Journal of Theoretical and Applied Information Technology
31st March 2022. Vol.100. No 6

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AN IN-SHIP LOCALIZATION ALGORITHM FOR CLOSE CONTACT IDENTIFICATION

QIANFENG LIN[1], JOOYOUNG SON[2]

[1]Department of Computer Engineering, Korea Maritime and Ocean University, Busan, Korea

[2](Corresponding author) Division of Marine IT Engineering, Korea Maritime and Ocean University, Korea

E-mail: [1]jmulqf@outlook.com, [2]mmlab@kmou.ac.kr

ABSTRACT

Concern about the health of people who traveled on board was raised in the COVID-19 outbreak on the Diamond Princess cruise ship. The ship narrow space offers an environment conducive to the virus spread. Close contact isolation remains one of the most important current measures to stop the rapid spread of the virus. Contacts can be identified efficiently by detecting smart devices nearby. The smartphone Bluetooth RSSI signal is significant data for positioning. The traditional indoor positioning algorithm cannot be directly applied in the mobile ship environment. It is necessary to study the indoor positioning algorithm which applies to the mobile ship environment. In this paper, we propose an in-ship localization algorithm, which can achieve indoor positioning without the fingerprint map, with an RMSE of 1.63 m.

Keywords: *Indoor Positioning, In-Ship Localization, Reference Point, Close Contact*

1. INTRODUCTION

The Japanese government ordered the passengers and crew on the Diamond Princess to start a two-week quarantine after a former passenger tested positive for COVID-19 [1]. The crew is one of the dominant groups that are more susceptible to outbreaks of this virus due to the cramped working environment [2]. The importation of viruses onboard is facilitated by regular and irregular contact with land-based populations. Living in confined spaces with limited air exchange promotes the spread of disease [3]. The safe running of any cruise ship typically necessitates the participation of multiple parties, including the ship management firm. The first stage for cruise firms is to identify any COVID-19-related dangers to their ships, crew, passengers, and other people and put in place suitable precautions [4].

For sustainable management of ship personnel under epidemic conditions, it is necessary to know where they are on board. Indoor positioning technology can help ship management personnel check the real-time location distribution of the crew. Once they are in danger, then the technology can also locate the accident location in time, and they can be quickly rescued.

In the mobile ship environment, existing positioning systems have significant application problems due to internal and external dynamic effects such as changes in sailing speed. The challenges are mainly due to arbitrary ship movements and the resulting complex effects on the indoor wireless signal. Such effects can cause constant changes in the fingerprint map, resulting in degradation of positioning accuracy.

To address these challenges, in this paper, new indoor positioning algorithms are proposed for the first time. The traditional fingerprint map localization algorithm has offline and online phases. In the new indoor positioning algorithm, the offline phase is not needed and goes directly to the online phase. This method does not depend on the fingerprint map, which reduces the accuracy error caused by the change of the fingerprint map.

2. PREVIOUS WORKS

Cruise ships have contributed to the spread of COVID-19 around the world [5]. The most critical aspect of pandemic preparation is the timely detection of close contacts on board. The location of smartphones can identify close contacts. But the environment is difficult to predict. For example, unpredictable dynamic factors can cause changes in the fingerprint map [6-7]. Such unpredictable dynamic factors such as sailing speed, acceleration motion, turning motion, weather conditions, and so on. It is inefficient and impractical to analyze and model the effects of each factor. To adapt existing positioning methods in the mobile ship environment, the model parameters need to be explored under all possible navigational conditions, which is not only very time-consuming but also leads to costly system deployment [8].

K-means clustering is a vector quantization method, originally from signal processing, that aims to divide n observations into k clusters, where each observation belongs to the cluster with the closest mean as clusters [9]. K-means is used as the extraction of representative RSSI values for comparing the localization accuracy between Average and Peak.

The beacon is an IoT device for short-distance communications based on BLE (Bluetooth Low Energy). Its advantages are low power consumption, miniaturization, wide signal range, and low cost. Although BLE beacons have advantages, their performance is not sufficiently good in terms of indoor positioning accuracy when used in ships such as passenger ships [10]. Larger error ranges of the existing indoor positioning algorithms may cause a serious problem in narrow and complex areas inside passenger ships [11]. Therefore, it is necessary to develop a novel indoor positioning algorithm with beacons to use in ship environments.

Most indoor positioning algorithms using infrastructure have two-phase processes: offline and online. For example, KNN (K-Nearest Neighbors) indoor positioning algorithm. In the offline phase, a fingerprint map is made with Peak or Average RSSI values gathered at RPs (Reference Points). In the online phase, a user position is estimated. The nearest K RPs are looked up in the fingerprint map using the Peak or Average RSSI values of beacon signals received by a user device. The Euclidean Distance among the K RPs is calculated to estimate the user position [12].

[13] proposes the key point algorithm to improve the KNN. The algorithm considers the nearest RP and the weighted centroid position of the beacons whose signals are received by the user device in the online phase. The distances between beacons and user devices are calculated. The strength values of the received signals and other information in signals are also used to get the distance.

The traditional indoor positioning algorithm is a fingerprint map approach that provides stable positioning accuracy and low positioning error in a stationary building. However, in the case of ship navigation, the fingerprint map changes according to different dynamic factors. Once the changes occur, the fingerprint map needs to be remeasured. This approach in the mobile ship environment leads to low efficiency and high error. Therefore, this paper proposes a new algorithm that does not rely on the fingerprint map in the offline phase. The localization is performed directly in the online phase and high accuracy localization results can be obtained. The new indoor positioning is called an in-ship localization algorithm.

The contribution of this paper is to change the traditional fingerprint map approach to localization, allowing high precision localization without relying on fingerprint map.

3. PROPOSED IN-SHIP LOCALIZATION ALGORITHM

Most indoor positioning algorithms using infrastructure are two-step processes: offline and online. In the offline phase, a fingerprint map of Peak or Average RSSI values gathered at RPs is made in advance. In the online stage, the nearest K RPs are looked up in the fingerprint map using the Peak or Average RSSI values of beacon signals received by a user device. The Euclidean Distance among the K RPs is calculated to estimate the user location.

However, the KNN only considers the center of K nearest RPs as the user location. The accuracy of predicted position is not sufficient to apply to complicated indoor environments like ships yet.

The key point algorithm considers the nearest RP and the weighted centroid position of the beacons whose signals are received by the user device in the online phase. The distances between beacons and user devices are calculated. The strength values of the received signals and other information in signals are also used to get the distance.

Journal of Theoretical and Applied Information Technology
31st March 2022. Vol.100. No 6

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

```
1. Receive RSSI from nearby beacons
2. Find the peak RSSI received from each beacon
3. Look up fingerprint map to find the RPs storing the same data
   as the beacon data received by user device
4. Find K nearest RPs in the fingerprint map
5. Estimate the center of RPs as user's position
```

Figure 1: KNN Algorithm

```
1. Receive peak RSSI from nearby beacons
2. Apply the KNN to get the key point P1
3. Look up the nearest beacon to the user device based on distance
   and the nearest RP (P2) to the nearest beacon
4. Calculate the weighted centroid position (P3) of the beacons whose
   signals are received
5. Get the centroid position of P1, P2, and P3 as the estimated user
   position
```

Figure 2: The Key Point Algorithm

This paper proposes an algorithm to adapt to the complex environment of ships. Unlike KNN and Key Point algorithms, the proposed algorithm in this paper does not require the construction of a fingerprint map.

The algorithm considers the nearest beacon whose signals are received by the user device in the online phase. The in-ship localization algorithm is based on the Peak value of RSSI to determine the distance between the two nearest beacons to the user. Since the position of the RP is known, the two RPs can be found if the two beacons are identified, and the user location can be estimated from the position of the nearest beacon and the position of the two RPs. However, considering that the user may be at the edge of the beacon coverage, then the RP closest to the beacon is estimated as the user location when the nearest beacon is known. The specific algorithm flow is shown in Figure 3.

Since it is necessary to calculate the location relationship between beacons and RPs. In steps 1-2, the coordinates of beacons and RPs can be stored as arrays C and R. Because beacons have limited coverage and often produce large errors at the covered boundaries. This situation should be recognized by the algorithm. In step 3, the beacon IDs of boundaries should be stored as list B. In step 4, the correspondence of the nearest RP to each beacon with arrays B and C are determined as dictionary D. The key of D is the beacon ID. The value of D is the RP ID. In step 5, the RSSI is received from the user device. In step 6, the Peak value of RSSI data in beacons is selected. This step can be described by the following equation:

$$RSSI_{beacon\ ID} = \max\left(RSSI_1, RSSI_2, RSSI_3 \ldots \ldots RSSI_N\right) \quad (1)$$

, where N is the number of RSSI received. In step 7, each beacon ID is sorted in descending order according to its Peak value. The sequence is called S. In step 8, the first-ranked beacon ID is b1. The second-ranked beacon ID is b2. In step 9, whether b1 exists in list B. If it exists, step 10 is executed. Otherwise, step 11 is executed. In step 10, the corresponding nearest rp1 in dictionary D is found based on the beacon ID. The rp1 is estimated as the user location. This step can be described by the following equation:

$$UL = rp1(x, y) \quad (2)$$

, where UL is the user location estimated by the algorithm. x is the horizontal coordinate and y is the vertical coordinate.

In step 11, the nearest rp2 of b2 in the dictionary is chosen.

In step 12, the centroid of rp1 and rp2 is calculated as P1. In step 13, the coordinates of b1 in list C are called P2. In step 14, the centroid of P1 and P2 is estimated as the user location. Steps 11-14 can be described by the following equation:

$$UL = \frac{\frac{rp1(x,y)+rp2(x,y)}{2}+b1(x,y)}{2} \quad (3)$$

1. The location list of Beacons is called C
2. The location list of RPs is called R
3. Store the serial numbers of beacons at endpoints in list B
4. Calculate the Euclidean distance between R and C to find the nearest RP for each beacon and store it as a dictionary D
5. Collect RSSI data from beacons on user devices
6. Select the peak value of RSSI data in beacons
7. Sort the beacons in descending order according to the peak value, and call the sequence as S
8. Take the first beacon number of the sequence S as b1, take the second beacon number of the sequence S as b2
9. Determine whether b1 exists in the list B, if it exists, then execute the tenth step. If not, execute step 11.
10. Find the nearest rp1 of b1 in the dictionary D, rp1 is the estimated position of the user
11. Find the nearest rp1 of b1 and the nearest rp2 of b2 in dictionary D
12. Calculate the centroid of rp1 and rp2 as P1
13. Get the coordinate information of b1 in list C called P2
14. Calculate the centroid of P1 and P2 as the estimated position of the user

Figure 3: In-Ship Localization Algorithm

Now, we show an example using Figure 4~8. The in-ship localization algorithm is running in a user device. The user device receives RSSI data from the beacons and identifies which beacon is the closest beacon to the user device based on the Peak value of each beacon RSSI data. Figure 4 shows that the user device is at the boundary of the beacon coverage. Usually, the signal reception decreases significantly when the user device is at the boundary of the beacon coverage. In the case of low reception, it is difficult to properly sequence the beacons. However, even in this case. For the nearest beacon, it is still possible to distinguish by the representative value of RSSI. To identify whether a user device is at the boundary or not, the nearest beacon ID is determined firstly, and the confirmed nearest beacon ID is compared with the boundary beacon ID that has been stored in advance. If the result is the same, then the user device is considered to be at the boundary.

First, the beacon sends a packet to the surrounding devices every 100 ms. The application on the smartphone parses the packet and calculates the RSSI value. By sorting the RSSI values within one second, the RSSI with the largest value is selected as the representative RSSI value of the beacon. Thus, as shown in Figure 4. Each of the three beacons has its RSSI representative value. By comparing the RSSI values of each beacon, the RSSI value of beacon 3 is found to be the largest. Therefore, beacon 3 is identified as the closest beacon to the user device.

Next, the next step is to identify the nearest RP to the user device. after the deployment planning of beacons and RPs is completed, each nearest RP affiliation to the beacon can be saved, and this affiliation will be stored in the form of a dictionary. The reason for this is that in the process of confirming the nearest RPs, there is no need to calculate the location relationship and compare the distance magnitude to obtain the result.

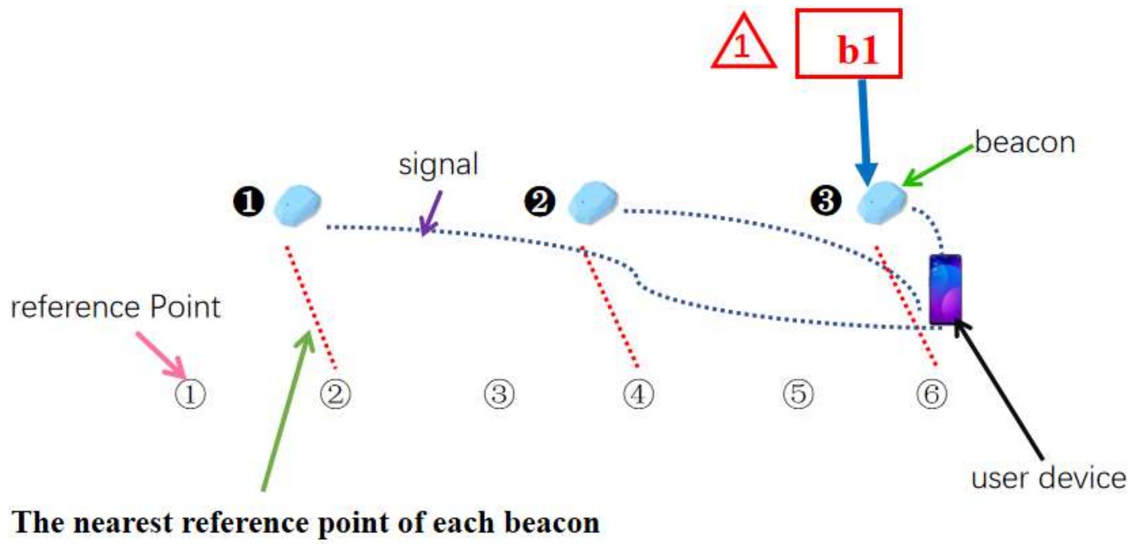

Figure 4: Processing Example of In-Ship Localization Algorithm (1)

Instead, the final result is obtained directly by querying the dictionary. This has the advantage of reducing the computational overhead and improving the efficiency of the algorithm operation. In the example of Figure 4, beacon 3 has been identified as the nearest beacon called b1, and according to the dictionary D, which has stored the affiliation between beacon and nearest RP, the ID of b1 is used as the key to query the corresponding value in the dictionary D. The final result of the query is that RP6 is the nearest RP of beacon 3 called rp1, and also is the nearest RP to the user device. The rp1 is then estimated to be the user location in the case that the user device is at the boundary. The reason for not considering the location information of b1 into the estimation process at the boundary is that including the location data of a single b1 into the calculation process without other RPs as a reference may result in greater errors.

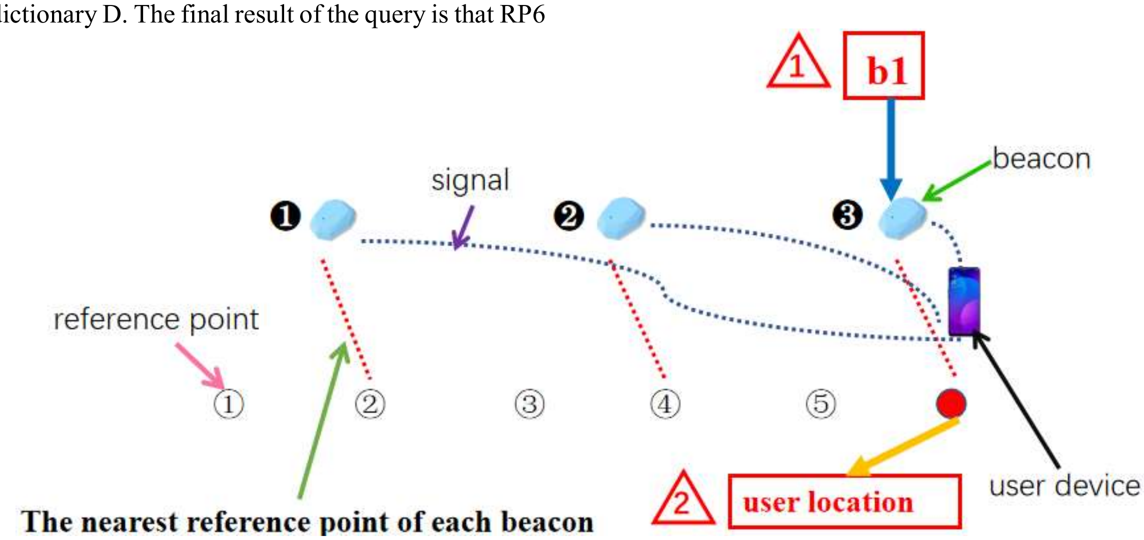

Figure 5: Processing Example of In-Ship Localization Algorithm (2)

The same process as above, starting with the example in Figure 5. First, we obtain the RSSI values of the three beacons in one second. The beacon IDs of the three beacons are compared and sorted in descending order. By comparing the beacon IDs, we can confirm whether the user device is at the boundary or not. In the example of Figure 5, the user device is not at the boundary. The first beacon ID is b1 and the second beacon ID is b2. Beacon 2 is b1. Beacon 3 is b2.

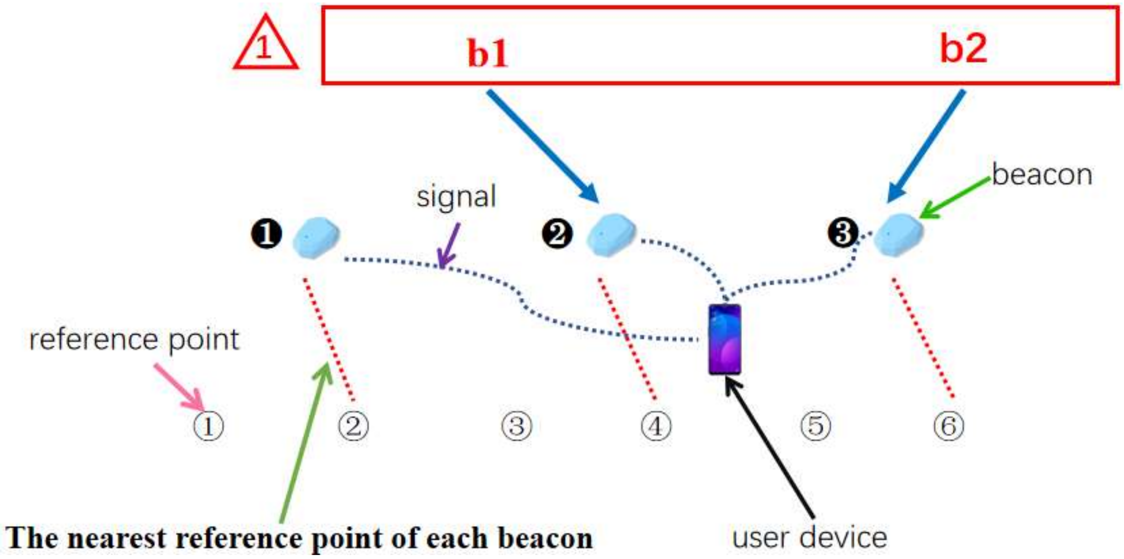

Figure 6: Processing Example of In-Ship Localization Algorithm (3)

Next, find the rp1 and rp2 corresponding to b1 and b2 respectively. Since the affiliation dictionary of beacon and recent RPs is already known. Therefore, b1 and b2 are used as keys respectively to find the corresponding value in the dictionary. Figure 7 is an example, RP4 is rp1 and RP6 is rp2. Because of the increase in signal reception, the proximity information that can be used to distinguish beacons is also increased. Since the distance between beacons is about 2.5 m. If more than 3 beacons are considered, the maximum distance between beacons will be more than 5 m. The longer the distance of user devices from beacons, the lower the signal reception, and the uncertainty of information increases. This will not only bring a computational burden but also a loss of localization accuracy. Therefore, only two beacons are considered in the proposed algorithm.

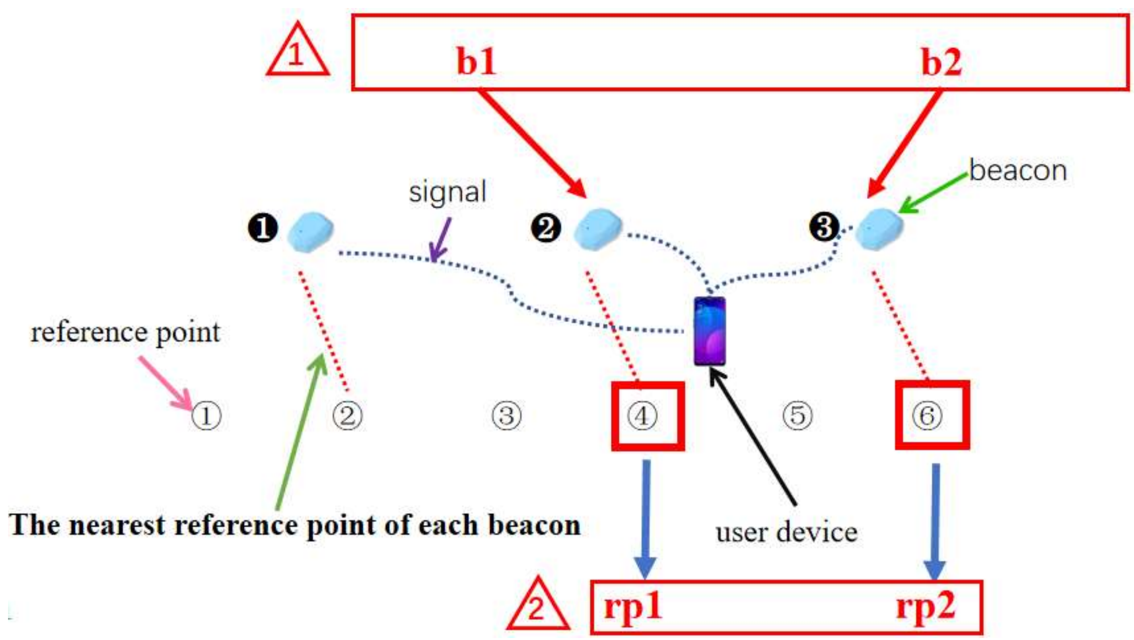

Figure 7: Processing Example of In-Ship Localization Algorithm (4)

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Finally, the user location is estimated. Figure 8 is an example. The centroids of rp1 and rp2 are computed as P1. The b1 is P2. The centroids of P1 and P2 are then estimated as user locations.

It can be seen from the whole process that the algorithm is continuously reducing the range of possible user locations. The unimportant beacons are excluded first by the Peak value of RSSI. This not only reduces the computational overhead but also enables the reduction of some errors. At this step, the range of possible user locations is reduced. In the next step, the range is further narrowed down by the nearest RPs that come to be found in the dictionary D of known beacons and nearest RP affiliations. In traditional indoor localization algorithms, it is usually necessary to compare the RSSI in the online phase with the RSSI in the fingerprint map. The RPs with similarity within a certain range are found to estimate the user location. In the algorithm proposed in this thesis, this localization method is changed. The step of finding the RP is retained and the step of generating the fingerprint map is removed. The Peak value of RSSI is used to determine the proximity of the beacon to the user device to narrow down the range of possible user locations. The RP information is used to further narrow the range. Finally, the user location is estimated by the centroid between the RP and the beacon (Figure 8).

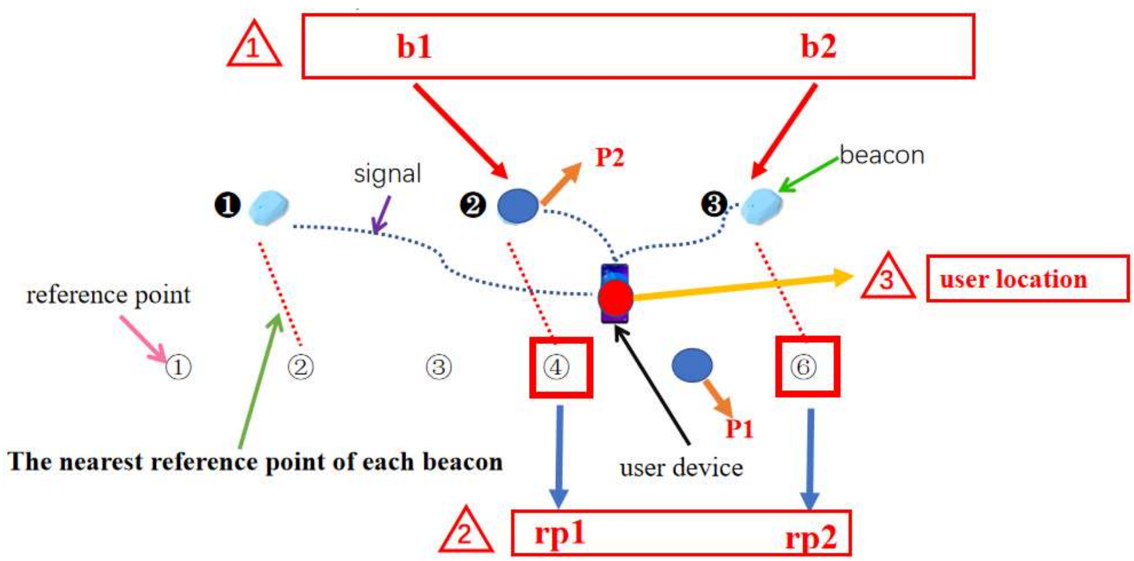

Figure 8: Processing Example of In-Ship Localization Algorithm (5)

4. PERFORMANCE EVALUATION

The test environment is set up on the HANNARA ship that is a training ship of Korea Maritime and Ocean University. In Figure 9, the red dot symbols stand for the positions of beacons, and the green dot symbols mean the RPs that are precisely determined and measured in advance. The marks (1) ~ (25) are arbitrarily chosen user locations at which the algorithm in this paper is applied to estimate the user position. The space between beacons is 2.5 m, and the space between RPs is 2 m.

The core of the in-ship localization algorithm is to find out the nearest beacon from the RSSI received from the user device. Therefore, it is necessary to find a representative value from the RSSI to determine which beacon is closest to the user device. To determine which RSSI representative value gives the best results, three different RSSI representatives are input into the algorithm, namely Average, K-means, and Peak, with Root Mean Square Error (RMSE) as the error measure. The RMSE is used to measure the deviation between the true value and the predicted value. The RMSE is more sensitive to outliers. If there is a predicted value that differs significantly from the true value, then the RMSE will be large. Therefore, RMSE is a good measure of the performance of the ship's indoor positioning algorithm. That is, the Euclidean Distance between each estimated position and the true position is calculated. Then these Euclidean Distances are summed up. Finally, by taking their average value and then opening the root, the accuracy of the indoor

Journal of Theoretical and Applied Information Technology
31st March 2022. Vol.100. No 6

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

positioning algorithm can be measured and thus a conclusion can be drawn.

The estimated errors of the 25 user locations are shown in Figure 10. By comparison, the Average case has the largest error with an RMSE of 2.08 m. K-means is used to find out different classes by clustering, and then determine the class that contains the most RSSI data, and find the center of this class as the representative value of RSSI, which has the RMSE of 2.21 m.

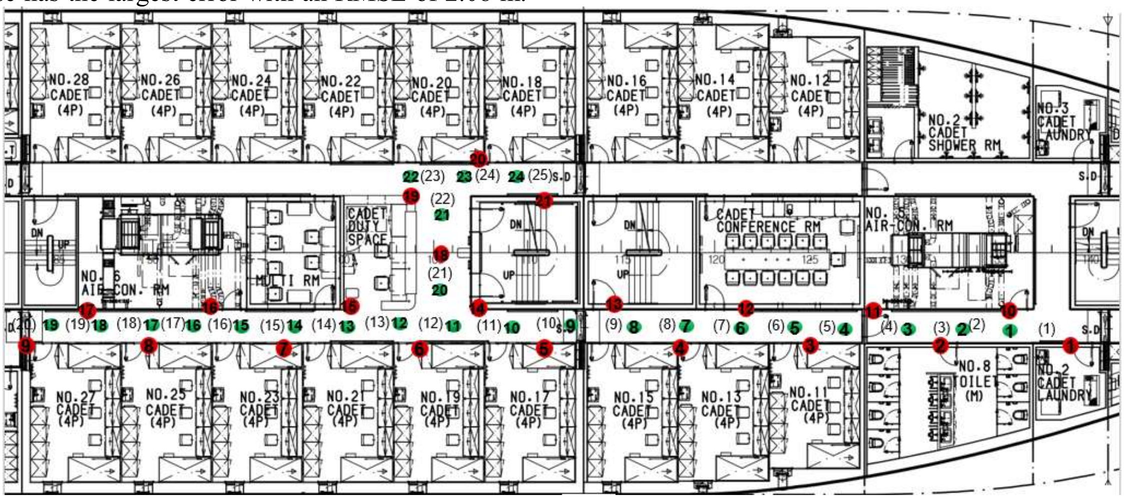

Figure 9: Experimental Environment in HANNARA ship

The best result is the Peak case with an error of 1.63 m. In general, the closer the user device is to the beacon, the stronger the RSSI. However, the complex environment of the ship can cause such a pattern to change. For example, the multi-path effect, which causes the RSSI strength to become stronger at distant places instead, is not in line with the normal rule. This also leads to errors in many indoor positioning algorithms, but such errors cannot be eliminated, only be continuously reduced. In the case of HANNARA, three different RSSI representations are compared in such a complex environment, and the Peak case is the one that minimizes the error. Therefore, the Peak value is used as the RSSI representative value in the in-ship localization algorithm.

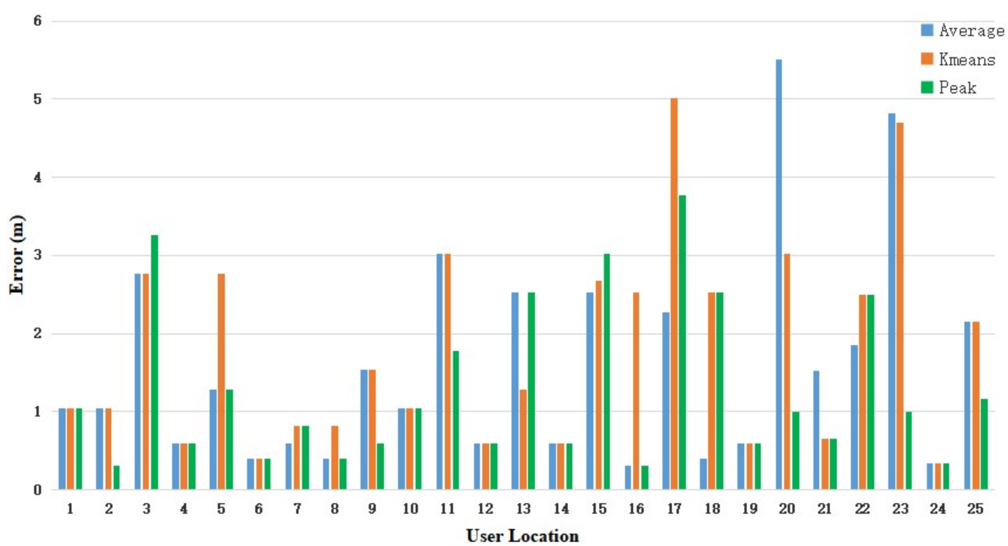

Figure 10: Errors of estimated user locations

The algorithm proposed in this thesis mainly considers the example when the user device is at the boundary. In traditional indoor positioning algorithms, there is no focus on considering the huge errors that occur when the user device is at the boundary. Therefore, this section will present how the proposed algorithm reduces the error from four examples of user devices at the boundary.

1) The first case of estimating the user location (1)

As the first case of our testing, Table 1 shows a comparison of the results of estimating the nearest beacon with different RSSI representative values at (1).

Table 1: Comparison of the results of estimating the nearest beacon with different RSSI representative values at (1)

	Average	Kmeans	Peak
Correct Beacon	10		
b1	10	10	10
b2	1	1	1

$$UL_{average} = UL_{kmeans} = UL_{peak} = \frac{\frac{RP1(x,y) + RP1(x,y)}{2} + beacon\ 10(x,y)}{2} \tag{4}$$

From Table 1, the results of Average, K-means, and Peak estimates are consistent. As shown in Figure 9, beacon 10 is not in the boundary. Therefore, beacon 1 is estimated as b2 based on the representative RSSI value of the beacon. According to the proposed algorithm flow, the centroid of b1 and b2 is calculated as P1, and then b1 is set as P2. Finally, the centroid of P1 and P2 is estimated as the user location. Here benefit from the advantages of beacon deployment. No matter beacon 10 or beacon 1, the nearest RPs is RP1.

2) The second case of estimating the user location (20)

Table 2: Comparison of the results of estimating the nearest beacon with different RSSI representative values at (20)

	Average	Kmeans	Peak
Correct Beacon	9		
b1	16	17	9
b2	17	16	

$$UL_{average} = \frac{\frac{RP16(x,y) + RP19(x,y)}{2} + beacon\ 16(x,y)}{2} \tag{5}$$

$$UL_{kmeans} = \frac{\frac{RP17(x,y) + RP16(x,y)}{2} + beacon\ 17(x,y)}{2} \tag{6}$$

$$UL_{peak} = RP19(x,y) \tag{7}$$

From Table 2, the Average estimates beacon 16 to be b1. Because beacon 16 is not in the boundary. Therefore, Average estimates beacon 17 as b2. K-means estimates beacon 17 as b1. Because beacon 17 is not at the boundary. Therefore, K-means estimates beacon 16 as b2. According to the algorithm flow, the Average estimates RP16 as rp1 and RP19 as rp2. K-means estimates RP19 as rp1 and RP16 as rp2. Regardless of Average or K-means, they both have the same P1. But the difference is that Average estimates beacon 16 as P2 and K-means estimates beacon 17 as P2. From Figure 9, we know that beacon 17 is closer to the user location than beacon 16 (20). Therefore Average causes more error than K-means. Peak estimates beacon 9 as b1. Since beacon 9 is at the boundary. Therefore, the nearest RP19 to beacon 9 is estimated as the user location. It can be seen from Figure 9 that RP19 is the closest to the user location (20). Therefore, from this example, it can be seen that Peak can bring more accuracy improvement compared to Average and K-means.

3) The third case of estimating the user location (23)

Table 3: Comparison of the results of estimating the nearest beacon with different RSSI representative v alues at (23)

	Average	Kmeans	Peak
Correct Beacon	19		
b1	6	6	19
b2	21	19	

$$UL_{average} = \frac{\frac{RP6(x,y) + RP21(x,y)}{2} + beacon\ 6(x,y)}{2} \tag{8}$$

Journal of Theoretical and Applied Information Technology
31st March 2022. Vol.100. No 6

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$UL_{kmeans} = \frac{\frac{RP6(x,y) + RP19(x,y)}{2} + beacon\ 6(x,y)}{2} \tag{9}$$

$$UL_{peak} = RP22(x,y) \tag{10}$$

From Table 3, Average and K-means estimate beacon 6 as b1. The beacon 6 is not at the boundary. Therefore, Average estimates beacon 21 as b2. K-means estimates beacon 19 as b2. From Figure 9, it can be seen that beacon 19 is closer to the user location than beacon 21 (23). Therefore, K-means introduces less error than Average. Peak estimates beacon 19 as b1. Because beacon 19 is at the boundary. Therefore, the nearest RP22 to beacon 19 is estimated as the user location. From Figure 9, we know that RP22 is very close to the user location (23). From this example, it can be seen that the improvement in accuracy brought by the Peak is very obvious.

4) The fourth case of estimating the user position (25)

Table 4: Comparison of the results of estimating the nearest beacon with different RSSI representative values at (25)

	Average	Kmeans	Peak
Correct Beacon	21		
b1	20	20	20
b2	19	19	21

$$UL_{average} = UL_{kmeans} = \frac{\frac{RP20(x,y) + RP19(x,y)}{2} + beacon\ 20(x,y)}{2} \tag{11}$$

$$UL_{peak} = \frac{\frac{RP20(x,y) + RP21(x,y)}{2} + beacon\ 21(x,y)}{2} \tag{12}$$

From Table 4, Average, K-means, and Peak estimate beacon 20 as b1. The beacon 20 is not at the boundary. Therefore, Average and K-means estimate beacon 19 as b2.

In contrast, Peak estimates beacon 21 as b2. From Figure 9, we can see that beacon 21 is closer to the user location (25), which shows that Peak significantly improves the accuracy compared to Average and K-means.

From the above four examples, Average, K-means, and Peak are analyzed separately. Therefore, in the proposed algorithm, the Peak of RSSI is used as a representative value to identify the nearest beacon.

In this paper, the proposed algorithm is compared with two other indoor localization algorithms. They are KNN and Key Point algorithms. Both algorithms use a fingerprint map to find the nearest RP, but the Key Point algorithm takes into account the nearest beacon location and is used to estimate the user location with less error than KNN. The in-ship localization algorithm determines the nearest beacon to the user device based on the user RSSI and determines the nearest RP to the user device based on the known relationship between the beacon and the RP. By reducing the uncertainty error through known information, the in-ship localization algorithm shows better results in Figure 11. The RMSE of KNN, Key Point, and the In-Ship Localization algorithm are 2.04 m, 2.21 m, and 1.63 m respectively. Among them, the in-ship localization algorithm has the highest accuracy and shows the best performance in complex ship environments.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

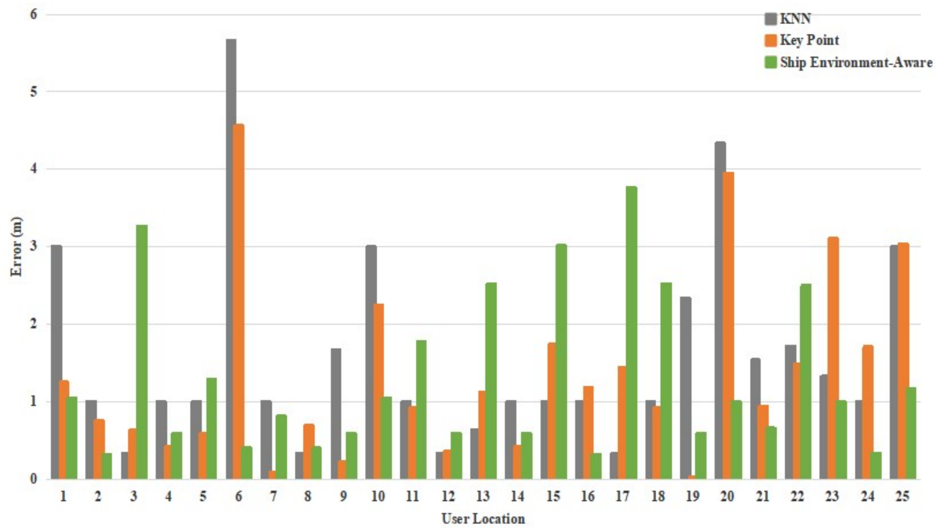

Figure 11: Comparison of error results of different indoor positioning algorithms

5. CONCLUSION

There are complex signal reflection and scattering phenomena in the ship environment. the reflection and scattering of the signal can have a serious impact on the stability of the fingerprint map, which in turn leads to an increase in the localization accuracy error. In this paper, we propose an in-ship localization algorithm for the mobile ship environment, which can achieve high accuracy localization without comparing fingerprint map. Because of the impact of different RSSI representative values on the accuracy, it is necessary to find the RSSI representative value with the highest accuracy. The accuracy of the algorithm is compared with three different RSSI representative values, namely Average, K-means, and Peak. It is found that the Peak value has the highest accuracy. Therefore, the Peak value is used as the representative value of RSSI in this algorithm. In addition, this paper also compares two different indoor positioning algorithms, that is, KNN and Key Point. RMSE is the standard for measuring indoor positioning accuracy. Their RMSE is 2.04 m and 1.79 m, respectively.

The results show that the in-ship localization algorithm has the highest accuracy. This algorithm can achieve high accuracy localization without fingerprint map, and the RMSE is 1.63m, which meets the localization requirements in the mobile ship environment.

The proposed method in this paper does not require the generation of a fingerprint map. In turn, it can reduce the deployment and maintenance cost of ship indoor positioning. It enables the ship's indoor positioning system to be applied to different types of ships faster and more effectively. Improve the productivity and safety performance of ships. In the future, based on the ship's indoor positioning, many applications can be extended around the ship's passengers, such as indoor navigation, escape route planning, etc.

6. ACKNOWLEDGEMENT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (grant number: 2021R1I1A3056125).

REFERENCES:

[1] Batista, B.; Dickenson, D.; Gurski, K.; Kebe, M.; Rankin, N. Minimizing disease spread on a quarantined cruise ship: A model of COVID-19 with asymptomatic infections", Mathematical biosciences, Vol. 329, No. 2020, 2020, pp. 108442-108452

[2] Fanoy, E.; Ummels, A. E.; Schokkenbroek, V.; Van Dijk, B.; Wiegmans, S.; Veenstra, T.; Van Der Eijk, A.; Sikkema, R.; De Raad A, "Outbreak of COVID-19 on an industrial ship", International Maritime Health, Vol. 72, No. 2, 2021, pp. 87-92

[3] Von Münster, T.; Kleine-Kampmann, S.; Hewelt, E.; Boldt, M.; Schlaich, C.; Strölin, P.; Oldenburg, M.; Dirksen-Fischer, M.; Harth, V,"When the ship comes in–COVID-19 Ausbrüche an Bord von Frachtschiffen im

Hamburger Hafen", Flugmedizin· Tropenmedizin· Reisemedizin-FTR, Vol. 28, No. 1, 2021, 13-24

[4] Plucinski, M. M.; Wallace, M.; Uehara, A; Kurbatova, E. V.; Tobolowsky, F. A.; Schneider, Z. D; Ishizumi, A.; Bozio, C.H, "Coronavirus disease 2019 (COVID-19) in Americans aboard the Diamond Princess cruise ship". Clinical Infectious Diseases, Vol. 21, No. 10, 2021, pp. 448-457

[5] Klein, N, "International law perspectives on cruise ships and covid-19", Journal of international humanitarian legal studies, Vol. 11, No. 2, pp. 282-294.

[6] Azimi, P.; Keshavarz, Z.; Laurent, J. G. C, "Mechanistic transmission modeling of COVID-19 on the Diamond Princess cruise ship demonstrates the importance of aerosol transmission", Proceedings of the National Academy of Sciences, Vol. 118, No. 8, 2021, pp. 1-8

[7] Mizumoto, K.; Chowell, G, "Transmission potential of the novel coronavirus (COVID-19) onboard the Diamond Princess Cruises Ship", Infectious Disease Modelling, Vol. 5, No. 2020, 2020, pp. 264-270

[8] Chen, M.; Liu, K.; Ma, J, "MoLoc: Unsupervised fingerprint roaming for device-free indoor localization in a mobile ship environment", IEEE Internet of Things Journal, Vol. 7, No. 12, 2020, pp. 11851-11862

[9] Wagstaff, K., Cardie, C., Rogers, S, "Constrained k-means clustering with background knowledge", Icml, Vol. 1, No. 1, 2001, pp. 577-584

[10] Hoshi, H.; Ishizuka, H.; Kobayashi, A.; Minamikawa, A, "An indoor location estimation using BLE beacons considering movable obstructions", In Proc. 2017 Tenth International Conference on Mobile Computing and Ubiquitous Network, IEEE(Japan), October 3-5, 2017, 1-2

[11] Chen, M.; Liu, K.; Ma, J, "SWIM: Speed-aware Wi-Fi-based passive indoor localization for mobile ship environment", IEEE Transactions on Mobile Computing, Vol. 20, No. 2, 2021, pp. 765-779

[12] Dongjun Kim, and Jooyoung Son, "A weighted preliminary cut-off indoor positioning scheme based on the similarity between peaks of RSSI", Journal of Korea Multimedia Society, Vol. 21, No. 7, 2018, pp. 772-778

[13] Lin, Q.; Son, J, "A Key Point Algorithm Improving Indoor Positioning Accuracy Based On Beacons". Journal of Theoretical and Applied Information Technology", Vol. 98, No. 17, 2020, pp. 3438-3448

[14] Lin, Q.; Son, J.; Xiao, D, "A ship environment-aware indoor positioning algorithm for passenger safety and health". 2021 Korean Multimedia Association Spring Conference, Korea, November 26-27, 2021, pp. 17-19