Study of Semantic Map Matching using Bluetooth Low Energy Sensing in Indoor Environment

Hani Ramadhan (201793254) and Yoga Yustiawan (201783629)

***Abstract*—In recent development of technology-enabled buildings, the use of the Bluetooth Low Energy (BLE) beacon has gained a lot of attention for obtaining the positions of objects in indoor spaces these days due to its stability. However, the position estimation should be appropriate for each use case, such as next item recommendation in museum. In case of the next item recommendation in museum, it is not recommendable to explore the visitor's position in finer-than-meters granularity. Then, the state space of visitors' position should be limited. To overcome that constraint, we propose a beacon-based multivariate hidden markov model map matching technique namely bMHMM utilizing the floor plan to detect users' semantic position using BLE beacons and simple state space design in smartphones. Then, a modified k-Nearest Neighbor (KNN) method is also proposed to infer a semantic position according to the state space model. The experiment results depict that the effectiveness of the two proposed methods is still not satisfying due to the small experiment environment and our arbitrary placement of BLE beacons. Therefore, a future work is proposed to obtain higher accuracy by expanding the experiment in larger environment and conducting a more detailed study in the placement of BLE beacons.**

***Index Terms*—Map Matching, Bluetooth Low Energy (BLE) Beacons, HMM, KNN, Indoor Positioning**

I. Introduction

INDOOR positioning is commonly used in technology-enabled buildings to enhance the visitors' experience. It helps to monitor and improve the visitors' experience based on their position. However, it is very challenging to accurately obtain the position in indoor environment in terms of finer granularity than meters. Instead of that, the less precise positioning might be useful based on several use cases, e.g. recommendation of next item to visit in museum, hotspot detection, or semantic analysis of visitors' behavior. Those use cases may not apply to distance granularity less than several meters. Thus, as long as the position of visitor is captured in a certain region of proximity, we can associate visitor's position to the item. Designing this concept of proximity is easier than to determine the exact position. Thus, a well devised state space will be beneficial to determine the semantic visitors' position to a certain objects or other spaces.

On the other hand, the research development on indoor positioning has been rapidly increasing as more and more positioning devices such as GPS [1], Wi-FI [2] and RFID readers [3] become available. The Bluetooth Low Energy (BLE) beacons are known to have more stable radio-field intensity measuring method to conduct object positioning than Wi-Fi and other signals.

Hani Ramadhan was with the Department of Big Data, Pusan National University, Pusan, South Korea; Corresponding e-mail: hani042@pusan.ac.kr.

Map matching is famously known to obtaining the position of detected objects based on known measurements. There are several methods using HMM (Hidden Markov Model) based techniques such as proposed in [4] and IR-MHMM [3]. However, these map matching methods mostly are conducted on trajectory data generated from GPS data and RFID data. Therefore, we are interested to incorporate the well-known HMM-based map matching technique to get semantic positioning results from signal data (Received Signal Strength Indicator (RSSI)) generated from BLE beacons. On the other hand, a k-Nearest Neighbor-based method [5] can also be used to separate signal measurements within different places.

HMM-based technique has been widely known as the prominent method for reducing the uncertainties in the trajectory data. In this paper, we propose a beacon-based multivariate hidden markov model map matching algorithm (bMHMM) utilizing a floor plan that aims to obtain semantic positions of detected objects using BLE beacons. This proposed algorithm incorporates the HMM-based technique with the state space modelling for capturing the uncertainties from signal data from BLE beacons. The BLE beacon tracking problem can be solved by two stochastic processes. The underlying Beacon area from one semantic location to another semantic location is a stochastic process that is not directly observable (*hidden*). The observation process that generates the beacon signal observations dependent the current object's location. Thus, given those BLE beacons' observations, we can use a HMM model to infer the semantic position of a visitor. In addition, we also propose a modified k-Nearest Neighbor method to infer a semantic position in accordance with state space model. [6]

The structure of the following sections of this report consists of related works, study of BLE signals, proposed method, experimental setup, discussion, and conclusion.

II. Related Works

In this section, the related works are briefly reviewed. Indoor positioning is very known as a quite challenging task due to several constraints in indoor spaces. One of the several issues that are still continued to be studied is obtaining the location or position of objects or people in indoor spaces, since such GPS devices that are widely used for gaining accurate location in outdoor spaces does not work very well in indoor spaces. Numerous methods utilizing various devices have been proposed to tackle this issue. Wi-Fi based localization technique is exercised with additional techniques of enhanced PDR (Pedestrian Dead Reckoning) technology & particle technique to conduct indoor positioning in [1]. Another [7] work studied quite similar technique but utilizes the image processing method to match PDR trajectory with all major possible trajectories based on Hough tranform and Harris corner detection. The Wi-Fi indoor positioning method called radius-based domain clustering (RDC) [8] is also proposed by using all available access points (APs) to estimate the position of a target point of interest to avoid the issue of AP selection. It can inferred from these three known methods that the RSS (Received Signal Strength) based methods is quite promising to locate objects or people in indoor spaces.

The RSS based indoor positioning methods gain a lot of attention due to its low cost and ease of deployment characteristics. RSS based indoor positioning methods also can be applied using other devices such as Bluetooth Low Energy (BLE) beacons. BLE beacons are widely becoming the prominent and stable devices with numerous methods which are developed to locate objects in indoor spaces. The PDR based positioning technique is also proposed by utilizing BLE beacons and gyroscope sensors but still gyroscope sensors have drift problem [9]. One work [10] depicts the efforts to maximize the stable BLE based positioning by designing the likelihood function in their proposed particle filter and to minimize the accumulating positioning errors caused by used smartphone sensors by introducing obstacle information given by the floor map. However, utilizing BLE beacons with smartphone sensors still shows inefficient performance. Another study [11] of using only BLE beacons for locating objects in indoor environment later shows the effective performance by proposing BST-Matching method based on ST-Matching that is usually applied for global map matching methods for GPS. Filtering by threshold is performed in this study to avoid detecting BLE beacons with low reliability. However, the proposed BST-Matching method in this study is relatively sensitive to the threshold, thus it is not suitable for practical use.

Furthermore, the map matching problem is intensively studied through various kinds of methods. One widely-known method such Hidden Markov Model (HMM) shows remarkable performance for GPS data in outdoor spaces [4]. Then, there is a study [3] of indoor tracking and cleansing technique using HMM for RFID data. This study indicates that using HMM is very effective, efficient and robust in solving indoor positioning problem. Despite HMM approach shows the good efficiency and effectiveness, the studies of implementing HMM based approach that utilizes BLE beacons to solve the indoor positioning issue are still quite rare. Therefore, this paper aims to study the efficiency and effectiveness of HMM based approach to locate objects or people in indoor spaces utilizing BLE beacon data.

Another study of indoor positioning using BLE signal utilizes k-Nearest Neighbor (KNN) [5] as its inference technique. Its indoor positioning technique uses weighted KNN with adaptive bandwidth mean based and achieves around 1-2 meters positioning error. However, we are interested in how can the kNN can be applied to produce indoor semantic trajectory as it exposed to a state space design. Hence, we also propose a kNN based semantic indoor positioning that takes account the state space model based on the indoor floor plan.

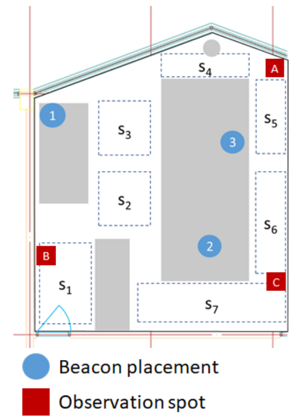


Fig. 1. Floor plan of our laboratory and the details of the deployment and semantic area definition

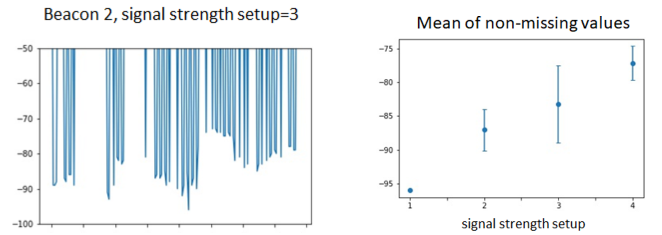
TABLE I
THE DENSITY OF BLE READINGS OF EACH OBSERVATION SPOT USING 3 BEACONS FOR 30S

Strength	Spot	Beacon 1	Beacon 2	Beacon 3
1	A	0.12	0.08	0.00
	B	0.00	0.00	0.00
	C	0.01	0.21	0.01
2	A	0.46	0.17	0.17
	B	0.27	0.14	0.13
	C	0.31	0.40	0.22
3	A	0.43	0.27	0.27
	B	0.36	0.30	0.22
	C	0.37	0.47	0.28
4	A	0.46	0.31	0.36
	B	0.41	0.48	0.27
	C	0.46	0.44	0.30

III. OBSERVATION OF BLUETOOTH LOW ENERGY RSSI

Before we discuss about proposed method, we will present the characteristics of bluetooth signal strength reading in an indoor environment. This observation takes place in our 7x9 m^2 laboratory which plan is depicted in Figure 1. The red square defines the observation spot of the BLE signal reading, the blue circle marks the BLE beacon deployment position, and the dashed rectangle specifies the semantic area definition, which later will be used in state space modelling.

We provide the reading observation for 30 seconds by staying at spot A, B, C density in Table I based on different signal strength setups. The density means the number of non-zero readings divided from all samples. The sample interval used in this observation is 200 ms. The behaviour of at spot C using beacon 2 is depicted in Figure 2. We see that none of the beacons produced more than 0.5 density, which means for more than the half of the observation period (not consecutively) the RSSI values are non-existant. Yet, we see that the behavior of the BLE signal readings is rather unstable, simultaneously exist and missing. And as the power of the signal strength setup increase, the signal strength reading increased and less vary. Thus, we will propose a windowed aggregation as preprocessing method to solve this reading stability issue.



(a) (b)

Fig. 2. Observation at spot C using beacon no. 3 (a) the captured signal strength for 30 s (b) the mean and variance based on signal strength setup

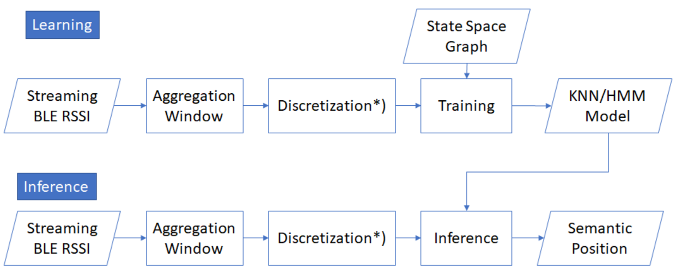


Fig. 3. The proposed method using either HMM or KNN based model

IV. Proposed Method

In this section, we will provide the overview of our map matching method in Figure 3. Note that the KNN-based method does not consider discretization for its preprocessing, as neglecting slight difference that caused by discretization may lead to different inference.

A. Preprocessing: Aggregation Window and Discretization

Given the unstable behaviour of the BLE readings, it is unlikely to use the streaming BLE RSSI input directly. Thus, to stabilize the behaviour of the signal strength reading we will use aggregation window strategy to gather common signal strength of consecutive samples. The aggregation window can be used in the shifting and sliding style, whereas the aggregation function also differs (SUM, MAX, MEAN, etc). Parameters of the aggregation window are discussed later in experiment section.

The discretization produces the discrete values of observation that used for HMM based method input. For a range of -1 to -100 of RSSI values, the discretization groups an interval of values to several number of bins C. For example, using 5 bins ($C = 5$) as discretization parameter will divide -1 to -20 to bin #1 and so on until bin #5. The discretization produces bin #0 to indicate the non-existant reading.

A raw sequence of beacon observation of length T_b is symbolized as $X = \{x^{(1)}, x^{(2)}, ..., x^{(T_b)}\}$. After aggregation, the length T_b reduces to T, thus an aggregated sequence can be written as $\overline{X} = \{\overline{x}^{(1)}, \overline{x}^{(2)}, ..., \overline{x}^{(T)}\}$. The discretization converts an aggregated sequence $\overline{X}$ to an discretized sequence $D = \{d^{(1)}, d^{(2)}, ..., d^{(t)}\}$. Any element of the raw, aggregated, and discretized sequence of each timestamp t always contains M observation values from M beacons, thus $e^{(t)} = \{e_1^{(t)}, e_2^{(t)}, ..., e_M^{(t)}\} | e^{(t)} \in \{x^{(t)}, \overline{x}^{(t)}, d^{(t)}\}$. Then, given the characteristic of RSSI value range and the discretization, several constraints for each observation of beacon m at timestamp t are introduced: $-100 \leq x^{(t)}, \overline{x}^{(t)} \leq 0$ and $0 \leq d_m^{(t)} \leq C$.

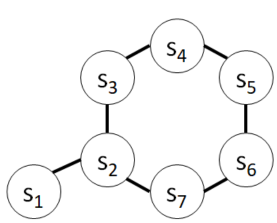


Fig. 4. The state space model of semantic locations in our laboratory

B. State Space Design

The proposed methods requires state space models to determine the hidden states and the transition from a semantic location to another. We model the state space from floor plan given a certain proximity of an exhibition object. In this case, the states can be simply defined as area covered by a distance to an exhibition object. Thus, we can define states with the intended semantic of proximity to the exhibition objects. On the other hand, it can be defined also as another preferred areas such as segmented corridors that may give possible information of congestion or resting area. Thus, the knowledge about the floor plan is highly beneficial to determine the state space. An example of a state space that based on our laboratory semantic region definition is depicted by graph in Figure 4. In the implementation, an adjacency matrix represents the state space model.

C. The Proposed bMHMM Method

Our proposed method, bMHMM (beacon-based Multivariate Hidden Markov Model), works both to model uncertainties for transition and emitting probability of each semantic location and the discretized BLE RSSI streaming data. We define our bMHMM as beacon-based Hidden Markov Model $\lambda = (S, O, A, B, \pi)$ where S, O, A, B, π are hidden states, all possible combination of observations set, transition matrix, emission matrix, and initial transition probability as in generic HMM definition.

The transition probability matrix A considers the transition based on state space model. In our example, movement from s_1 to s_6 should not be possible. Thus, any impossible movement in the state space model from state i to j is represented by $a_{ij} = 0$. Then, the emission probability matrix B refers to the emission of a single hidden state given multiple M observation source with $C + 1$ values. Here, C refers to the number of discretization bin plus one (for bin #0). Thus, the possible combination of observations space counts up to $|S| \times M \times (C + 1)$. A single entry of B, which is $b_{imk} = P(s_i^{(t)} | d_m^{(t)} = k)$, gives the probability of a hidden state s_i occurs given a discretized RSSI value k from beacon m at time t.

The learning process of bMHMM could use two different approaches: supervised and unsupervised. Both approaches are used regarding whether the hidden states of the sequences are already known (labeled) and not, respectively.

The supervised method uses the known hidden states (ground truth) of the sequence to determine the probability of matrix A and B. Each entry a_{ij} and b_{imk} is computed easily by counting the occurrence of hidden states transition from state i to j and when a hidden state i emitting a value k from observation m for entire sequence. Both formulas are depicted in Eq. 1 and Eq. 2.

$$a_{ij} = \frac{\sum_{t=1}^{T-1} 1|s^{(t)} = i \wedge s^{(t+1)} = j}{\sum_{t=1}^{T-1} \sum_{p \in S} 1|s^{(t)} = i \wedge s^{(t+1)} = p} \tag{1}$$

$$b_{imk} = \frac{\sum_{t=1}^{T-1} 1|s^{(t)} = i \wedge d_m^{(t)} = k}{\sum_{t=1}^{T-1} \sum_{c=0}^{C} 1|s^{(t)} = i \wedge d_m^{(t)} = c} \tag{2}$$

The unsupervised learning of bMHMM does not use the knowledge of ground truth to learn the parameters of HMM. Based on random initialization for each probability matrix, the learning process works in term of Expectation Maximization (EM). Given a random model θ, it will predict a new probability of π, A, and B. Because our model has been modified to multiple observation fashion, thus the equations of the known EM is also modified. The formula for forward probability, backward probability, and updated entries of $\hat{A}$, $\hat{B}$, and $\hat{\pi}$ are depicted in Eq. 3, 4, and 5 respectively. Then, to generate a single bMHMM model, we use the algorithm 1, the learn function in line 9 of algorithm 1 refers to unsupervised learning of bMHMM repeated I times or until convergence reached.

$$\alpha_i^{(1)} = 0; \;\; \alpha_i^{(t)} = \prod_{m=1}^{M} b_{imc} \sum_{j \in S} a_i^{(t-1)} a_{ji} \;\mid\; c = d_m^{(t)} \tag{3}$$

$$\beta_i^{(T)} = 1; \;\; \beta_i^{(t)} = \sum_{j \in S} a_{ij} \beta_j^{(t+1)} \prod_{m=1}^{M} b_{jmc} \;\mid\; c = d_m^{(t)} \tag{4}$$

$$\begin{aligned} \hat{\pi}_i &= \frac{\alpha_i^{(1)} \beta_i^{(1)}}{\sum_{j \in S} \alpha_j^{(1)} \beta_j^{(1)}}; \\ \hat{a}_{ij} &= \sum_{t=1}^{T-1} \frac{\alpha_i^{(t)} \beta_j^{(t+1)} a_{ij} \prod_{m=1}^{M} b_{jmc} \;\mid\; c = d_m^{(t)}}{P(D|\theta)}; \\ \hat{b}_{imc} &= \sum_{t=1}^{T} \frac{\alpha_i^{(t)} \beta_i^{(t)} \;\mid\; c = d_m^{(t)}}{P(D|\theta)} \end{aligned} \tag{5}$$

Then, to determine the semantic location from an unlabeled discretized sequence D, a decoding algorithm is required. We use modified viterbi algorithm to predict the semantic location $s^{(t)}$ in streaming fashion as depicted in algorithm 2 and 3. We only predict a single state from a single observation vector to fit the streaming fashion. If we predict more than one hidden state, it is possible to infer inconsistent semantic trajectories from multiple consecutive observations. For our case, as example, predicted states at time $\{t, t+1\}$ are $\{s_1, s_2\}$ and time $\{t, t+1\}$ are $\{s_4, s_3\}$, which is nonsense considering our state space design.

Algorithm 1: Generate bMHMM Algorithm

Input : - Training Sequences $TS = \{D1, D2, ..., DQ\}$, consists of Q discretized sequences
- S hidden states set
- O all possible observations set
- State Space Design G, $|S| \times |S|$ adjacency matrix
- Max Iteration I, the maximum learning iteration
- Random Restart R, the number of initial randomization used to learn
- Discretization bin C

Output: bMHMM Model $\theta = (S, O, A, B, \pi)$

1 **Function** *Generate bMHMM(*TS*,* G*,* I*,* R*,* C*)*
2   $\pi^* \leftarrow \{0\}^{|S|}$;
3   $A^* \leftarrow \{0\}^{|S| \times |S|}$;
4   $B^* \leftarrow \{0\}^{|S| \times M \times (C+1)}$;
5   $learn\_rate \leftarrow \frac{1}{R}$;
6   **while** $R \neq 0$ **do**
7     $\pi \leftarrow \text{rand}(0,1)$ where $\sum_{i=1}^{N} \pi_i = 1$;
8     $a_{ij} \in A \leftarrow \text{rand}(0,1)$ where $G_{ij} = 1 \wedge \sum_{j=1}^{N} a_{ij} = 1$;
9     $A', B', \pi' \leftarrow learn(S, O, A, B, \pi, TS, I)$;
10     $A^* \leftarrow A^* + learn\_rate \times A'$;
11     $B^* \leftarrow B^* + learn\_rate \times B'$;
12     $\pi^* \leftarrow \pi^* + learn\_rate \times \pi'$;
13     $R--$;
14   **end**
15   Return $\theta = (S, O, A^*, B^*, \pi^*)$;

Algorithm 2: Streaming Viterbi bMHMM

Input : - DiscretizedReading $d^{(t)} = \{d_1^{(t)}, d_2^{(t)}, ..., d_M^{(t)}\}$
- bMHMM Model $\theta = (S, O, A, B, \pi)$
- $isBeginning$: information whether $\mathbf{d}^{(t)}$'s timestamp is the beginning of the sequence
- $Vprev$: the value of previous timestamp's likelihood of each state, length N, optional when $isBeginning$ is $True$

Output: predictedState $= s^{(t)}$

Function *StreamingViterbi(DiscretizedReading* $d^{(t)}$*, Double* V_{curr}*, Double* V_{prev}*)*
  $(S, O, A, B, \pi) \leftarrow \theta$;
  if $isBeginning \; is \; True$ **then**
    $V_{prev} \leftarrow \pi$;
  end
  $V_{curr} \leftarrow \{0\}^{|S|}$;
  $s^{(t)} \leftarrow null$;
  ComputeViterbi$(d^{(t)}, V_{curr}, V_{prev})$;
  $s^{(t)} \leftarrow argmax_{i \in S} V_{curr_i}$;
  Return $s^{(t)}$;

Algorithm 3: Compute Viterbi

Input : - DiscretizedReading $d^{(t)} = \{d_1^{(t)}, d_2^{(t)}, ..., d_M^{(t)}\}$
- $V_{curr}, \{[0,1]\}^{|S|}$, current timestamp likelihood
- $V_{prev}, \{[0,1]\}^{|S|}$, prev. timestamp's likelihood

Procedure *ComputeViterbi(DiscretizedReading* $d^{(t)}$*, Double* V_{curr}*, Double* V_{prev}*)*
  foreach $i \in S$ **do**
    $V_{curr_i} = \prod_{m=1}^{M} b_{imc} \max_{j \in S} a_{ji} V_{prev_j} | c = d_m^{(t)}$;
  end
  Normalize $V_{curr_i} = \frac{V_{curr_i}}{\sum_{j=1}^{N} V_{curr_j}}$;

D. The Proposed Modified KNN Method

We also propose another method that benefits the simple k-Nearest Neighbor (KNN) inference but applies the designed state space model. KNN performs lazy learning as it does not produce any discriminative function in training, it just stores all the training data as reference to incoming test data. We modify the reference dataset of KNN $\{(\overline{x}^1, s^1), (\overline{x}^2, s^2), ..., (\overline{x}^Q, s^Q)\}$ by filtering only the neighboring labelset $NL(s^{(t-1)})$ of previous timestamp's predicted state $s^{(t-1)}$. The neighboring labelset $NL(s)$ is all the labels that are the neighbors of state s given the state space design, including itself. In our example, state s_2 has neighboring labelset $NL(s_2) = \{s_1, s_2, s_3, s_7\}$. After that, we perform simple k-Nearest Neighbor using K, input readings $\overline{x}^{(t)}$, and the filtered dataset that returns the predicted semantic location $s^{(t)}$.

Algorithm 4: Filter kNN

Input : - AggregatedReading $\overline{x}^{(t)} = \{\overline{x}_1^{(t)}, \overline{x}_2^{(t)}, ..., \overline{x}_M^{(t)}\}$
- $s^{(t-1)}$, previous timestamp's predicted state, $null$ when $t = 0$
- $TS = \{(\overline{x}^1, s^1), (\overline{x}^2, s^2), ..., (\overline{x}^Q, s^Q)\}$ Q aggregated readings, used as reference dataset
- K, number of considered nearest neighbor

Output: predictedState $= s^{(t)}$

Function *FilterKNN(AggregatedReading* $\overline{x}^{(t)}, s^{(t-1)}, TS, K$*)*
  $s^{(t)} \leftarrow null$;
  $ds \leftarrow \{\}$;
  if $s^{(t-1)}$ *is not null* **then**
    $ds \leftarrow TS.filterBy(NL(s^{(t-1)}))$
  else
    $ds \leftarrow TS$
  end
  $s^{(t)} \leftarrow kNearestNeighbor(\overline{x}^{(t)}, ds, K)$;
  Return $s^{(t)}$;

V. Experimental Design

In this section, we will describe our dataset and experiments' parameters setup. The experiment will use real movement dataset with BLE beacons and smartphone as bluetooth strength (RSSI) sensor.


```
{"timestamp":"2018-12-06 10:46:39.364"},
{"readings":[    {"mac-address":"XX:XX:XX:XX:XX:XX","rssi":0},
                 {"mac-address":"XX:XX:XX:XX:XX:XX","rssi":0},
                 {"mac-address":"XX:XX:XX:XX:XX:XX","rssi":0}]}
```


Fig. 5. Beacon Data Example

TABLE II
Walking behaviours for experiments

Behaviour	Count
A. Walking constantly	4
B. Walking and stop for each step for 2s	3
C. Walking and stop in each different semantic location for 2s	3
Total	10

A. Dataset

To collect the bluetooth observation data, we developed an Android application that records the bluetooth RSSI from predefined BLE beacons. It also captures an image for each RSSI sampling from the smartphone's camera. We performed the sampling as the phone's camera facing down. Thus, from the captured image, a ground truth label can be assigned to each observation. The sampling data sequences is stored as JSON file, where the structure of each sample can be seen in Figure 5. The predefined BLE beacons are marked by the MAC addresses. The data collection uses beabig BLE beacon with firmware Bluno version of 1.8 [12] with sampling rate of 200 ms each and Android OS 8.10 smartphone with 3 GB RAM, 1.6 GHz processor.

For every signal strength setup, we performed 10 samplings with length of 2 minutes. These samplings have different behaviours of walking, whose details can be found in Table II. We assume those behaviours as they reflect how people walk inside a museum. For the training set, we use 2 samples of each behaviour. Thus, we will conduct the experiment using 6 sequences as a training set and 4 sequences as a test set.

B. Parameter Setup

In this study, we deploy 3 beacons to emit bluetooth strength in certain places of our 7×9 m^2 laboratory. The placement of the beacons is already depicted in Figure 1. However, we arbitrarily place the beacons without any proximity or noise consideration just to study the possibility of performing this map matching in indoor environment.

The default setting and parameters of the experiment is shown in Table III, where the default setting is in bold. We divide the parameter study in two steps. The first one is to find the best parameter specific for our methods (aggregation, discretization, and KNN's). And the second one is to examine the non-method specific parameters, such as hyperparameters (learning time) and environmental (signal strength, configuration of deployed beacons) setting, thus we will see which parameter that affect the setting of the learning. For the second step, we will use the method specific parameters given the results in step one.

The explanation of some parameter setups is as follows. The discretization bin $C = 1$ means we only check whether

TABLE III
PARAMETER SETTING

No.	Parameter Setting	Values
1	Interval window range (in s)	0.25, 0.5, 1.0, **1.5**, 2.0
	Window style	Shifting, **Sliding**
	Sliding window range (in s)	0.25, 0.5, **1.0**
	Aggregation function	**Max**, Mean
	Discretization bin C	1, 2, 5, 10, **20**, 50, 100
	Random Restart (for HMM)	**100**
	Max. iteration (for HMM)	**10**
	K (for KNN)	1, 3, 5, 10, **15**, 20
2	Beacon signal strength	1, **2**, 3, 4
	Deployed beacon l	1, 2, **3**
	Learning time (in s)	30, 40, 50, 60, 90, **All**

a beacon signal exists or not, while $C = 100$ we leave the RSSI as it is. We set the interval window range no later that 2 seconds as longer range may cause skipping a semantic location track, considering the human's walking speed and the semantic region division. For HMM's parameter we fix the parameters as our training sequences set is quite few, thus any large random restart and maximum iterations should not affect the result too much. Then, the beacon signal strength accommodates the detection range of the distance from the beacon, where the lower number refer to narrower distance and vice versa. The deployed beacon parameter study will be presented with the different combination of deployed beacons. We study the learning time parameter because it is possible to have overfitting if we learn using the whole sequences.

To see how well our proposed methods perform, we will compare them with two methods. We will use the naïve KNN method and the supervised Multivariate MHMM method with MVNHMM library[1] against our modified KNN and bMHMM. The difference of the MVNHMM with the our bMHMM method is that we will perform full sequence viterbi on MVNHMM rather than the streaming style viterbi using bMHMM. The experiment's development environment is using Java JDK 1.8 and is ran in Windows 10 PC with 16 GB RAM and 3.6 GHz processor.

C. Evaluation Metric

To measure the performance of our proposed method, we will use accuracy metric $acc(\%)$ as described in Eq. 6. The values of the inferred state at time t $s^{(t)}$ are compared with ground truth $g^{(t)}$ within the length of aggregated sequence T. We will also present the visualization of all methods and the ground truth to see the predicted trajectory of the captured RSSI set.

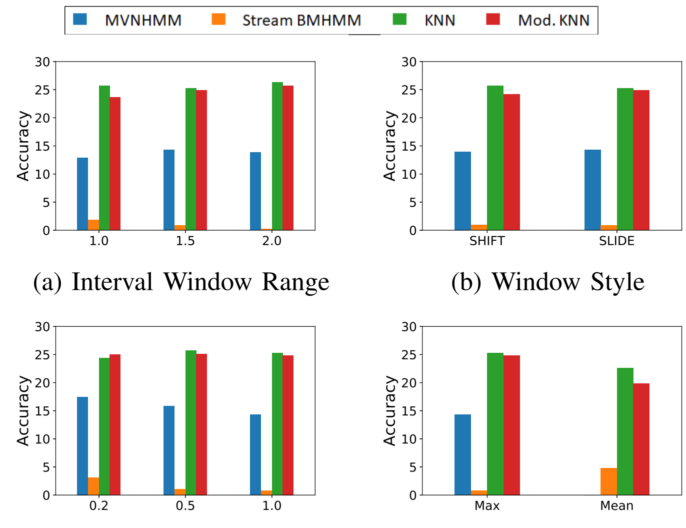
$$acc = \frac{1}{m}\sum_{i=1}^{m}\begin{cases}1, s_i = g_i\\0, s_i \neq g_i\end{cases} \tag{6}$$

VI. Results & Discussion

A. Step 1 - Method Specific Parameters

The result of our methods by varying the parameters setting of interval window (interval window range, window style, sliding window range, and aggregation function) is depicted in Figure 6. In majority, the result shows that the best method

[1]http://www.sergeykirshner.com/software/mvnhmm



(a) Interval Window Range (b) Window Style
(c) Sliding Window Range (d) Aggregation Function

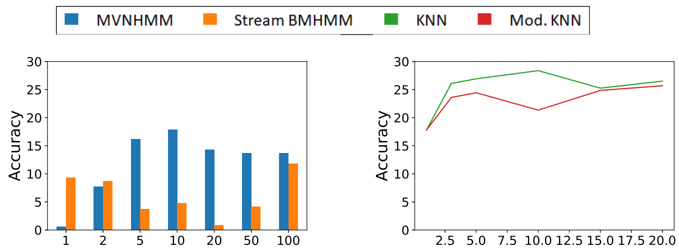
Fig. 6. Accuracy of different methods based on window parameters setting

is the naïve KNN, slightly followed by our proposed modified KNN, then the MVNHMM, and our proposed bMHMM, separated by extremely wide gaps. Overall, we see that the accuracy of all methods are poor (less than 50 percent). This may be caused by our noisy and quite small experiment environment. This poor result also might be affected by our arbitrary placement of beacons.

The interval window range variation did not seem affect the performance of each method very much. But for bMHMM, shorter window leads to a slightly better result, unlike the other methods. The change of the interval window style and the sliding window range also show similar triviality. However, we see that in the aggregation function variation result, the MVNHMM does not yield any correct result. That means the "mean" aggregation window is not suitable for the case of indoor positioning with beacons.

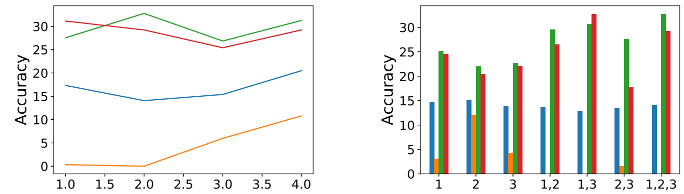
Figure 7 presents the results of altering the specific parameters for each method (number of discretization bin for HMM-based methods and number of neighbors (K) for KNN-based methods). We see that the discretization bin affects the performance of MVNHMM but not for the proposed bMHMM. We see that by checking the existence of non-zero captured RSSI or no discretization at all yields bMHMM the best accuracy, while dividing the RSSI by 10 bins gives the MVNHMM the best result. For KNN-based methods, we see that larger number of neighbors influences a slightly better result for both naïve KNN and modified KNN. But, modified KNN shows huge decrease in accuracy and KNN has the best result when K=10.

By those results, we see that the KNN-based methods have better performance than the HMM-based methods. The MVNHMM based inference, that was trained using ground truth, depended heavily on the ground truth. To move from a semantic location to another requires roughly more than 1 second, even in the walk constantly behaviour scenario, thus the transition probability from a semantic location to itself become higher than to the other connected semantic



(a) Num. of Discretization Bin (b) Number of neighbors

Fig. 7. Accuracy of different parameter settings specific of each method



(a) Signal Strength setup (b) Deployed beacons config.

Fig. 8. Performance of the methods using different environments

TABLE IV
THE PARAMETER SETTING FOR EXPERIMENT STEP 2

Parameter	bMHMM	MVNHMM	Mod. KNN	Naïve KNN
Window style	Slide	Slide	Slide	Slide
Sliding range	0.2 s	0.2 s	0.5 s	0.5 s
Agg. function	Mean	Max	Max	Max
C	1	10	-	-
K	-	-	20	15

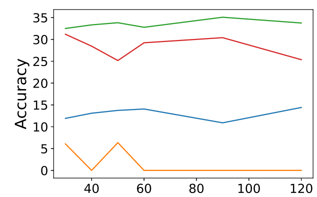


Fig. 9. Performance based on different learning times

locations. This affects the prediction of the next state that should be to different semantic location to be redirected to the current semantic location. The bMHMM approach, which streaming viterbi greatly depends on the only previous state, has less likelihood to refer to much past sequences. Thus, the prediction of bMHMM is not likely to be accurate. In future, it is likely to add a windowed based streaming viterbi to avoid this inaccuracy.

Finally, we set the parameter setting of the next stage's experiment according to the best result from this stage's experiment. However, we set the interval window range to 1 s because it will affect the length of the predicted sequence. The full details on the parameter setting is described in Table IV.

B. Step 2 - Learning Time and Environment Influence

This experiment provides the information about how does the environmental setting, e.g. signal strength setup and combination of deployed beacons, influence the performance of our methods. On the other hand, we will see whether longer training time affects the performance of our methods.

Figure 8 shows the performance of the methods by the different signal strength and the different deployed beacons configurations. We see that the modified KNN and naïve KNN yields the best accuracy with small differences among others given different settings, while bMHMM approach has the worst performance.

Figure 9 depicts the trend of the accuracy between the approaches according to the length of learning time. Except the bMHMM, it can be concluded that some longer learning time tends to hurt the performance of our predictions. According to the trend, it is better to have learn for the first 90 seconds of data.

C. Visualization

The visualization of the prediction results for 5 sample points is depicted in Figure 10. We see that the best performing method, naïve KNN, has non-sensible trajectory (jumping from a semantic trajectory to non-connected locations) that is far from ground truth and the previous points. We may conclude that modified KNN is currently our best option to predict trajectories from the BLE observations. Then, it also requires another metric that that penalizes the ”jumpy” trajectory, i.e. the next predicted semantic location moves to non-neighboring locations from the current prediction, and measures the proximity of the predicted point to the actual ground truth.

D. Comparison to Similar Method

Compared to [11], this result is still far beyond good. The experiments in [11] use wider area of exploration and more beacons, while we study the effectiveness of BLE in more sparse and smaller environment. Another difference of our study is that the design of the state space, which is similar to the path network. We only use the path network to infer the semantic location, rather than the two-stage network matching using the path network and the beacon network. On the other hand, we also show the visualization of the route estimation of our proposed method to prove that the inferred trajectory is sensible.

VII. Conclusion

In this paper, we propose two methods utilizing the floor plan to detect the semantic positions of objects using Bluetooth Low Energy (BLE) beacons. The indoor positioning system should be able to process the streaming data of BLE beacons. However, the well-known HMM method used in the indoor positioning such as MVNHMM does not support to obtain the semantic positions of objects in the streaming manner. Therefore, the first method based on Multivariate HMM method namely bMHMM is proposed to model uncertainties for transition and emitting probability of each semantic location and the discretized BLE RSSI streaming data. In addition, we

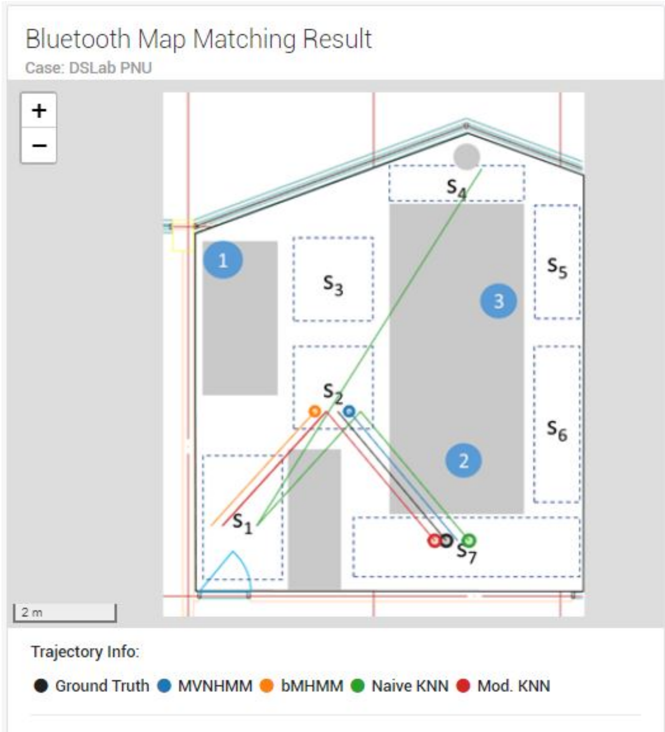


Fig. 10. Visualization of the methods' prediction

propose the second method that benefits the simple k-Nearest Neighbor (KNN) inference but applies the designated state space model called modified KNN. The two proposed methods are executed in the same two-step approach that consists of learning step and inference step.

The experiment results indicate that two proposed methods do not return quite high accuracy. The proposed modified KNN method performs significantly more accurate than the proposed bMHMM method in various parameter settings. However, the accuracy for the two proposed methods still is poor that might be affected by our arbitrary placement of BLE beacons and the small experiment environment. Compared to the bMHMM method, the modified KNN method is a better option to predict trajectories from the BLE observations. Furthermore, the two proposed methods works quite well since they do not have non-sensible trajectory compared to a baseline approach such as KNN.

As a future work, we require to expand the experiment in larger environment and put more detailed study in the placement of BLE beacons to obtain higher accuracy. As it can be seen, the two proposed methods show the possibility of obtaining higher accuracy to support practical use of indoor positioning.

REFERENCES

[1] A. Y. Lee, A. Kumar, P. Wilk, W. Chmielowiec, W. Jaworski, M. Skorupa, and P. Zborowski, "High accuracy indoor localization with low cost based on wireless lan, mobile sensors and floor layout," in *Consumer Electronics-Asia (ICCE-Asia), IEEE International Conference on*. IEEE, 2016, pp. 1–6.

[2] S. He and S.-H. G. Chan, "Wi-fi fingerprint-based indoor positioning: Recent advances and comparisons," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 466–490, 2016.

[3] A. I. Baba, M. Jaeger, H. Lu, T. B. Pedersen, W.-S. Ku, and X. Xie, "Learning-based cleansing for indoor rfid data," in *Proceedings of the 2016 International Conference on Management of Data*. ACM, 2016, pp. 925–936.

[4] P. Newson and J. Krumm, "Hidden markov map matching through noise and sparseness," in *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*. ACM, 2009, pp. 336–343.

[5] Q. Wang, R. Sun, X. Zhang, Y. Sun, and X. Lu, "Bluetooth positioning based on weighted K-nearest neighbors and adaptive bandwidth mean shift," *International Journal of Distributed Sensor Networks*, vol. 13, no. 5, p. 155014771770668, may 2017. [Online]. Available: http://journals.sagepub.com/doi/10.1177/1550147717706681

[6] P. Widya, Y. Yustiawan, and J. Kwon, "A onem2m-based query engine for internet of things (iot) data streams," *Sensors*, vol. 18, no. 10, p. 3253, 2018.

[7] L. Ma, Y. Fan, Y. Xu, and Y. Cui, "Pedestrian dead reckoning trajectory matching method for radio map crowdsourcing building in wifi indoor positioning system," in *Communications (ICC), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1–6.

[8] W. Zhang, X. Hua, K. Yu, W. Qiu, X. Chang, B. Wu, and X. Chen, "Radius based domain clustering for wifi indoor positioning," *Sensor Review*, vol. 37, no. 1, pp. 54–60, 2017.

[9] J. Heo and Y. Kwon, "Improved indoor positioning system using ble beacons and a compensated gyroscope sensor," in *International conference on intelligent robotics and applications*. Springer, 2017, pp. 69–76.

[10] R. Momose, T. Nitta, M. Yanagisawa, and N. Togawa, "An accurate indoor positioning algorithm using particle filter based on the proximity of bluetooth beacons," in *Consumer Electronics (GCCE), 2017 IEEE 6th Global Conference on*. IEEE, 2017, pp. 1–5.

[11] D. Yamamoto, R. Tanaka, S. Kajioka, H. Matsuo, and N. Takahashi, "Global map matching using ble beacons for indoor route and stay estimation," in *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2018, pp. 309–318.

[12] DFRobot, "Bluno backup SKU:DFR0267 - DFRobot Electronic Product Wiki and Tutorial: Arduino and Robot Wiki-DFRobot.com," 2017. [Online]. Available: https://www.dfrobot.com/wiki/index.php/Bluno_backup_SKU:DFR0267