# **Topological Nodes Generation Using Anomaly Scores of** WiFi Signal Strengths

Nosan Kwak<sup>\*</sup>, Sukjune Yoon, Soonyong Park, Jimin Kim, Sohee Lee, Kyungshik Roh Samsung Electronics Co., Ltd. Maetan-dong, Youngtong-gu, Suwon, Gyeunggi-do, Korea.

\* Corresponding author. Email: nosan.kwak@samsung.com Manuscript submitted July 25, 2015; accepted June 13, 2016. doi: 10.17706/ijcce.2017.6.1.29-39

**Abstract:** The essential information for location based service (LBS) is, of course, the location of a mobile robot providing the services. In outdoor environments, GPS is definitely the solution for LBS, however it is difficult to use it in indoor environments due to its weak signal strength. The two main radio-based approaches to the indoor localization are trilateration and fingerprinting, which require either many signal sources or heavy calibration. In this paper, we focus on the fact that the purpose of the localization is to provide services not at the whole locations but at certain locations. For this purpose, we propose a novel topological node generation method using anomaly scores of WiFi signal strengths, enabling to localize the robot under a single WiFi access point. Learning of sequential RSSIs with a hierarchical temporal memory model is able to detect a distinct node. We show the experimental results on node generation and recognition of a node using only one WiFi access point.

Key words: Anomaly detection, hierarchical temporal memory, topological node, WiFi.

# 1. Introduction

Indoor localization is one of the hottest issues in the field of Simultaneous Localization and Mapping (SLAM). Companies like Google and Apple are providing location based services (LBSs) in several shopping malls. The main problem of indoor localization is that it cannot fully resort to the Global Positioning System (GPS) due to its weak signal inside a building; signal strength of GPS is less than -1000dBm [1]. There are several radio-based methods using RFID, Bluetooth, and WiFi. A well-documented reference for comparing these radio-based methods is presented in [2]. The main concern in our research is practicality as to whether a localization method can be used in a real situation like a robotic cleaner or not. In this sense, we think the WiFi is the best option because a robot can readily receive signal strengths from any WiFi access points (APs), and most of buildings have several WiFi APs, which requires no need to install them. In addition, a WiFi AP covers large areas about 50m<sup>2</sup>. If there are more than one AP, the coverage areas enlarge and the accuracy of localization becomes much accurate. However, WiFi signals are not accurate in terms of localization, and it is necessary to use more than one AP to increase the location accuracy. Received signal strength indicator (RSSI) is an indication of the power level being received, and it is commonly used to estimate locations. However, RSSI is a relative term, thus it may differ from a vendor to a vendor; some vendors give dBm (decibel per milliwatt), others give quality of a signal in percentages. Also, RSSI varies with the chipset [3]. Therefore, it is better to use quality of received signals, which is adopted in this study.

# 1.1. Related Work

There have been several studies on indoor localization using WiFi, which are well surveyed in [4]. Two main radio-based approaches to localization are trilateration and fingerprinting methods. Trilateration uses measured distances from more than three known signal sources to determine the location of a robot as shown in Fig. 1. This method is cost-effective but the disadvantages are that the signal strengths are attenuated due to walls or doors and that the WiFi signals are interfered by devices using 2.4 GHz frequency. In addition, the converting RSSI into a distance requires a steep linear characterization curve in order to be properly implemented [5]. WiFi fingerprinting method [6], [7] creates a radio map of given areas based on the RSSI data from several APs and generates a probability distribution of RSSI values for a given location. Live RSSI values are then compared to the fingerprint map to find the closest match and to generate a predicted location. The method provides relatively good location accuracy but requires rigorous site survey to build the radio map in off-line.



Fig. 1. Standard dilatation method using three WiFi APs.

Not only localization but also mapping of environments are studied in the field of SLAM. Initial SLAM study using WiFi signals appears on [8], and a recent work with a RGB-D camera [9] is published.

Recently Apple bought a start-up company that developed *WiFiSLAM* by using WiFi and inertial measurement units (IMUs). The localization accuracy of the *WiFiSLAM* is known to about 1m.

The Google Indoor Map [10] provides location information inside several buildings. It mostly uses WiFi APs in order to localize mobile devices. It is able to distinguish which floor a device is in.

### 1.2. The Proposed Method

The ultimate goal of this work is to provide a robot cleaner with rough but meaningful location information using only one WiFi AP. It is known that WiFi signals are attenuated due to walls or doors and that the WiFi signals are interfered by devices, which are definitely undesirable characteristics. However we observed that the characteristics can be used to generate distinct places where signal strength suddenly decreases or increases. The odometry of the robot can be used to locally estimate the location, and place or node information can be used to globally correct the location in case that a node is recognized. Thus, the basic idea of this work is to build a topological map consisting of several distinct nodes only using a single WiFi AP, which can be used to correct the location estimate by the odometry. Then, the topological WiFi map can be used to instantly and globally localize the robot with a node recognition method. In this work, we focus on the generation of topological nodes. For the robustness of the node generation, we use Hierarchical Temporal Memory (HTM) and Cortical Learning Algorithm (CLA) [11], which is recently proposed by modeling the mechanism of operations of a human brain. HTM with CLA is good at detecting anomaly in a

series of sequential data. It is like convolutional neural networks for deep learning, but it can handle temporal information of data. In this paper, HTM with CLA is used to learn the pattern of sequential WiFi RSSIs and detect anomalies of them. The recognition of a node is also an important process in order for the proposed method to operate in real situations. In this work, we simply adopt a standard approach to recognizing a node with given RSSI data. We believe that the anomaly scores of sequential RSSI data provide much robust node estimate than the raw RSSI data.

The paper is organized as follows: anomaly score of WiFi signals and the method to learn the signals using HTM are provided in Section 2.

In Section 3, generation of topological nodes is explained, In Section 4, recognition of topological nodes is explained. Experimental results on node generation and recognition are discussed in Section 5. Finally, conclusion is provided in Section 6.

# 2. Background

In this Section, the supporting materials and main ideas are briefly introduced for the topological nodes of WiFi RSSIs. RSSI varies with the distance between an AP and a receiving device. The relationship between the distance and RSSI is not linear, but the RSSI decreases as the device moves away from the AP. In general, RSSI may vary at the same location without any movement, so it is required to take an average to obtain a stable RSSI. Therefore, localization using RSSI is not accurate, especially when the number of APs in a shared region is low. Based on the above considerations, we think that the changes of RSSIs can give more meaningful information than the RSSI itself. The proposed method in this paper computes anomaly scores of sequential RSSIs and learns their patterns using HTM and CLA.

# 2.1. HTM with CLA

HTM is a new and powerful mechanism to learn temporal data. It mimics the way that the human brain works. In a nutshell, HTM

- accepts stream of data.
- learns spatial and temporal patterns.
- creates a model of the data and predicts the data.

The learning mechanism especially for HTM is Cortical Learning Algorithm (CLA), which is briefly summarized as follows:

- CLA aims to simulate the way that the neo-cortex works.
- CLA is an algorithm that describes the operation of a single layer of cortical neurons.
- This layer can learn spatial and temporal patterns in data on its own.



#### Feed-forward input

Fig. 2. HTM region. Each column in an HTM region has one or more cells (here four), each of which is in either inactive, active, or predictive state. Some cells in an HTM region will be active due to feed-forward input (light gray), and other cells that receive lateral input from active cells will be in a predictive state.

A part of the architecture of HTM is described in Fig. 2. Here, we briefly introduce how it is able to predict and recognize data. Firstly, each column receives the same feed-forward input from the previous level or raw data. By selecting different active cells in each column, HTM can represent the exact same input differently in different contexts. Here we introduce a conceptual example from [11]. There are two sentences: "I ate a pear" and "I have eight pears". The "ate" and "eight" are homonyms but have different contexts. In Fig. 2, there are four cells in a column meaning that a column has four different contexts. Secondly, columns only activate predicted cells. If an input pattern is expected then the HTM confirms that expectation by activating only the cells in the predictive state.

If the input pattern is unexpected then the HTM activates all cells in the column as if to say "all possible contexts are valid". Thirdly, in each column one or more cells become active, which makes the nearby cells predictive states through learned connections. Lastly, the output of the HTM region is the activity of all cells in active or predictive state.

HTM can predict not only the next one step but also the several time steps. For example, after learning melody A, B, C, D, E, and then hearing A and B, HTM predicts the C, D, E.

A comparison study between HTM and Deep Belief Network (DBN) is presented here [12], where the classification performance of HTM is better than that of DBN. The main difference between HTM and DBN is that HTM can deal context of input patterns but DBN cannot.

In this paper, we use HTM with CLA to learn WiFi RSSIs and then generate topological WiFi nodes by detecting anomalies of sequential RSSIs. It is difficult to generate distinct nodes using only RSSIs. Instead of RSSIs, we use anomaly scores of RSSIs, which are computed by the outputs of HTM.

#### 2.2. Why Anomaly Score of RSSIs?

Anomaly means something that deviates from what is standard, normal, or expected. Fig. 3 shows a stream of RSSIs and their anomaly scores computed by a learned HTM.



Fig. 3. Example of RSSIs and their anomaly scores.

Since WiFi RSSIs are not robust on locations, which means RSSIs vary at the same location. An anomaly score measures the degree to which each data is predictable. A zero (0) anomaly score represents a completely predicted data whereas a one (1) represents a completely anomalous data.

Since HTM can predict values, the anomaly score (*AS*) can be computed using the predicted values and the real input values as follows:

$$AS = \frac{|A_t - (P_{t-1} \cap A_t)|}{|A_t|}$$
(1)

where  $P_{t-1}$  is the predicted columns at time t and  $A_t$  is the active columns at time t. Thus, an anomaly score of 1 means that no predicted cells in HTM became active. The AS might be high where RSSIs abruptly change because of environmental structure.

Anomaly score often becomes high because new data to HTM is not predictable. In this case, one can use anomaly likelihood, which is a probability of anomaly and is obtained as follows:

- A. For each new data point compute anomaly score.
- B. Estimate the probability distribution of historical anomaly scores.
- C. Compute likelihood that the recent anomaly scores come from same distribution as historical anomaly scores.

#### 3. Generation of Topological Nodes



Fig. 4. Flowchart of generating topological nodes.

The flowchart of generating topological nodes is presented in Fig. 4. For brevity, the explanation will be presented for a mobile robot such as *Turtlebot2*. The node generation scheme can be applied to any mobile devices or robots without loss of generality.

A robot can be moved by a user or for itself using exploration techniques such as frontier-based exploration [13]. While a robot is roaming around the environment, it records RSSIs at each sampling time. In addition, it records the position of itself to display the node map and to cluster anomalous positions if the number of the generated nodes is too many. Note that the positions are not input to the HTM. The position of the robot can be estimated by using *GMapping* [14] that showed good localization performance on mobile robots. The HTM for the topological node map takes only the RSSI as an input to compute the anomaly score. The position information will not be used to recognize a node since the purpose of the recognition is to globally localize a robot using only WiFi signals.

After covering the whole areas of the environment by manually controlling the robot, the robot builds the topological WiFi node map in off-line with the flow as shown in Fig. 4. At first, the HTM for the topological node map is initialized. Then RSSIs are input to the HTM, and the anomaly score is computed using (1). If the anomaly score is greater than the threshold,  $T_{AS}$  then the robot position is kept as a tentative node. This

process is iterated until the last RSSI is dealt. Then, there might be nodes that are nearby each other. If then, the mean-shift algorithm [15] is executed to cluster the nearby nodes. The mean-shift algorithm does not require prior knowledge of the number of clusters, and does not constrain the shape of the clusters, which is suitable in this study to merge the nearby nodes.

In order for a node to be recognized through sequential RSSI data, a node should have distinct features. In this work, we use a few RSSIs received around the node because we want to conduct global localization once the node map is built. In order to construct the feature vector for a node, we collect the RSSIs nearby the node by using k-nearest neighbor search with the position information. Then a *KD-tree* is built with the feature vectors of all the nodes. The data for the *KD-tree* has  $N_F$  length, which is the length of the feature vector of a node.

**Recognition of Topological Nodes** 

4.



Fig. 5. Flowchart of recognizing WiFi nodes.

Fig. 5 represents a flowchart to recognize a topological node, where querying a feature vector within a KD-tree is used to recognize a node in the generated map. One scenario for the node recognition is global localization where a robot receives sequential RSSI data and matches the data with ones in the *KD*-tree. The result will give the nearest node to the given RSSI data and the matching distance. If the matching distance is smaller than the matching threshold  $T_M$ , then the query is a success. Otherwise, it is a failure. This recognition method is simple but has a limitation that requires the exact same number of features as the ones in the *KD*-tree. If there are more numbers of data than that of feature vector in the *KD*-tree, an averaging RSSI data might be used to construct the same size as the feature vector.

The better method to recognize a node is to use another HTM to train the relationship between the sequential RSSI data and its node. HTM can be used to learn spatial information for this kind of classification problem. However, lots of training data are required for HTM to learn the relationship, which is not suitable if the number of nodes is small. In a house or an office, the number of topological nodes does not exceed more than a hundred. Thus, in this work we simply adopt the simple method that matches the given stream of RSSIs with the ones in the *KD-tree*. Note that positions where RSSI data are captured are not used to recognize a node. In order for a robot to globally localize, the robot position will be something to be

corrected using other sensors, thus it is excluded from the input to the HTM. To distinguish nodes having similar RSSI features, information providing absolute position such as magnetic sensors or visual features can be used.

# 5. Experimental Results

In order to initially construct the topological WiFi nodes we made Turtlebot2 as shown Fig. 6 navigate an environment. The positions of the robot were estimated by ROS *GMapping* [14]. The ASUS laptop on the robot has Celeron 1007U dual-core @1.5GHz, 4GB of RAM, attached with 802.1n WiFi module. Only one AP (ipTime N8004) was used for this experiment.



Fig. 6. Turtlebot2.

The test environment and the traveled path are shown in Fig. 7.

# 5.1. Node Generation



Fig. 7. Floor plan of the testing environment and the traveled path.

The generated node map is presented in Fig. 8.

We set the size of feature vector  $N_F$  to 10 for building this map. In the figure, the blue dots are robot positions where the robot recorded the RSSI data, and the anomalous positions which of anomaly scores are over 0.9 are represented as red circles. The dark green circles are the generated nodes after being merged using a mean-shift algorithm (quantile=0.1). In this test, 10 nodes were generated, and the information of each node is shown in Table 1. The information consists of the RSSI feature vector, the average *AS*, and the position.

Note that the RSSI feature vectors from a node to a node did not change much, so that the node recognition using only the raw RSSI data would become difficult.



Fig. 8. Resulting node map with anomalous positions in red.

Node No.	RSSI Feature Vector	AverageAS	Position ( <i>x</i> , <i>y</i> ) <i>m</i>
1	(89, 89, 89, 89, 89, 95, 89, 89, 89, 89)	1.0	(8.2, -0.2)
2	(87, 85, 80, 85, 87, 87, 83, 82, 87, 85)	1.0	(11.8, 0.0)
3	(82, 85, 82, 84, 85, 85, 83, 87, 87, 82)	0.97	(-3.9, -12.8)
4	(89, 95, 91, 93, 93, 93, 89, 91, 91, 89)	0.98	(2.1, -0.3)
5	(84, 83, 82, 82, 82, 84, 82, 78, 85, 89)	0.97	(-11.8, 2.5)
6	(83, 82, 82, 83, 78, 82, 85, 83, 83, 85)	0.97	(-12.5, -6.2)
7	(72, 76, 74, 74, 70, 78, 76, 76, 82, 74)	0.98	(25.7, -1.0)
8	(82, 82, 78, 84, 84, 78, 74, 80, 82, 82)	0.97	(-9.0, -11.6)
9	(87, 89, 89, 91, 83, 85, 89, 91, 89, 93)	1.0	(-9.0, -11.6)
10	(78, 72, 74, 72, 70, 76, 72, 74, 74, 76)	0.98	(28.2, 2.2)

Table 1. Feature Vectors of Generated Nodes

# 5.2. Node Recognition

We obtained RSSI data in online from a few places in the test environment and queried them in the *KD-tree* built in the generation process. Fig. 9-Fig. 11 show the node recognition results, and the testing RSSI data and matching distances are presented in Table 2.

Node No.	RSSI Feature Vector	Matching Distance
3	(82, 84, 82, 82, 81, 84, 82, 84, 82, 78)	8.54
3	(84, 84, 83, 81, 84, 84, 81, 81, 81, 79)	10.01
10	(65, 65, 65, 64, 74, 62, 68, 67, 63, 67)	29.01

Table 2. Testing RSSI Vector and Matching Distance

The matching distance is Euclidean distance between RSSIs of the matched node and testing RSSIs.

The first testing position (magenta circle) as shown in Fig. 9 was recognized as the node 3, shown in the light green circle. The matching distance was 8.54 even though the two positions were apart from each other about 5m. On the other hand, the third testing position located just nearby the node 10 had much larger matching distance, 29.41, compared to its actual distance to the node 10. These results show that the matching distance of the Euclidean form is not a good performance measure because RSSIs vary from time to time at the same location. When the variation is large, methods to use raw RSSIs cannot match a node.



Fig. 9. Matched node is the node 3 shown in light green circle.



Fig. 10. Matched node is the node 3 shown in light green circle.



Fig. 11. Matched node is the node 10 shown in light green circle.

Especially using anomaly score to recognize the node would produce poor performance in case of open environments where there does not exist structures to alter radio signals. To distinguish nodes having similar RSSI features, magnetic sensors or visual features can be added to construct the feature vector of a node.

# 6. Conclusion

We proposed the method to automatically generate topological WiFi nodes using anomaly scores of sequential RSSI data, specifically from a single WiFi AP. In addition, we showed the usage of the generated nodes by recognizing given RSSI data. The major difference of the proposed approach from previous ones is that it uses anomaly scores rather than raw RSSIs. The anomaly score computed by HTM with CLA means more than just the variation of RSSIs. It can automatically separate regions of an environment when the score is over a threshold. We showed that the robot having a WiFi module could be localized only using

sequential RSSI data. However, the recognition results are not robust enough to use the method for a global localization or re-localization problem. The proposed work is a preliminary research to implement global localization for a product such as a robotic cleaner. As future work, we will conduct rigorous experiments to generate WiFi nodes and develop a method that robustly recognizes the generated nodes.

# References

- [1] Nirjon, S., Liu, J., Dejean, G., Priyantha, B., Jin, Y., & Hart, T. (2014). COIN-GPS: Indoor localization from direct GPS receiving. *Proceedings of Int'l Conf. on Mobile Systems, Applications and Services*.
- [2] What's the best micro-location technology? From http://localz.com/blog/whats-best-micro-location-technology
- [3] Lui, G., Gallagher, T., Li, B., Dempster, A. G., & Rizos, C. (2011). Differences in RSSI readings made by different WiFi chipsets: A limitation of WLAN localization. *Proceedings of International Conference on Localization and GNSS (ICL-GNSS)*.
- [4] Farid, Z., Nordin, R., & Ismail, M. (2013). Recent advances in Wireless indoor localization techniques and system. *Journal of Computer Networks and Communications*.
- [5] Cook, B., Buckberry, G., Scowcroft, I., Mitchell, J., & Allen, T. (2005). Indoor location using trilateration characteristics. *Proceedings of London Communications Symposium*.
- [6] Shin, H., & Cha, H. (2010). Wi-Fi fingerprint-based topological map building for indoor user tracking. Proceedings of IEEE Int'l Conference on Embedded and Real-Time Computing Systems and Applications (pp. 105-113).
- [7] Husen, M. N., & Lee, S. (2014). Indoor human localization with orientation using WiFi fingerprinting. Proceedings of the 8th International Conference on Ubiquitous Information Management and Communication.
- [8] Ferris, B., Fox, D., & Lawrence, N. (2007). WiFi-SLAM using Gaussian process latent variable models. *Proceedings of International Joint International Joint Conference on Artificial Intelligence.*
- [9] Ito, S., Endres, F., Kuderer, M., Tipaldi, G. D., Stachniss, C., & Burgard, W. (2014). W-RGB-D: floor-plan-based indoor global localization using a depth camera and WiFi. *Proceedings of IEEE Int. Conf. on Robotics and Automation*.
- [10] Google Indoor Maps. From https://www.google.com/maps/about/partners/indoormaps
- [11] HTM white paper. From http://numenta.org/resources/HTM\_CorticalLearningAlgorithms.pdf
- [12] Valentin, K., Bajla, I., & Stolc, S. (2013). A performance comparison of HTM and DBN models applied to visual object classification. *Proceedings of the 9th International Conference on MEASUREMENT.*
- [13] Yamauchi, B. (1997). A frontier-based approach for autonomous exploration. *IEEE International Symposium on Computational Intelligence in Robotics and Automation*.
- [14] Grisetti, G., Stachniss, C., & Burgard, W. (2007). Improved techniques for grid mapping with Rao-Blackwellized particle filters. *IEEE Transactions on Robotics*, *23*, 34-46.
- [15] Cheng, Y. (1995). Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *17(8)*.



**Nosan Kwak** received the Ph.D. degree in electrical engineering and computer science from Seoul National University, Korea, 2008. He is currently a senior engineer in the Global Technology Center at Samsung Electronics, Korea. His research interests include neuroscience and robot navigation.



**Sukjune Yoon** received the Ph.D. degree in mechanical engineering from Korea Advanced Institute of Science and Technology, 2005. He is currently a principal engineer in the Global Technology Center at Samsung Electronics, Korea. His research interests include computer vision and robot navigation.



**Soonyong Park** received the Ph.D. degree in electrical and electronic engineering from Yonsei University, Seoul, 2010. He is currently a senior engineer in the Global Technology Center at Samsung Electronics, Korea. His research interests include computer vision, machine learning, and mobile robot navigation.



**Jimin Kim** received the Ph.D. degree in electrical engineering and computer science from Seoul National University, Korea, 2012. She is currently a senior engineer in the Global Technology Center at Samsung Electronics, Korea. Her research interests include robot navigation and motion planning.



**Sohee Lee** received her Ph.D. degree in mechanical and aerospace engineering from Seoul National University, Korea, 2011. She is currently a senior engineer in the Global Technology Center at Samsung Electronics, Korea. Her research interests include robot manipulation and motion planning.



**Kyungshik Roh** received the Ph.D. degree in robot vision from Korea Advanced Institute of Science and Technology, 1998. He is currently a research director in the area of intelligent robotics at Samsung Electronics. His research interests include computer vision and robot navigation.