Implementation and Evaluation of BLE Proximity Detection Mechanism for Pass-by Framework

Ryoma Tabata, Arisa Hayashi, Seiki Tokunaga,
Sachio Saiki and Masahide Nakamura
Graduate School of System Informatics,
Kobe University
1-1 Rokkodai, Nada, Kobe 657-8501, Japan
Email:tabata@ai.cs.kobe-u.ac.jp

Shinsuke Matsumoto
Graduate School of Information Science and Technology,
Osaka University
1-1 Yamadaoka, Suita, Osaka 565-0871, Japan

***Abstract*—To fix various dependencies of application development using pass-by detection by a mobile device, we propose Pass-by Framework that handles data with standardization. In this study, we evaluate effects of performance of pass-by detection by differences in methods of implementation the sonar of Pass-by Framework. Therefore, we develop pass-by application using Bluetooth Low Energy as a first effort. We then conduct evaluation experiments for confirmation of change pass-by detection behavior depends on the difference of parameters.**

1. Introduction

In recent years, mobile devices such as smartphones or tablets have been becoming common rapidly all over the world. On such devices, people can use application for various purposes such as improving QoL or providing entertainment. Then, more and more applications use the advantage of "the mobility". Some applications of them provide services when they detect a proximity of other devices using sensors such as Bluetooth, Wi-Fi or GPS. In this paper, we define an event that entities get close in a certain distance within a certain period of time as *pass-by*. We can detect pass-by by proximity detection mechanism. Besides, we define a system which utilizes pass-by as *pass-by system*. For an example of a pass-by system, "TohakuNavi" [1] which is provided at Tokyo National Museum can display an explanation about nearby showpiece automatically. As another example, "Streetpass Communication" [2], which is a function of Nintendo 3DS, can convert facts of pass-by in an actual world into values in a virtual world of games. In this wise, We expect creations of new value by pass-by systems.

However, a pass-by system has a problem that development makes a big burden on developers. This is because a behavior of pass-by detection depends on a format of pass-by data and management of pass-by data. Furthermore, a pass-by system using different communication technology is incompatible. Hence, the reusability and the interoperability of the implementation developed is low.

To solve these problems, we have proposed *Pass-by Framework* [3] which aims to support for application developers. Pass-by Framework separate proximity detection which depends on the device, and a management of pass-by data which use applications. Then, a developer can decrease costs of development by abstracting them. Furthermore, the Framework can make pass-by data between different communication technologies because the Framework can handle pass-by data without dependence on proximity detection technology.

So far the research about Pass-by Framework concerned with the management of pass-by data mainly. However, we have to implement proximity detection for a realization of the pass-by system. The behavior of devices depends on parameters, which is different depending on technology or execution environment, such as "Communication Frequency" or "Communication Range". The modification of parameter settings influences an accuracy of pass-by detection and a current consumption of devices. To implement the pass-by system for satisfying users, the application developer must set appropriate parameters. However, if each developer research appropriate parameters, their burden of development increase. Therefore, when the developer utilizes Pass-by Framework, we provide the data which is a good reference for determination setting parameter at various technology or execution environment. Thereby, we can achieve "Reduction of the development cost", which is the purpose of Pass-by Framework.

Then, in this study, we consider about the correlation between settings of parameters depending on technology and behavior of pass-by detection. In order to do that, in the first place, we actually develop an application for pass-by detection. In this paper, we use Bluetooth Low Energy (*BLE*) as a proximity detection technology for the application. Next, we conduct evaluation experiments for researches: the correlation between parameters and current consumption; effects of distance between devices on the Received Signal Strength Indication (*RSSI*); and effects of moving speed on accuracy and frequency of detection. Finally, we analyze the experimental data and consider it.

978-1-5090-0806-3/16/$31.00 copyright 2016 IEEE
ICIS 2016, June 26-29, 2016, Okayama, Japan

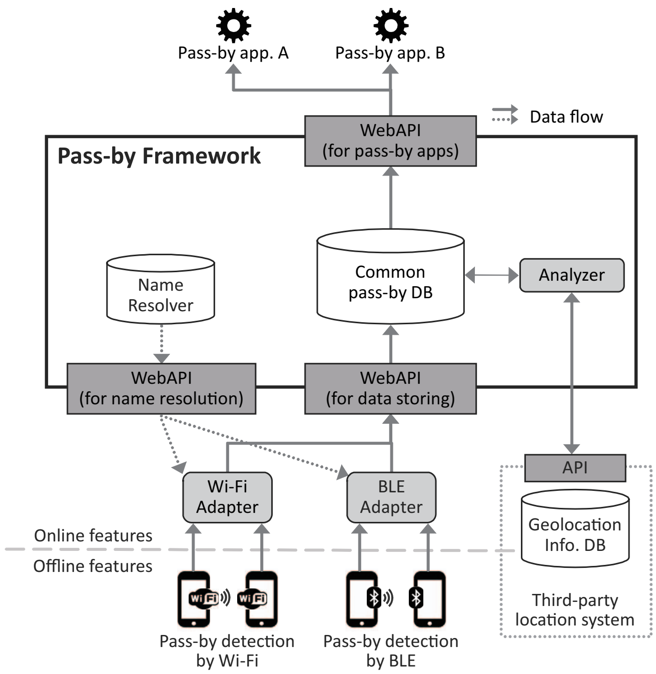

Figure 1: Pass-by Framework

2. Preliminary

2.1. pass-by system

We define an event that entities get close in a certain distance within a certain period of time as *pass-by*. Besides, we define a system which detects pass-by, record it and use this data for arbitrary operations as *pass-by system*. There are two methods of detecting pass-by. The first one is the method which detects pass-by from relative location information between devices. Pass-by detection by BLE or Ad-hoc mode of Wi-Fi corresponds to this. The second one is the method which detects pass-by from absolute location information such as latitude and longitude. The technologies of using absolute location information are that of GPS or infrastructure mode of Wi-Fi.

2.2. Pass-by Framework

On pass-by system, the behavior of pass-by detection, the format of pass-by data, and management of the data depends on proximity detection technology or implementation. Thereby, there is the large burden on application developer for developing. In order to solve the problem, our research group proposes Pass-by Framework [3] which is the mechanism to reduce the burden of development. This Framework separate and abstract the pass-by detection mechanism depending on a device and a communication technology; and management of the pass-by data which is utilized by applications. Thereby, an application developer can develop without minding types of device or technology.

The architecture of Pass-by Framework is shown in Figure 1. The Framework consists of the pass-by detection mechanism (*sonar*) and the server for data management. Pass-by is detected by sonar of the device. Then, the pass-by data is transmitted to *Adapter* which is set for each pass-by detection technology, and the data is converted into the standardized one at Adapter. This standardized data passes *WebAPI*, and then it is recorded in *Common pass-by DB*. Besides, location information recorded by GPS is transmitted to Geolocation Info. DB. *Analyzer* reasons new pass-by by both data of Common pass-by DB and location information of Geolocation Info. DB. Then Analyzer transmits a new pass-by data to Common pass-by DB. When somebody use pass-by data in Common pass-by DB from applications, it can get pass-by data for applications by using WebAPI.

2.3. BLE

BLE [4] is a part of the Bluetooth4.0 standard, and its correct consumption is lower than Bluetooth before 3.0 (*Classic Bluetooth*). BLE utilize 2.4 GHz band that is utilized by Classic Bluetooth. However, BLE does not have backward compatibility with Classic Bluetooth. Classic Bluetooth has a rule that it can connect up to seven devices, but BLE does not have a limited number of connectable devices concurrently. In recent year, the devices supporting Bluetooth4.0 are pervasive. We expect that many devices utilize BLE as pass-by detection mechanism on pass-by systems in the future.

2.4. The scope on this study

As for the past research of Pass-by Framework, we focused on the data management. Therefore, there is no examination about sonar yet. However, we have to develop sonar to utilize the Framework. The behavior of pass-by detection changes with implementation method of sonar. For utilizing a proximity detection technology using radio wave such as BLE on applications, the developer must set parameters, which affect the accuracy of pass-by detection and the current consumption, such as the intensity and frequency of radio wave. If a developer set inappropriate parameters, serious problems such as rapid battery consumption or low accurate consumption. Parameters are different by pass-by detection technology. In this study, we analyze and consider changes (of behavior) of pass-by detection which depends on the difference in setting the parameters. In this paper, as a first step, we use BLE as a pass-by detection mechanism. The reason for selecting BLE at first, many devices adapt it, and we expect that it is utilized as a typical pass-by detection in the future. The parameters of BLE on Android are “Transmission Frequency”, “Transmission Frequency” and “Reception Frequency”. From the point of view of battery consumption, RSSI and relative speed, we consider the setting of appropriate parameters for pass-by detection.

3. Implementation of proximity detection mechanism using BLE

3.1. Requirement of the experimental pass-by application

The application developer who implements a proximity detection mechanism using radio wave need to develop software while considering the following E1-E3, which is elements depending on differences in implementation.

- E1 Influence of parameters to current consumption
- E2 Correlation between distance of targets and RSSI
- E3 Influence of the movement of target to pass-by detection

By researching E1, the developer can predict the current consumption by the application. By researching E2, the developer can understand attenuation of radio waves from the device which is transmitting. By researching E3, the developer can understand the influence of increasing speed on pass-by detection. Considering three elements of the above, we have to implement the following three functions in the application.

- R1 Current consumption measurement function
- R2 Distance measurement function
- R3 Speed measurement function

Furthermore, the following three elements are also necessary.

- R4 Switching execution state of transmitting or receiving
- R5 Modifying settings of parameters
- R6 Recording result of pass-by detection

3.2. The experimental pass-by application

We develop the pass-by application for experimants in the following development environments.

IDE : Android Studio 1.4
Target OS : Android5.0 Lollipop
Target device : Nexus9 (HTC, 2014)

In advertiser application, a byte string of advertising data is created according to the data format of iBeacon [5]. On this format, data have UUID for identification of the application and major and minor for identification of the device. In scanner application, a byte string which is transmitted from advertiser application is analyzed, and we can check UUID, major and minor.

The screenshots of applications are shown in Figure 2. The left hand is the screen of the application for advertising (**BLEAdvertiser**). The right hand is the screen of the application for scanning (**BLEScanner**). The both screens have the button for switching execution state (ON/OFF) in the upper right corner. The both screens have a combo box for change parameter settings in the upper left corner.

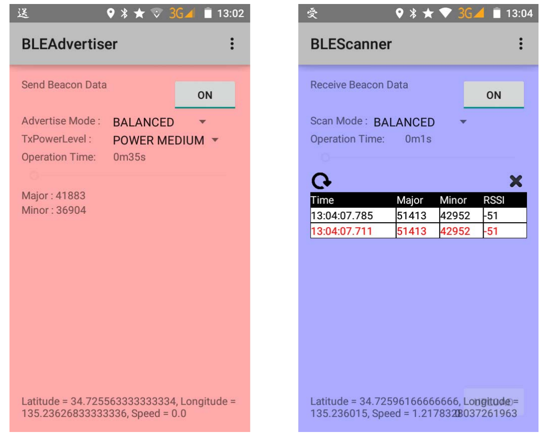

Figure 2: Screenshots of applications

Changeable parameters are "*AdvertiseMode*" (transmission frequency) and "*TxPowerLevel*" (transmission strength) in BLEAdvertiser; and "*ScanMode*" (reception frequency) in BLEScanner.

On the bottom of the both screens, the location information (latitude, longitude) and speed are displayed. The location information, the speed and the time are related and saved in a local file in the device. From location informations of the same time that have been recorded in BLEAdvertiser and the BLEScanner, we calculate a distance between devices.

Besides, when receiving a BLE signal, BLEScanner displays the information of the signal such as receive time, major, minor, and RSSI. Therefore, we can grasp the reception situation of the signal in real time.

For measuring current consumption, we develop another application (*PowerConsumptionMonitor*). PowerConsumptionMonitor has a function that measures average current consumption and records it at regular time intervals [6].

4. Evaluation experiments

4.1. Current consumption

The desired current consumption of pass-by application is different by each purpose of using. For example, in the disaster situation, many people cannot find an opportunity to charge the device easily. In this case, applications such as the safety confirmation should make battery consumption as small as possible. Therefore, the pass-by application developer should know the relationship between the current consumption and battery life. There is the trade-off relationship between a performance of the BLE communication and battery consumption. Thus, the developer should know the difference between the current consumption depending on a difference of parameters.

4.1.1. Conditions.

In Android OS environment, three types AdvertiseMode ("LOW POWER", "BALANCE" and "LOW LATENCY"),

TABLE 1: Conditions of experiment **4.1** on BLEScanner

	w/ callback	w/o callback
w/ surrounding BLE signals	S1	S2
w/o surrounding BLE signals	S3	-

TABLE 2: BLEAdvertiser's current consumption on Nexus9

Advertise Mode	Tx Power Level	CC (μA)	Increase (%)
LOW POWER	U.LOW	125.2	0.4
	P.LOW	437.8	1.3
	P.MEDIUM	562.8	1.6
	P.HIGH	2187.6	6.3
BALANCE	U.LOW	1625.2	4.6
	P.LOW	1593.8	4.6
	P.MEDIUM	1594.2	4.6
	P.HIGH	2250.0	6.4
LOW LATENCY	U.LOW	2187.6	6.3
	P.LOW	2594.0	7.4
	P.MEDIUM	3062.6	8.8
	P.HIGH	3187.6	9.1

and four types TxPowerLevel ("ULTRA LOW", "POWER LOW", "POWER MEDIUM" and "POWER HIGH") are existed for BLE advertising. As the evaluate of current consumption of BLE advertising, we measure current consumption about each combination of three AdvertiseMode and four TxPowerLevel. We calculate the result by the average of five trials. Similarly, there are three types ScanMode ("LOW POWER", "BALANCE" and "LOW LATENCY") for BLE scanning. Besides, Android device operates callback function when scanning all BLE signal (not only BLE signals from BLEAdvertiser). Therefore, there is a possibility that the change occurs in the current consumption depending on the amount to receive the BLE signal and the contents of the callback function. To check it, as the evaluate of current consumption of BLE scanning, we measure current consumption about each combination of three ScanMode and three situations S1-S3 shown in Table 1. We calculate the result by the average of five trials.

For measuring current consumption, we use PowerConsumptionMonitor which we developed in **3.2**. Using this Application in Nexus9, we can calculate the average of current consumption during 11.25 seconds just before. This application calculates and records the average of current consumption every 12 seconds even if a screen of the device is off. When we measure current consumption, we make the device operate nothing tasks without the exception of PowerConsumptionMonitor, BLEAdvertiser, and BLEScanner. Then, we measure the current consumption when a monitor of the device is off and a CPU of the device operate stably.

4.1.2. Result.

The result of measuring the current consumption of BLE advertising is shown in Table 2. We define the current consumption when the device operate nothing (34999.4μA) as *natural current consumption*. "CC" in Table 2 is the difference between the current consumption when advertising and natural current consumption. (In other words, it is the current consumption by BLE advertising or scanning). "Increase" is a rate of increase the current consumption

TABLE 3: BLEScanner's current consumption on Nexus9

Situation	Scan Mode	CC (μA)	Increase (%)
S1	LOW POWER	10281.6	29.4
	BALANCE	36437.8	104.1
	LOW LATENCY	70281.4	200.8
S2	LOW POWER	6500.2	18.6
	BALANCE	18875.2	53.9
	LOW LATENCY	50844.2	145.3
S3	LOW POWER	2655.6	7.59
	BALANCE	4125.4	11.8
	LOW LATENCY	17625.2	50.4

compared to natural current consumption. From this result, we find out that the current consumption of BLE is 9% of natural current consumption at most.

The result of measuring the current consumption of BLE scanning is shown in Table 3. When we analyze UUID in BLE signal, the device consumes twice the current consumption as compared with natural current consumption. Besides, the current consumption changes depending on the number of detection times of signal. Also, at the place where there are many BLE signals, the current consumption of BLE scanning is large.

4.2. Relationship of distance between devices and RSSI

The more far away from the advertising device the scanning device is, the lower RSSI of radio waves are. In BLEAdvertiser, communication distance of the signal by modifying AdvertiseMode is controllable. In BLEScanner, the device can check RSSI of scanned BLE signals.

In this experiment, we use two devices and try BLE communication while changing the distance between devices. Then we check RSSI and confirm how radio waves intensity decays. By referring data that obtained in this experiment, the developer can set the maximum distance to be included in the pass-by (pass-by distance *L*) by sifting data by RSSI.

4.2.1. Conditions.

This experiment is carried out by **A** and **B** who each have a device. First, **A** start to advertise using BLEAdvertiser with setting AdvertiseMode to LOW LATENCY, and **A** stands still while having the device at the height 1m. **B** stands a certain distance away and scan signals by BLEScanner during 20 seconds with setting ScanMode to LOW LATENCY. **A** and **B** conduct this trial about various distance (1m, each 5m of 5m -100m) between devices. Furthermore, **A** and **B** conduct this trial about each TxPowerLevel in BLEAdvertiser.

4.2.2. Result.

The experimental result about four TxPowerLevels and various distance (1m, each 5m of 5m -100m) between device are shown in Figure 3. Four curves show logarithmic approximations of results about each TxPowerLevels. The maximum distance which can be communicated by BLE is about 50m in nominal. However, we can detect pass-by

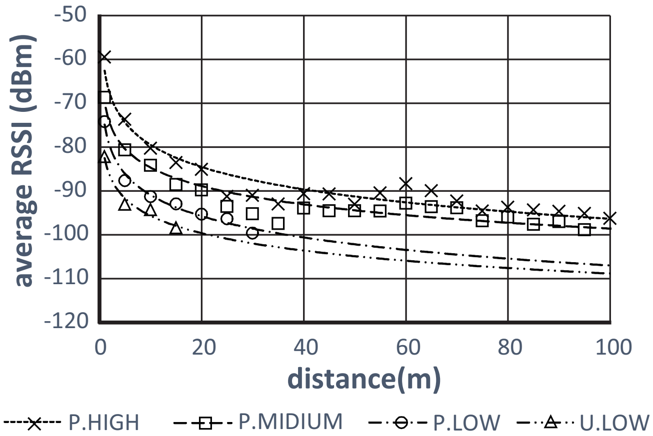

Figure 3: The distance between devices and average RSSI

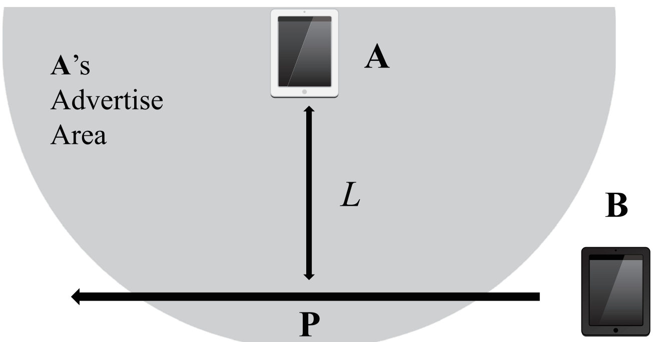

Figure 4: Conditions of experiment **4.3**

at a distance of more than 50m about POWER HIGH and POWER MEDIUM.

Average RSSI is -93dBm when the distance between devices is 5m and TxPowerLevel is ULTRA LOW. If the developer implements a program that judges BLE signals as pass-by when RSSI is over -93dBm and TxPowerLevel is ULTRA LOW, the developer can create pass-by detection mechanism that pass-by distance *L* is 5m.

4.3. Relationship of speed and detection accuracy/frequency

In common pass-by situations, a device might move at any speed. Then, it seems that there is dependence between speed and RSSI; or speed and detection accuracy about pass-by. We do not know whether this application can detect all pass-by by in the same design as application supposing pass-by with pedestrians when a developer develops the pass-by application that supposes pass-by with bikes. The developer should know how a speed of the device influence to pass-by.

4.3.1. Conditions.

In this experiment, we set pass-by distance *L* to 5m. This experiment is carried out by **A** and **B** who has a device each. **A**'s device handles BLEAdvertiser, and **B**'s device

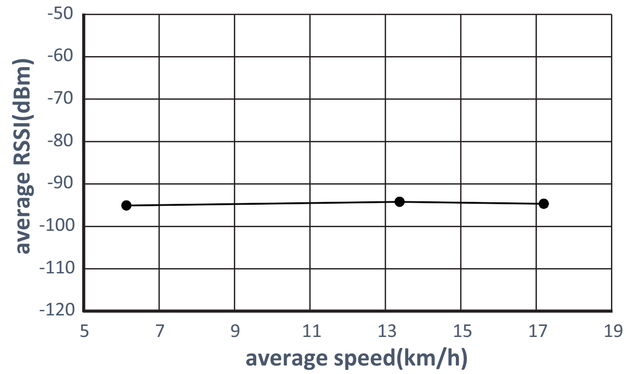

Figure 5: The average of RSSI about each transportation

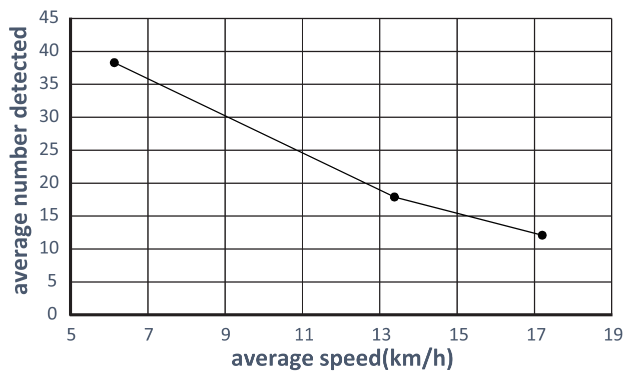

Figure 6: The averages number of scanned signal about each transportation

handles BLEScanner. In this experiment, both devices are put in rucksacks. The illustration of the method in this experiment is shown in Figure 4. At First, **A** stands still and **B** moves out of **A**'s BLE advertising range. A point that is away from **A** is defined as **P**. **B** moves at a constant rate on the line which is perpendicular to **AP** and passes on **P** to the another side of out of **A**'s BLE advertising range. In this experiment, **A** and **B** do pass-by with speeds of the walk (about 5km/h), run (about 10km/h) and bike (about 20km/h). After finishing a total of thirty trials, **A** and **B** exchange device and conduct the same experiment. Besides, TxPowerLevel in the experiment is set the smallest parameter that can detect pass-by certainly when **B** stands still at point **P**. This time, **A** and **B** set TxPowerLevel to POWER LOW.

4.3.2. Result.

The success rate of pass-by detection is over 90% about each three transportations. The following shows a result when the receiver moves. The result of analyzing RSSI about each speed is shown in Figure 5, and the average number of scanned signals is shown in Figure 6. We understand that the average of RSSI does not change mostly with all transportations. In conclusion, the influence to RSSI by speed is small enough to be ignored. Besides, the average number of scanned signals tends toward a decrease in the form being direct proportion to speeds.

4.4. Examination

The current consumption of BLE advertising is under 10% of natural current consumption with all combination. Therefore, when the developer develops power saving application, there are a few effects on power saving by setting low parameter about the setting of TxPowerLevel. Besides, because BLEScanner executes the callback function when a device scan BLE signals, the current consumption change by two elements; the content of the callback function and ScanMode. Therefore, we think that the current consumption increase in a situation that there are many people using BLE application. Developing the pass-by application, the developer should consider a situation of using the application at where there are many people.

We understand that RSSI does not change almost by changing speeds of the device. Probably, if we conduct the experiment with even greater speed, RSSI does not change too large. However, the number of scanned signals decrease as speed ups. It is considered this is cause that the time when the scanning device shortens as speed ups. Hence, the magnitude of the speed that the device can detect pass-by is finite.

All devices used in these experiments is Nexus9. If we use other devices than Nexus9, the results of these experiments are not same as this time. In the future, if we want to utilize the result of this experiment as a useful data for sonar of Pass-by Framework, it is necessary to conduct experiments using other devices and research the difference of results between types of devices.

5. Conclusion

In this study, for implementation sonar of Pass-by Framework, we aim at taking the reference data about the difference of pass-by detection behavior between settings of parameter about each proximity detection technologies. In this paper, we showed the reference data of BLE in Android device by evaluation experiment. As future challenges, we will conduct the same experiment using other Android devices or iOS devices, and we should show data that is more generic. Besides, we should consider parameters of other proximity detection mechanisms such as Ad-hoc mode of Wi-Fi. Furthermore, we will consider situations which the pass-by system is utilized while setting appropriate parameters indexed the reference data by for safety confirmation system in the disaster. Then, we should actually confirm general versatility after implementation by the method such as a simulation.

Acknowledgments

This research was partially supported by the Japan Ministry of Education, Science, Sports, and Culture [Grant-in-Aid for Scientific Research (B) (No.26280115, No.15H02701), Young Scientists (B) (No.26730155), and Challenging Exploratory Research (15K12020)].

References

[1] "Tokyo national museum - applications about "tohakunavi"," http://www.tnm.jp/modules/r_free_page/index.php?id=1467.

[2] "Streetpass communication and spotpass communication," https://www.nintendo.co.jp/3ds/hardware/features/network.html.

[3] A. Hayashi, S. Saiki, S. Tokunaga, S. Matsumoto, and M. Nakamura, "Formulating device-independent pass-by rendezvous," in *Asia-Pacific Symposium on Information and Telecommunication Technologies (APSITT2015)*, August 2015, colombo, Sri Lanka.

[4] "Bluetooth low energy — bluetooth development portal," https://developer.bluetooth.org/TechnologyOverview /Pages/BLE.aspx.

[5] A. Uehara, "Data structure of beacon," in *iBeacon handbook*. Tatsujin Press, 2014, pp. 28–30.

[6] "Measuring device power — android open source project," https://source.android.com/devices/tech/power/device.html.