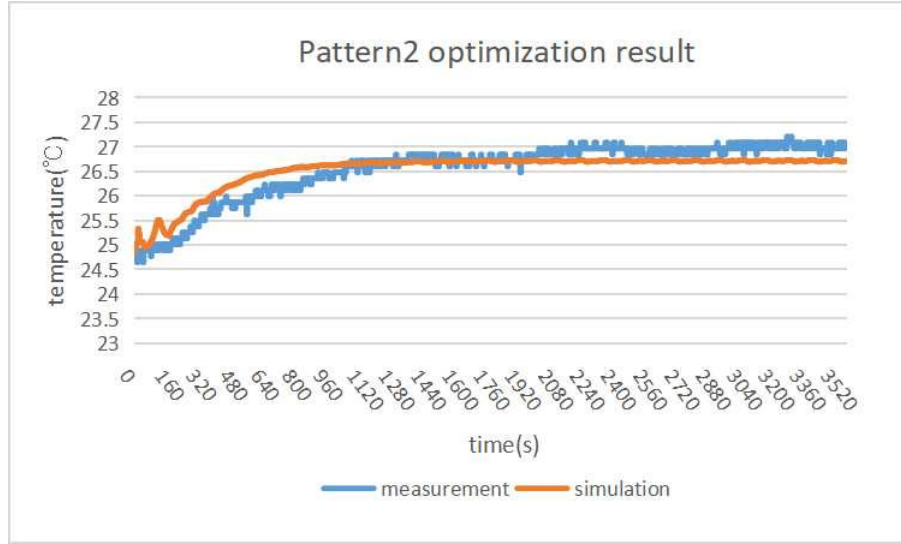


Figure 5.5: Simulation result after optimization with *pattern2*.

5.7 Four Parameters Evaluations

To accommodate higher temperature variations as well as to improve accuracy, the four parameters are optimized in this chapter.

5.7.1 Experiment Setup

Table 5.3 shows the parameters for the boundary conditions related to *heat flux* whose values are optimized in this study.

Table 5.3: Parameters for boundary conditions.

parameter	description
Q	power of heater
T_h	temperature of heater
h	heat transfer coefficient
T_w	temperature of wall

Then, at each wall, the following *heat flux* q is transferred to the wall from the chamber:

$$q = h(T_f - T_w) \quad (5.3)$$

where

- q represents the *heat flux* (W/m^2).
- h represents the *heat transfer coefficient* $W/(K \cdot m^2)$ [25].
- T_f represents the temperature of the fluid along the wall at a certain moment (*Kelvin*(K)).
- T_w represents the temperature of the wall (*Kelvin*(K)).

Table 5.4: Initial parameter values for *paraOpt*

parameter	unit	initial value	range
Q	W	1	1~30
T_h	K	345	300~350
h	$W/(K \bullet m^2)$	1	1~30
T_w	K	345	300~350

Table 5.4 shows the parameter values for *paraOpt*. The initial temperature of the room at both the inside and the outside of the wall is $26.76^\circ C$.

To evaluate the difference between the estimated temperature and the measured one, the *coefficient of determination* R^2 is calculated by the following equation. R^2 becomes closer to 1 as the difference becomes smaller.

$$R^2 = 1 - \frac{\sum_{i=0}^m (y_i - \hat{y}_i)^2}{\sum_{i=0}^m (y_i - \bar{y}_i)^2} \quad (5.4)$$

where:

- y represents the measured temperature ($^\circ C$).
- \hat{y}_i represents the estimated temperature ($^\circ C$).
- \bar{y}_i represents the average measured temperature ($^\circ C$).

5.7.2 Optimization Results

The model estimation accuracy is evaluated when all of the four parameters in Table 5.4 are optimized and their values are fixed.

Figures 5.6 and 5.7 show the estimated and measured temperature results using the training data and the validation data, respectively. The average temperature difference between estimated and measured during one hour is $0.79^\circ C$ and $0.68^\circ C$. The difference becomes much smaller.

However, when the temperature becomes high as the time elapses, the difference increases. To improve it, the parameter values should be changed at the high temperature of $341K$. Table 5.5 shows their optimized values, and the average difference and the coefficient of determination between the estimated and measured temperature results. The result suggests the four parameters optimization can improve the accuracy, but is still not sufficient.

Last, the model estimation accuracy is evaluated when all of the four parameters are optimized and their values can be changed by the temperature. Here, the different values can be given to each parameter by the temperature range of $300 - 317K$, $317 - 341K$, and $341 - 346K$. These ranges are also optimized by the parameter optimization tool.

Figures 5.8 and 5.9 show the estimated and measured temperature results using the training data and the validation data, respectively. The average temperature difference is $0.06^\circ C$ and $0.31^\circ C$. The difference becomes small at any temperature and is not increasing as the temperature increases. Table 5.6 shows the optimized values, and the average difference and the coefficient of determination between the estimated and measured temperature results. The result suggests the four parameters optimization with changed values can improve the accuracy sufficiently.

Table 5.5: Results by four parameters optimization with constant value.

parameter	after proposal	temperature difference	R^2
Q	10	$0.79^\circ C$	0.9878
T_h	345		
h	3		
T_w	321		

Table 5.6: Results by four parameters optimization with changed value.

parameter	after proposal			temperature difference	R^2
Q	300-317K	317-341K	341-346K	$0.06^\circ C$	0.9994
	10	10	10		
T_h	300-317K	317-341K	341-346K		
	317	341	346		
h	300-317K	317-341K	341-346K		
	5.5	3	0.5		
T_w	300-317K	317-341K	341-346K		
	300	317	341		

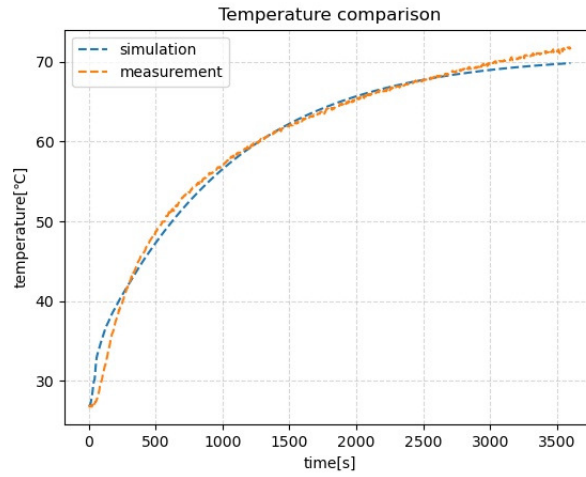


Figure 5.6: Temperature results by four parameters optimization with constant value for training.

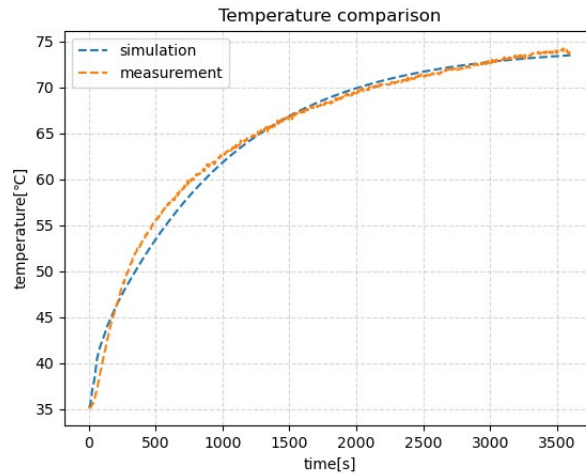


Figure 5.7: Temperature results by four parameters optimization with constant value for validation.

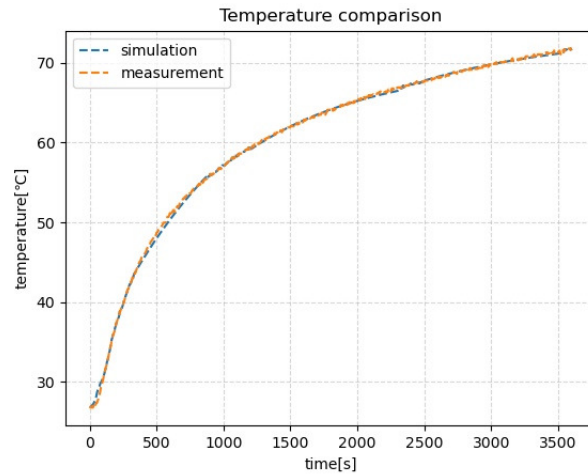


Figure 5.8: Temperature results by four parameters optimization with changed value for training.

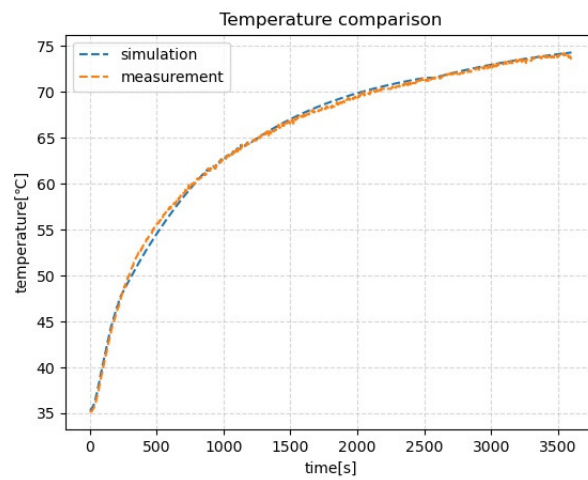


Figure 5.9: Temperature results by four parameters optimization with changed value for validation.

5.8 Summary

This chapter presented the improvement of the *CFD* model by optimizing the one and four parameters together with the *parameter optimization algorithm* and allowing different values to them by the temperature. The evaluation results show that the average difference between the estimated and measured temperature is reduced to 0.22°C with one parameter optimization, and 0.06°C with four parameters optimization. In the next chapter, I will discuss the results for three applications.

Chapter 6

Discussions

In this chapter, I discuss the performances of *paraOpt* in the three applications in this thesis.

6.1 Performances in Three Applications

First, I examine the effectiveness of the proposed algorithm by comparing the accuracies in the three applications before and after applying it. Basically, the accuracy after applying the proposal is sufficiently high, where the one for *Face Model* may be improved. The **before** and **after** represent the before optimization results and optimized results. For *FILS15.4*, the average detection accuracy is increased from 81.2% to 99.01%, where the improvement is 17.81%. For *Face Contour Approximation Model*, the average Euclid distance is decreased from 373.90 to 355.92, where the improvement is 17.98. For *CFD model* of one parameter optimization, the average temperature difference is decreased from 0.9 °C to 0.22 °C, where the improvement is 0.68 °C. For *CFD model* of four parameters optimization, the average temperature difference is decreased from 0.79 °C to 0.06 °C, where the improvement is 0.73 °C.

6.2 Complexities of Three Applications

Among the three applications, the parameter optimization of *FILS15.4* is the most complicated, because it has a lot of critical parameters to determine the accuracy, and even the number of parameters needs to be optimized. For this application, the detection interval, the number of fingerprints for each room or detection unit, and the fingerprint values should be optimized. They are related to each other. Since the fingerprint values can be optimized after the detection interval and the number of fingerprints is selected, we optimize them sequentially in this order in the thesis.

The remaining two applications, *Face Model* and *CFD*, have less complexity than *FILS15.4*, where the number of parameters is fixed and is relatively small. However, they keep non-linearity in optimizing the parameter values in terms of accuracy. I believe that they are still complicated problems where the initial value selection is critical to improving the accuracy.

6.3 Parametrizations in Three Applications

For *FILS15.4*, all the possible parameters are parameterized to optimize their values except for the number and locations of receivers that should be allocated in the field. Currently, these parameters

can be optimized by manually inserting, moving, or removing receivers. They can be optimized if the accurate model of the signal propagation is available for the field, which will be in future works.

For *Face Model*, currently, only two simple functions, *half circle* and *quadratic curve*, are considered. Then, there is a gap in approximating the jaw part of the face contour. The *half circle* can be too fat, whereas the *quadratic curve* can be too thin. Therefore, other functions will be necessary to continuously approximate it. Thus, the optimization of the approximate function should be more generalized and parameterized to further improve the accuracy, which will be in future works.

For *CFD*, to improve the calculation accuracy of *CFD*, basically, finer meshes and more physical parameters should be considered. However, they will further increase the calculation time. Since *heat flux* often differs from the value described in the specifications of the wall materials due to construction conditions, several relative parameters are optimized in this thesis. Other parameters, such as the wall thickness, the number of meshes, and the time step, can be optimized to further improve the accuracy, which will be in future works with the speedup of *CFD*.

6.4 Summary

This chapter discussed the performances of *paraOpt* in the three applications in this thesis. In the next chapter, the related work in literature will be presented.

Chapter 7

Related Works in Literature

In this chapter briefly review related works to this thesis.

In [14], Uradzinski et al. proposed the nearest neighbor and Bayesian methods using *IEEE 802.15.4* protocol devices, which promised less than or equal to the $0.81m$ accuracy. They first collected data and created a fingerprint database. Next, they used the nearest-neighbor and Bayesian methods to detect the indoor positioning of each person. However, they did not evaluate the proposal in multiple rooms and considered human effects in experiments. In our *FILS15.4* application, we considered human effects and we optimize the fingerprint value and number of fingerprints by using *paraOpt* to improve application robustness.

In [26], Xi et al. proposed a smart *hill-climbing* algorithm based on *RHC* to configure the parameters in the server that can influence the server response automatically. They formulated the problem of finding the optimal configuration for a given application as the black-box optimization problem. They carried out extensive experiments with an online brokerage application running in a *WebSphere* environment. The results demonstrated that the algorithm is superior to traditional heuristic methods. To compare with *paraOpt*, hill-climbing does not involve any backtracking mechanism. If a move leads to a worse solution, the algorithm abandons that path entirely and doesn't attempt to backtrack to a previous better solution.

In [27], Bernas et al. introduced a method that improves localization accuracy of the signal strength fingerprinting approach. In the proposed method, the entire localization area was divided into several regions by clustering the fingerprint database. For each region, a sample of the received signal strength was determined and a dedicated artificial neural network (*ANN*) was trained by using only the fingerprints that belonged to this region (cluster). Compared with our fingerprint method, their detection accuracy is higher. However, *ANN* model needs plenty of data to train and adjust hyperparameters.

In [28], Zhao et al. proposed a hybrid annealing particle swarm optimization localization algorithm based on the *simulated annealing*. They proposed the minimum positioning error weighting model to reduce the non-line-of-sight error of anchor nodes during positioning. In experiments, they deployed 25 nodes on the square area of $100m \times 100m$ where the communication radius of a node is $20m$. The results showed that the localization average error of distance when locating these nodes by using the algorithm is $0.3775m$.

In [29], Ghadimi et al. proposed an algorithm to optimize the shape of the centrifugal blood pump based on the *genetic algorithm*. They applied the proposal to optimize the parameters of the *CFD* simulation to improve the performance. The results showed that the hydraulic efficiency was improved 11.1% and the hemolysis index was reduced to 11.8% by using the optimized shape of the centrifugal blood pump. To compare with our *CFD* optimization method, they used global

optimization algorithm *GA*, which can find global optimal results. For parameters, they focus on which parameters can influence the model quality, not parameters in physical equations.

In [30], Yanan et al. propose an automatic *CNN* architecture design method by using genetic algorithms, to effectively address the image classification tasks. The proposed algorithm is validated on widely used benchmark image classification datasets, by comparing it to state-of-the-art peer competitors covering eight manually-designed *CNNs*, seven automatic + manual tunings, and five automatic *CNN* architecture design algorithms. The experimental results indicate the proposed algorithm outperforms the existing automatic *CNN* architecture design algorithms in terms of classification accuracy, the number of parameters, and consumed computational resources. The proposed algorithm also shows a very comparable classification accuracy to the best one from manually-designed and automatic + manual tuning *CNNs*, while consuming much less computational resources.

In [31], A.A.N. et al. propose SA to solve the CVRP problem. The problem is modeled as the capacitated vehicle routing problem (*CVRP*). The *CVRP* is known as an NP-Hard problem. The SA algorithm is compared to a commonly used heuristic known as the nearest neighborhood heuristics for the case study dataset. The results show that the simulated annealing and the nearest neighbor algorithms are performing well based on the percentage differences between each algorithm with the optimal solution being 0.03% and 5.50%, respectively. Thus, the simulated annealing algorithm provides a better result compared to the nearest neighbor algorithm. Furthermore, the proposed simulated annealing algorithm can find the solution as same as the exact method quite consistently.

In [32], Setiabudi et al. proposed a method using Bluetooth low energy (*BLE*) to estimate the position of a dynamic user based on fingerprinting with the weighted sum of five nearest reference points using the extended Kalman filter. Unfortunately, to compare with IEEE 802.15.4 protocol, even though they conducted measurements in a real environment, the proposed method needs to allocate a lot of transmitters in the target field, and the positioning accuracy is not sufficient.

In [33], Ashraf et al. presented a comprehensive review of the approaches that made use of data from one or more sensors to estimate the user's indoor location. By analyzing the approaches leveraged on smartphone sensors, the review discusses the associated challenges of such approaches and points out the areas that need considerable research to overcome their limitations.

In [34], Njima et al. proposed generative adversarial networks for the *RSSI* data augmentation to generate fake *RSSI* data based on a small set of real collected labeled data. The developed model utilizes the semi-supervised learning in order to predict the pseudo-labels of the generated *RSSI*. Their extensive numerical experiments show that the proposed data augmentation and selection scheme leads to the localization accuracy improvement of 21.69% for simulated data and 15.36% in the experiment data.

In [35], Fahmy et al. proposed a *Wi-Fi*-based indoor localization system named *MonoFi*. It relied on the received signal strength from a single access point and trained the recurrent neural network with sequences of signal measurements. They conducted measurements in real environments. The results show that the median localization error was 0.80 *m* in their experiments.

In [36], Jiang et al. proposed a fingerprint-based indoor localization method named the *fingerprint feature extraction (FPFE)*. It uses *Wi-Fi* signals to detect human locations. The average detection error in experiments using one room in real environments was 0.68 *m*. To compare with our experiment layout, they didn't conduct experiments in multiple rooms.

In [37], Ezhumalai et al. proposed an *RSS* measurement technique named (*IRSSMT*) to minimize the error of *RSS* observations by using several selected *RSS* and its median values, and the *strongest access point (SAP)* information-based clustering technique that groups the *reference points (RPs)* using the *SAP* similarity.

In [38], Noor et al. presented the non-Newtonian fluid simulations via *OpenFOAM*. They focus on the implementation and functionality of the code of the non-Newtonian power law equations and used the *finite volume method (FVM)*. The simulation results were shown with graphs and animated videos. The flow analysis states the behavior of the velocity field when the fluid hits the obstacle. The animated videos further include the behavior of velocity in the leaving zone of the cylinder obstacle. A clear view of fluid flow can be seen far from the cylindrical object.

In [39], Zhang et al. they presented the fins were introduced to form a secondary channel to optimize the heat dissipation effect of the microchannel heat sink. They combined with *CFD* method, the structure parameters (the angle, lateral spacing and vertical spacing) and arrangement ways (the aligned, staggered) of internal fins in microchannel were analyzed. The results are shown compared with the straight microchannel heat sink, the maximum temperature and average temperature of optimized model are reduced by 3.04 K (6.67%), 2.86 K (6.75%), respectively. To compare with our *CFD* model, they also reduced temperature difference by optimizing parameters that are related to heat transfer. However, they focus on the parameters of the structure parameters of shape, not the physical parameters of heat transfer.

Chapter 8

Conclusion

This thesis presented three applications of the parameter optimization algorithm (*paraOpt*) and showed the superiority of the proposal.

Firstly, I reviewed the optimization algorithm called *paraOpt* that has been proposed as the general-purpose parameter optimization algorithm in our previous researches. *paraOpt* was designed on metaheuristic algorithms used for solving combinatorial optimization problems. This algorithm starts with an initial solution that is generated randomly or is sequentially constructed based on heuristics. A neighborhood of the current solution is defined by generating a set of candidate solutions that can be obtained by making small modifications to the current solution. Each candidate solution is evaluated using the score function, and the best neighborhood is selected as the new current solution.

Secondly, I study the applications of *paraOpt* in three diverse applications.

The first application of *paraOpt* is the *fingerprint-based indoor localization system* using IEEE 802.15.4 devices called *FILS15.4* that can detect the location of a user in an indoor environment. The use of *FILS15.4* consists of two phases. In the calibration phase, the pattern of the typical received signal strength or *link quality indicator (LQI)* for every room is collected as the fingerprint by locating the transmitter there. In the detection phase, the location of the transmitter is continuously found by comparing the current *LQI* pattern with every fingerprint and by selecting the closest one. In *FILS15.4*, the number of fingerprints for each detection point, the fingerprint values, and the detection interval are optimized together by applying *paraOpt*, which achieves the average detection accuracy with higher than 98%.

The second application of *paraOpt* is the *human face contour approximation model* that is described by a combination of half circles and quadratic curves. To approximate this model from the face image, *OpenPose* is used to extract the human face contour. In the human face contour approximation model, the coordinates, radius, and coefficients of several simple functions to compose the model are optimized, which can well approximate the face contours of various persons. The average face contour Euclidean distance difference of *OpenPose* from the face contour approximation model is reduced to 17.98.

The third application of *paraOpt* is the *computational fluid dynamic (CFD) simulation* to estimate temperature changes in a room. In the *CFD* simulation, one or four parameters of the model are optimized to minimize the average temperature difference between the estimated and measured ones. The experiment results show that when the value of one parameter is optimized, the average temperature difference between the estimated and measured ones becomes 0.22°C, and when the values of four parameters are optimized, the average temperature difference between the estimated and measured ones becomes 0.06°C.

In future works, I will improve the parameter optimization algorithm and evaluate it in other applications.

Bibliography

- [1] “Convolutional neural network,” Wikipedia, https://en.wikipedia.org/wiki/Convolutional_neural_network (accessed May 9, 2023).
- [2] “Recurrent neural network,” Wikipedia, https://en.wikipedia.org/wiki/Recurrent_neural_network (accessed May 9, 2023).
- [3] Y. Huo et al., “A proposal of the fingerprint optimization method for the fingerprint-based indoor localization system with IEEE 802.15.4 devices,” *Information*, vol. 13, no. 5, p. 211, 2022. doi:10.3390/info13050211
- [4] “OpenPose: Human pose estimation method,” GeeksforGeeks, <https://www.geeksforgeeks.org/openpose-human-pose-estimation-method/> (accessed May 14, 2023).
- [5] “Computational Fluid Dynamics,” Wikipedia, https://en.wikipedia.org/wiki/Computational_fluid_dynamics (accessed May 14, 2023).
- [6] N. Funabiki, C. Taniguchi, K. S. Lwin, K. K. Zaw, and W.-C. Kao, “A parameter optimization tool and its application to throughput estimation model for Wireless LAN,” *Advances in Intelligent Systems and Computing*, pp. 701–710, 2017. doi:10.1007/978-3-319-61566-0_65
- [7] K. Curran et al., “An evaluation of Indoor Location Determination Technologies,” *Journal of Location Based Services*, vol. 5, no. 2, pp. 61–78, 2011. doi:10.1080/17489725.2011.562927.
- [8] J. Kunhoth, A. Karkar, S. Al-Maadeed, and A. Al-Ali, “Indoor positioning and wayfinding systems: A survey,” *Human-centric Computing and Information Sciences*, vol. 10, no. 1, 2020. doi:10.1186/s13673-020-00222-0
- [9] P. Davidson and R. Piche, “A survey of selected indoor positioning methods for smartphones,” *IEEE Commun. Survey. Tutor.*, vol. 19, no. 2, pp: 1347-1370, Dec. 2016.
- [10] B. Molina, E. Olivares, C. E. Palau, and M. Esteve, “A multimodal fingerprint-based indoor positioning system for airports,” *IEEE Access*, vol. 6, pp. 10092–10106, 2018. doi:10.1109/access.2018.2798918
- [11] P. Puspitaningayu et al., “A fingerprint-based indoor localization system using IEEE 802.15.4 for staying room detection,” *International Journal of Mobile Computing and Multimedia Communications*, vol. 13, no. 1, pp. 1–21, 2022. doi:10.4018/ijmcmc.301275
- [12] Mono Wireless, Mono Wireless Product Information. <https://mono-wireless.com/jp/products/index.html> (accessed May 16, 2023).

- [13] L. Luoh, "ZigBee-based intelligent indoor positioning system soft computing", *Soft Comput.*, vol. 18, pp. 443-456, June 2013.
- [14] M. Uradzinski, H. Guo, X. Liu, and M. Yu, "Advanced indoor positioning using Zigbee wireless technology," *Wirel. Per. Commun.*, vol. 97, pp. 6,509-6,518, Aug. 2017.
- [15] B. Wang, Y. Zhao, T. Zhang, and X. Hei, "An improved integrated fingerprint location algorithm based on WKNN," in *Proc. Chinese Cont. Deci. Conf.*, pp. 4580-4584, May 2017.
- [16] M. Alfakih, M. Keche, "An enhanced indoor positioning method based on Wi-Fi RSS fingerprinting," *J. Commun. Soft. Syst.*, vol. 15, no. 1, Mar. 2019.
- [17] "The standard for IOT messaging," MQTT, <https://mqtt.org/> (accessed May 16, 2023).
- [18] R. Yi, Y.-J. Liu, Y.-K. Lai, and P. L. Rosin, "Apdrawinggan: Generating artistic portrait drawings from face photos with hierarchical gans," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019. doi:10.1109/cvpr.2019.01100
- [19] "Facial proportions - how to draw a face," How to Draw a Face - Facial Proportions, <https://thevirtualinstructor.com/facialproportions.html> (accessed May 17, 2023).
- [20] D. Dixon, M. Prasad, and T. Hammond, "ICanDraw," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2010. doi:10.1145/1753326.1753459
- [21] H. Chen et al., "Pictoon," *Proceedings of the tenth ACM international conference on Multimedia*, 2002. doi:10.1145/641007.641040
- [22] "Contents," OpenPose: OpenPose Doc - Output, https://cmu-perceptual-computing-lab.github.io/openpose/web/html/doc/md_doc_02_output.html (accessed May 17, 2023).
- [23] "Unique, worry-free model photos," Generated Photos, <https://generated.photos/> (accessed May 17, 2023).
- [24] OpenFOAM, <https://www.openfoam.com/> (accessed May 24, 2023).
- [25] "Heat transfer coefficient," Wikipedia, https://en.wikipedia.org/wiki/Heat_transfer_coefficient (accessed May 20, 2023).
- [26] B. Xi, Z. Liu, M. Raghavachari, C. H. Xia, and L. Zhang, "A smart hill-climbing algorithm for Application Server Configuration," *Proceedings of the 13th international conference on World Wide Web*, 2004. doi:10.1145/988672.988711
- [27] M. Bernas and B. Płaczek, "Fully connected neural networks ensemble with signal strength clustering for indoor localization in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 11, no. 12, p. 403242, 2015. doi:10.1155/2015/403242
- [28] R. Zhao and Y. Shi, "Indoor localization algorithm based on hybrid annealing particle swarm optimization," 2018 Tenth International Conference on Advanced Computational Intelligence (ICACI), 2018. doi:10.1109/icaci.2018.8377479
- [29] B. Ghadimi, A. Nejat, S. A. Nourbakhsh, and N. Naderi, "Shape optimization of a centrifugal blood pump by coupling CFD with Metamodel-assisted genetic algorithm," *Journal of Artificial Organs*, vol. 22, no. 1, pp. 29-36, 2018. doi:10.1007/s10047-018-1072-z

- [30] Y. Sun, B. Xue, M. Zhang, G. G. Yen, and J. Lv, “Automatically designing CNN architectures using the genetic algorithm for Image Classification,” *IEEE Transactions on Cybernetics*, vol. 50, no. 9, pp. 3840–3854, 2020. doi:10.1109/tcyb.2020.2983860
- [31] A. A. Redi et al., “Simulated annealing algorithm for solving the capacitated vehicle routing problem: A case study of pharmaceutical distribution,” *Jurnal Sistem dan Manajemen Industri*, vol. 4, no. 1, pp. 41–49, 2020. doi:10.30656/jsmi.v4i1.2215
- [32] C. A. Setiabudi, “Indoor positioning system using ble for Tracking Dynamic User Positions,” *International Journal of Emerging Trends in Engineering Research*, vol. 8, no. 2, pp. 455–463, 2020. doi:10.30534/ijeter/2020/33822020
- [33] I. Ashraf, S. Hur, and Y. Park, “Smartphone sensor based indoor positioning: Current status, opportunities, and future challenges,” *Electronics*, vol. 9, no. 6, p. 891, 2020. doi:10.3390/electronics9060891
- [34] W. Njima, M. Chafii, A. Chorti, R. M. Shubair, and H. V. Poor, “Indoor localization using data augmentation via selective generative Adversarial Networks,” *IEEE Access*, vol. 9, pp. 98337–98347, 2021. doi:10.1109/access.2021.3095546
- [35] I. Fahmy, S. Ayman, H. Rizk, and M. Youssef, “Monofi,” *Proceedings of the 29th International Conference on Advances in Geographic Information Systems*, 2021. doi:10.1145/3474717.3486808
- [36] J.-R. Jiang, H. Subakti, and H.-S. Liang, “Fingerprint feature extraction for indoor localization,” *Sensors*, vol. 21, no. 16, p. 5434, 2021. doi:10.3390/s21165434
- [37] B. Ezhumalai, M. Song, and K. Park, “An efficient indoor positioning method based on Wi-Fi RSS fingerprint and classification algorithm,” *Sensors*, vol. 21, no. 10, p. 3418, 2021. doi:10.3390/s21103418
- [38] N. Muhammad, “Finite volume method for simulation of flowing fluid via openfoam,” *Europe. Physic. J. Plus*, vol. 136, no. 10, 2021.
- [39] F. Zhang, B. Wu, and B. Du, “Heat transfer optimization based on finned microchannel heat sink,” *International Journal of Thermal Sciences*, vol. 172, p. 107357, 2022. doi:10.1016/j.ijthermalsci.2021.107357