Machine Learning Based Indoor Localization Using a Representative k-Nearest-Neighbor Classifier on a Low-Cost IoT-Hardware

Matthias Dziubany*, Rüdiger Machhamer*, Hendrik Laux†,
Anke Schmeink†, Klaus-Uwe Gollmer* Guido Burger‡ and Guido Dartmann*
*Trier University of Applied Sciences
†Research Group ISEK, RWTH Aachen University
‡Expert Group Internet of Things - Digital Gipfel Germany

***Abstract*—In order to make Internet of Things (IoT) applications easily available and cheap, simple sensors and devices have to be offered. To make this possible, our vision is to use simple hardware for measurements and to put more effort in the signal processing and data analysis to the cloud. In this paper, we present a machine learning algorithm and a simple technical implementation on a hardware platform for the localization of a low accuracy microphone via room impulse response. We give a proof-of-concept via a field test by localization of multiple positions of the IoT device. The field test shows that the recorded signals from the same source are unique at any position in a room due to unique reflections. In contrast to other methods, there is no need for high accuracy microphone arrays, however, at the expanse of multiple measurements and training samples. Our representative k-nearest-neighbor algorithm (RKNN) classifies a recording using a k-nearest-neighbor method (KNN) after choosing representatives for the KNN classifier, which reduces computing time and memory of the KNN classifier.**

I. Introduction

In multiple Internet of Things (IoT) applications, localization is an important service. However, in especially indoor applications localization becomes difficult due to the missing GPS-link. Current indoor localization concepts either require microphone arrays or multiple radio beacons and expensive hardware for the synchronization. Our idea is to use simple hardware (sensors, IoT devices) for the measurements and machine learning algorithms on cloud computers for the localization of objects at the expense of a continuous generation of training samples. The idea of applying machine learning to indoor localization is introduced in [1] and is based on the experiment explained in [2]. In the experiment, a group of people localized a sound source in the dark. In case a plastic strip was put in their outer ear they lost the ability to localize. In order to learn the localization with a plastic strip in the outer ear, the subject group kept the plastic strip in the ear for a few weeks. When the experiment was repeated with the plastic strip, the accuracy of the localization increased, which showed that the localization of a sound source can be learned. Figure 1 illustrates the idea of unique reflection pattern for any position and Figure 2 underlines this idea by showing two almost identical room impulse responses recorded at the same position. Further explanation can be found in [1].

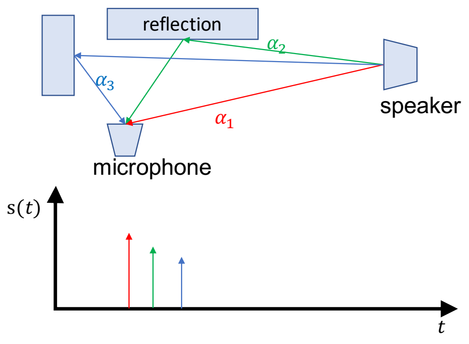

Fig. 1. Idea of unique reflection pattern. The first impuls α_1 comes directly from the source. Further impulses like α_2 and α_3 are delayed and weakened because of reflections.

Related Work: There are only a few approaches in indoor localization, which can be implemented on simple hardware [3]. These approaches differ in the physical medium they use for localization. Common media, which can be sensed at low cost are sound, light and radio-signal-strength. In the following, we briefly discuss the three techniques that can be implemented on simple hardware with low cost. Most of the recent literature uses WiFi access points for localization [7]. As WiFi access points already exist in many buildings and many common devices can measure radio-signal-strength of WiFi access points, this technique can be easily implemented. However, if there are not enough WiFi access points, new ones have to be installed thereby increasing the total cost. In [4], a localization using visible light communication is proposed. This technique can be implemented on simple hardware, nevertheless the costs to set up the LED-spots is expensive. For techniques based on sound, expensive microphone arrays are necessary for most examples in literature [5], [6]. In contrast to the techniques discussed above, our localization based on sound does not require many installations.

Contribution: Compared to other localization methods like localization via signal-strength-based positioning with WiFi access points, bluetooth beacons [7] or localization by a high accuracy microphone array [5], our method based on room impulse responses has lower hardware requirements.

It may be applied to storehouse indoor-navigation purposes. In the following, we present the signal processing chain for the preprocessing and an adapted k-nearest-neighbor classifier (RKNN) and its localization results in a field test on a new customized simple IoT hardware device.

Structure of the Paper: After introducing the signal model in Section II, we present the applied k-means clustering in Section II-A and the k-nearest-neighbor classifier in Section II-B. Then, we introduce a solution based on the representative k-nearest-neighbor classification in Section II-C and discuss its advantages over the original KNN classifier. Section III presents the technical implementation with a detailed description of a field test. We conclude our paper with a discussion and further work in Section IV.

Notations: Bold letters $\mathbf{x}$ denote vectors and bold capital letters $\mathbf{M}$ denote matrices. The transpose of a matrix $\mathbf{M}$ or vector $\mathbf{x}$ is denoted by $\mathbf{X}^T$ or $\mathbf{x}^T$, respectively. Sets are denoted by a calligraphic font, e.g., $\mathcal{S}$ and the cardinality of a set is given by $|\mathcal{S}|$.

II. Signal Model and Learning Algorithms

Based on the reflection scheme given in Figure 1, the room impulse response (RIR) $s(t)$ is given by N scaled real valued amplitude pulses α_i delayed in time. The length or number of pulses N depends on the size of the room where the measurement is performed. Furthermore, we assume that $x(t)$ is the sampled signal of $s(t)$

$$x(t) = \sum_{m=0}^{\infty} s(mT)\delta(t - mT),$$

where $f_a = 1/T$ is the sampling rate. All sampled measurements $s(mT)$ are stored in a vector $\mathbf{x} = [s(T), s(2T), ..., s(dT)]^T$. Here, we assume that d measurements are gained from the microphone of the IoT device. For two measurements $\mathbf{x}$ and $\mathbf{y}$ with $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, we apply the cosine distance

$$d(\mathbf{x}, \mathbf{y}) = 1 - \frac{\mathbf{x}^T \mathbf{y}}{\sqrt{\mathbf{x}^T \mathbf{x}} \cdot \sqrt{\mathbf{y}^T \mathbf{y}}} \tag{1}$$

The distance between two room impulse responses is small, if the peaks, which can be seen in Figure 2, are similar at many time points.
Instead of applying the KNN classifier directly on all training signals, we apply k-means clustering with the same distance measure to the training data with the same label and get k representatives for every label. This reduces both execution time and memory of the KNN classifier and thus it's complexity. It is important to note that the k-means clustering has to be performed on a better device, e.g., a cloud computer, since the memory of the low-power device is limited and the whole training data has to be considered in the k-means clustering. Since the representatives are optimized regarding the distance measure of the KNN classifier they represent the training data well for prediction. Further, using the same amount of representatives for each label ensures equal chances for each label to get the majority in the KNN classification.

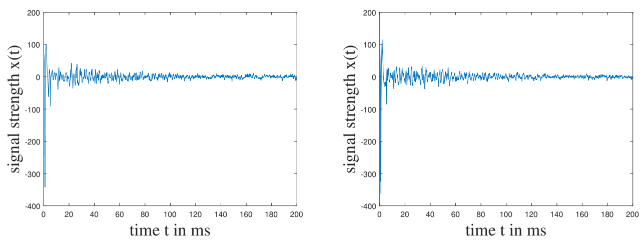

Fig. 2. Two room impulse responses recorded at the same position.

In order to introduce our machine learning algorithm in detail, we start with a short explanation of the well known k-means clustering and the KNN classifier.

A. k-Means Clustering

Suppose we are given a set $\mathcal{M} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ of n data points $\mathbf{x}_i \in \mathbb{R}^d$ with $i \in \{1, \ldots, n\}$. The goal of the k-means algorithm is to partition the data points in k groups $\mathcal{C}_1, \ldots, \mathcal{C}_k$ called clusters and to assign a representative $\mathbf{r}_j \in \mathbb{R}^d$ with $j = 1, \ldots, k$ to each cluster $\mathcal{C}_j$, such that the sum of distances dist $= d(\mathbf{x}_i, \mathbf{r}_j)$ of each data point to his representative is minimal:

$$\min_{\mathbf{r}_j \in \mathbb{R}^d} \sum_{\mathbf{x}_i \in \mathcal{C}_j} d(\mathbf{x}_i, \mathbf{r}_j).$$

This can be done by an iterative procedure with two successive optimization steps per iteration. In the first step, we optimize the representatives $\mathbf{r}_j$ with $j = 1, \ldots, k$ and in the second step the assignment of points to the representatives calculated before. Further explanations can be found in [8].
If a lot of training data is collected, choosing the representatives of big clusters instead of the original training data for training a machine learning algorithm does not only reduce the amount of training data, it also reduces the impact of outliers. Note that the number of clusters k has to be determined before running the k-means algorithm. In our experiment, we tested different numbers of clusters and chose a small number with high accuracy on the test set. The pseudo code of k-means clustering is presented in Algorithm 1.

B. k-Nearest-Neighbor Classifier

Suppose we are given a set $\mathcal{M}^{\mathcal{L}} = \{(\mathbf{x}_1, l_1), \ldots, (\mathbf{x}_n, l_n)\}$ of n labeled training data points $\mathbf{x}_i \in \mathbb{R}^d$ and label $l_i \in \{1, \ldots, L\} = \mathcal{L}$ with $i \in \{1, \ldots, n\}$. The goal of the KNN classifier is to predict the label of a new test data point $\mathbf{x}_0 \in \mathbb{R}^d$ given a labeled training set $\mathcal{M}^{\mathcal{L}} = \{(\mathbf{x}_1, l_1), \ldots, (\mathbf{x}_n, l_n)\}$. In order to predict the label of a test data point $\mathbf{x}_0$, the KNN classifier determines the k nearest training data points $\mathcal{N}_k(\mathbf{x}_0) = [(\mathbf{x}_{j_1}, l_{j_1}), \ldots, (\mathbf{x}_{j_k}, l_{j_k})]$ to $\mathbf{x}_0$ with $(\mathbf{x}_{j_i}, l_{j_i}) \in \mathcal{M}^{\mathcal{L}}$. Let label$_k(i)$ denote the label of the ith nearest training data point x_{j_i} with $i \in \{1, \ldots, k\} = \mathcal{K}$ to x_0, then

$$l = \text{argmax}_{l \in \mathcal{L}} \{ \ |\{i : \text{label}_k(i) = l, \ i \in \mathcal{K}\}| \ \}$$

denotes the label with highest occurrence in set $\mathcal{N}_k(\mathbf{x}_0)$. Ties are broken at random. More details are given in [9].

Algorithm 1 k-Means: kMeans($\mathcal{M}$)
Input: Set of n data points $\mathcal{M} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$
Output: $k < n$ representatives $(\mathbf{r}_1, \dots, \mathbf{r}_k)$ for the data points

```
1:  for $j = 1, \dots, k$ do
2:    $\mathbf{r}_j^{new} = \mathbf{x}_j$ % Random initialization
3:  end for
4:  $\mathbf{r}^{old} = [\infty, \dots, \infty]^T$
5:  $\mathbf{r}^{new} = [\mathbf{r}_1^{new}, \dots, \mathbf{r}_k^{new}]^T$
6:  while $\mathbf{r}^{new} \neq \mathbf{r}^{old}$ do
7:    for $i = 1, \dots, n$ do
8:      $best = \infty$
9:      for $j = 1, \dots, k$ do
10:       $dist = d(\mathbf{x}_i, \mathbf{r}_j^{new})$
11:       if $dist < best$ then
12:         $best = dist$
13:         representative$(\mathbf{x}_i) = j$
14:       end if
15:     end for
16:     add $\mathbf{x}_i$ to $\mathcal{C}_{\text{representative}(\mathbf{x}_i)}$
17:   end for
18:   $\mathbf{r}^{old} = \mathbf{r}^{new}$
19:   for $j = 1, \dots, k$ do
20:     $\mathbf{r}_j^{new} = \text{argmin}_{\mathbf{r}_j \in \mathbb{R}^d} \sum_{\mathbf{x}_i \in \mathcal{C}_j} d(\mathbf{x}_i, \mathbf{r}_j)$
21:   end for
22: end while
23: return$\big((\mathbf{r}_1^{new}, \dots, \mathbf{r}_k^{new})\big)$
```

Algorithm 2 k-Nearest-Neighbor Classifier: KNN($\mathcal{M}^{\mathcal{L}}, \mathbf{x}_0$)
Input: Labeled training data
$\mathcal{M}^{\mathcal{L}} = \{(\mathbf{x}_1, l_1), \dots, (\mathbf{x}_n, l_n)\}$ and test date $\mathbf{x}_0$
Output: Label prediction l of $\mathbf{x}_0$

```
1:  for $j = 1, \dots, k$ do
2:    dist$_k(j) = \infty$
3:    label$_k(j) = 0$
4:  end for
5:  for $i = 1, \dots, n$ do
6:    $d = d(\mathbf{x}_i, \mathbf{x}_0)$
7:    $j = 1$;
8:    while $j < k + 1$ do
9:      if $d <$ dist$_k(j)$ then
10:       for $p = k, \dots, j + 1$ do
11:         dist$_k(p) =$ dist$_k(p - 1)$
12:         label$_k(p) =$ label$_k(p - 1)$
13:       end for
14:       dist$_k(j) = d$
15:       label$_k(j) = l_i$
16:       $j = k + 1$
17:     else
18:       $j = j + 1$
19:     end if
20:   end while
21: end for
22: $l = \text{argmax}_{l \in \mathcal{L}} \{ \ |\{i : \text{label}_k(i) = l, \ i \in \mathcal{K}\}| \ \}$
23: return($l$)
```

Algorithm 3 Representative KNN: RKNN($\mathcal{M}^{\mathcal{L}}, \mathbf{x}_0$)
Input: Labeled training data $\mathcal{M}^{\mathcal{L}} = \{\mathcal{M}_1, \dots, \mathcal{M}_L\}$ with $\mathcal{M}_i = \{(\mathbf{x}_1^i, i), \dots, (\mathbf{x}_{m_i}^i, i)\}$, $i \in \mathcal{L}$ and test date $\mathbf{x}_0$
Output: Label prediction l of $\mathbf{x}_0$

```
1: for $i = 1, \dots, L$ do
2:   $(\mathbf{r}_1, \dots, \mathbf{r}_k) = k$Means$(\mathcal{M}_i)$
3:   $\mathcal{R}_i = ((\mathbf{r}_1, i), \dots, (\mathbf{r}_k, i))$
4: end for
5: $l =$ KNN$(\{\mathcal{R}_1, \dots, \mathcal{R}_L\}, \mathbf{x}_0)$
6: return($l$)
```

In Algorithm 2, dist$_k(i)$ denotes the distance of the ith nearest training data point $\mathbf{x}_{j_i}$ with $i = 1, \dots, k$ to $\mathbf{x}_0$. Since the KNN algorithm is able to classify data directly into more than two classes, it is well suited for localization. Furthermore, it is outlier resistant as it uses only the k nearest signals for the prediction and not the far away lying outliers.

C. Representative k-Nearest-Neighbor

Suppose we are given labeled recordings from different locations in the room. Let $\mathcal{M}^{\mathcal{L}} = \{\mathcal{M}_1, \dots, \mathcal{M}_L\}$ be the set of labeled training data and $\mathcal{M}_i$ a subset containing all signals with label $i \in \mathcal{L} = \{1, \dots, L\}$. For every set $\mathcal{M}_i$, we apply the k-means algorithm with $k < |\mathcal{M}_i| = m_i$ and obtain k labeled representatives $\mathcal{R}_i = \{(\mathbf{r}_1^i, i), \dots, (\mathbf{r}_k^i, i)\}$ with label i for each set $\mathcal{M}_i$, $i = 1, \dots, L$. To classify a test signal $\mathbf{x}_0$, we apply KNN on the labeled training data $\mathcal{R}^{\mathcal{L}} = \{\mathcal{R}_1, \dots, \mathcal{R}_L\}$. The pseudo-code is presented in Algorithm 3.
For the classification of Indonesian news [10], the combination of k-means and k-nearest-neighbor was also useful.

III. Technical Implementation

In this section, we give a detailed description of our field test. Since our goal is to make IoT-applications easily available, we start with an introduction to our low cost equipment. Especially, we introduce the IoT-Kit depicted in Figure 3, which we developed together with the expert group Internet of Things on the Digital Gipfel. After the description of our experimental setup, the M2M-communication and the preprocessing of the recorded signals will be explained. Finally we visualize the results of our field test.

A. Hardware IoT Device

The main circuit diagram of the IoT device Octopus is presented in Figure 3. The device uses the analog-digital converter (ADC) to measure the impulse responses. Although it was primarily developed for educational purposes, it has the potential to advance the digitization of companies. Since the IoT device is equipped with two grove-connectors, various sensors like our microphone can be easily connected. As the name suggests, the IoT device provides Internet access and is therefore flexible in communication. In our field test, for example, we used Message Queue Telemetry Transport (MQTT) for the transmission of signal recordings, which are

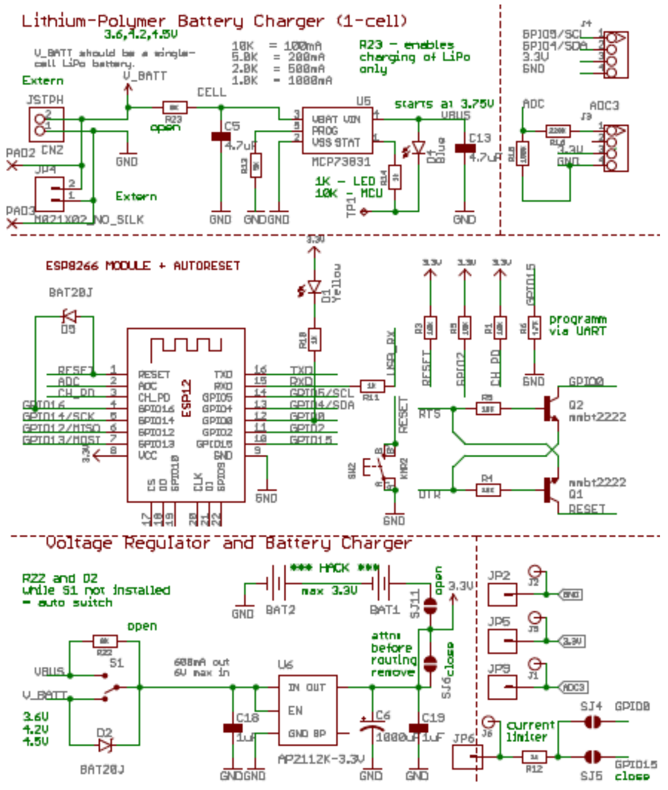
Lithium-Polymer Battery Charger (1-cell)
ESP8266 MODULE + AUTORESET
Voltage Regulator and Battery Charger
MCP73831
ESP12
AP2112K-3.3V

Fig. 3. The circuit of the used IoT device [11].

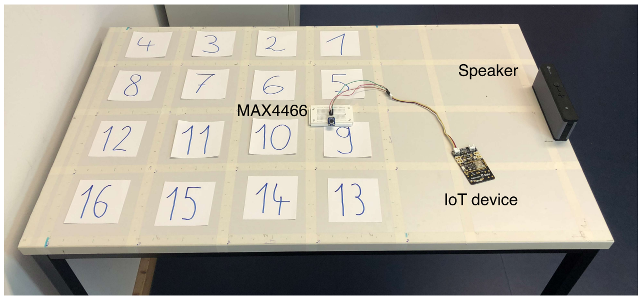

Fig. 4. The measurement setup.

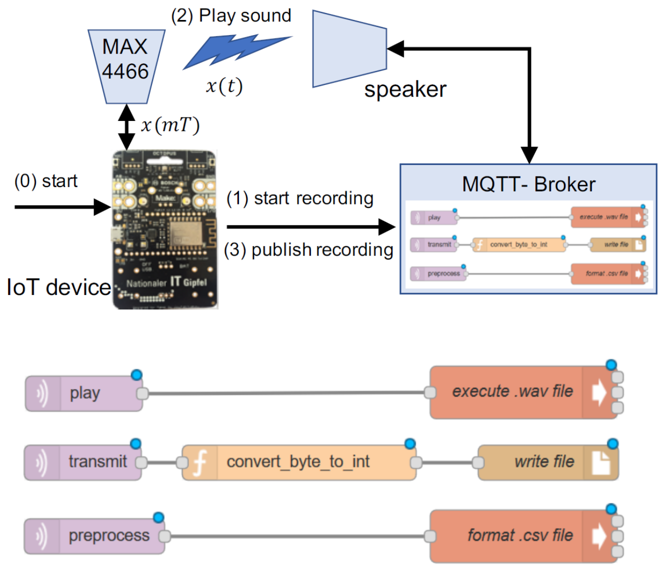

Fig. 5. Block diagram for communication and preprocessing.

explained in the communication and preprocessing section. The main core of the device is an ESP8266 [16], a very low-cost core with wireless local area network (WLAN). The presented customized device provides various sensors and actuators [11]. In our case, we use the ADC of the IoT-Kit to convert the analog signal recorded by the low cost microphone MAX4466 [12] with a sampling rate of $f_a = 10$ kHz. As analog input signal to obtain the room impulse response $s(t)$, we use a 100 Hz sinus sound with a duration of 1 ms played by a ordinary bluetooth speaker [13].

B. Experimental Setup

The measurement setup is depicted in Figure 4. Instead of recording room impulse responses throughout the room at great expense, we restricted the measurements to one table and predicted in which area of the table the microphone lies. Therefore we divided the table into 16 squares and assigned a label to every square. To check and visualize the results, the (x, y) position of the microphone is noted for each recorded signal. In the following section communication and preprocessing of the measurement sequence is described with the help of a block diagram given in Figure 5.

C. Communication and Preprocessing

Figure 5 shows the information flow during each test period. Triggered by a user command the IoT device instructs the speaker to play the input sound signal. Meanwhile, the IoT-Kit starts recording 10.000 samples $x(mT)$ at a sample rate of $f_a = 1/T = 10$ kHz. After one second of recording, the IoT device uses the MQTT protocol [17] to send raw data to the cloud computer, an ordinary Core i5 personal computer. Both communication setup and data preprocessing are realized via Node-RED [14] in order to prepare the raw data for the use of the KNN algorithm in Matlab. Node-RED enables the organization of data and communication flows using MQTT and other protocol standards. It also offers the possibility to manipulate data using JavaScript.
Figure 5 also shows the structure of the processes in our experiment. The cloud computer subscribes to the MQTT topics broad-casted by the IoT device. Topic *play* receives the command to execute a python script playing a sound file. Due to limited storage size on the IoT device the raw data is stored as a byte array. This array is split into byte-packages which are received and transformed back to integer values via topic *transmit*. Finally, topic *preprocess* gives the command to filter the relevant time frame of the recording and to convert the resulting data of the room impulse response into the csv format for further use in Matlab.

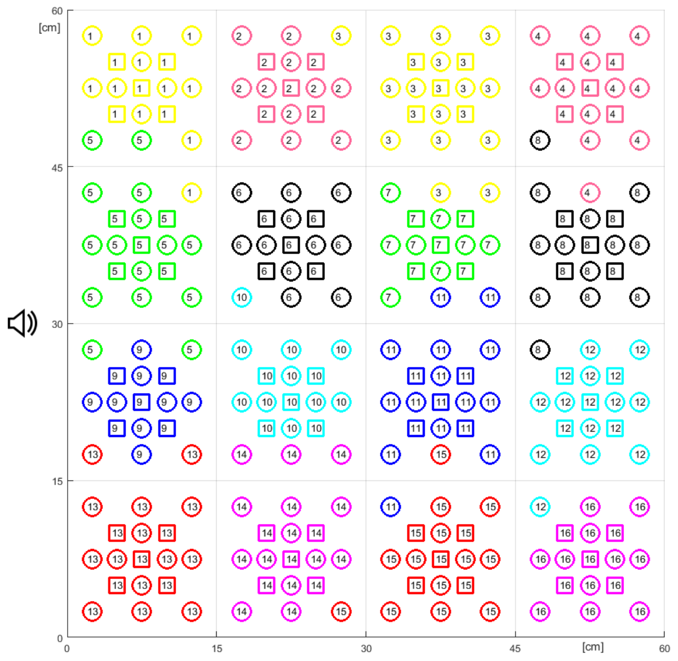

Fig. 6. Results of the RKNN classifier with 3 representatives for every label and 1 nearest neighbor for classification

D. Results

After collecting 272 measurements at different positions of the table, the measurements are divided into training and test data. The training data with 80 samples is replaced by the representatives of the k-means clusterings and fed in the KNN classifier. The results of a representative 1-nearest-neighbor classifier with 3 representatives for every label are visualized in Figure 6. In this visualization position and label of the original training data before clustering is given by the position of the rectangles and the numbers in it. Labels surrounded by circles depict the label predictions of the measurements located at the position of the circles.
Our experiment has an accuracy of 88.02%. It can be observed that label predictions near training recordings are correct. Only some recordings at the boundaries are misclassified. The accuracy can be improved by taking further training samples. Note, that only the area in which the microphone lies is determined and not the exact position of the microphone. In order to give a precision in cm, one can choose the middle of a square as a reference point. Then, a correctly classified recording is at most 10.61 cm (half a diagonal of a square) away from the corresponding reference point. If all recordings were correctly classified, we would achieve a precision of 10.61 cm. In order to improve the precision, one has to further divide the localization squares. In our experiment, incorrectly classified recordings are at most 11.19 cm away from the predicted reference point. That is why one can say that the localization in our experiment is up to 11.19 cm precise.
The recordings and the representative k-nearest-neighbor classifier used in our experiment can be found in [18].

IV. Conclusion

The results of the field test reveal the performance of the presented concept for indoor localization via room impulse response using a representative k-nearest-neighbor classifier. The classification has high accuracy and can be used for semantic localization in indoor environments. In a future work, we will extend this concept to online learning to enable a robust solution for changing environments.

Acknowledgment

This project was funded by the Federal Ministry of Education and Science (BMBF) grant 01IS17073.

References

[1] H. Laux, A. Bytyn, G. Ascheid, A. Schmeink, G. Karabulut Kurt and G. Dartmann; ”Learning-Based Indoor Localization for Industrial Applications” in Workshop on Sensor Data Fusion and Machine Learning for next Generation of Cyber-Physical-Systems in conjunction with ACM International Conference on Computing Frontiers, (2018).
[2] P.M. Hofman, J.G.A. Van Riswick and A.J. Van Opstal; ”Relearning sound localization with new ears” in Nature Neuroscience 1, pp. 417–421 (1998); doi: 10.1038/1633,
[3] R.F. Brena, J.P. García-Vázquez, C.E. Galván-Tejada, D. Muñoz-Rodriguez, C. Vargas-Rosales and J. Fangmeyer, Jr.; Evolution of Indoor Positioning Technologies: A Survey in Journal of Sensors, vol. 2017, 21 pages, (2017); doi:10.1155/2017/2630413.
[4] X. Guo, S. Shao, N. Ansari and A. Khreishah; ”Indoor Localization Using Visible Light Via Fusion Of Multiple Classifiers” in IEEE Photonics Journal, vol. 9, no. 6, pp. 1–16 (2017); doi: 10.1109/JPHOT.2017.2767576.
[5] A. Mandal, C. Videira Lopes, T. Givargis, A. Haghighat, R. Jurdak and P. Baldi; ”Beep: 3D Indoor Positioning Using Audible Sound” in IEEE Consumer Communications and Networking Conference (2005); doi: 10.1109/CCNC.2005.1405195.
[6] J.-M. Valin, F. Michaud, J. Rouat, D. Létourneau; ”Robust Sound Source Localization Using a Microphone Array on a Mobile Robot” in Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1228–1233 (2003); doi: 10.1109/IROS.2003.1248813.
[7] H. Blunck, T.S. Prentow, S. Temme, A. Thom, J. Vahrenhold; ”Deviation Maps for Robust and Informed Indoor Positioning Services” in SIGSPATIAL Special, pp. 27–34 (2017); doi: 10.1145/3124104.3124110.
[8] C.M. Bishop; ”Pattern Recognition and Machine Learning”, Springer-Verlag New York (2006).
[9] T. Hastie, R. Tibshirani, J. Friedman; ”The Elements of Statistical Learning”, Springer-Verlag New York (2009).
[10] P.W. Buana, D.R.M.S. Jannet, D. Putra; ”Combination of K-Nearest Neighbor and K-Means based on Term Re-weighting for Classify Indonesian News” in International Journal of Computer Applications, pp. 37–42 (2012); doi: 10.5120/7817-1105.
[11] fab-lab.eu, ’Octopus’ [Online]. Available: http://fab-lab.eu/octopus/ [Accessed: 21-Feb-2018].
[12] adafruit.com, ’Datasheets’ [Online]. Available: https://cdn-shop.adafruit.com/datasheets/MAX4465-MAX4469.pdf [Accessed: 21-Feb-2018].
[13] taotronics.com, ’TT-SK09’ [Online]. Available: https://de.taotronics.com/TT-SK09-Bluetooth-Lautsprecher.html [Accessed: 21-Feb-2018].
[14] nodered.org, ’Node-RED Flow-based programming for the Internet of Things’ [Online]. Available: https://nodered.org/ [Accessed: 21-Feb-2018].
[15] Expert Group Internet of Things [Online]. Available: http://deutschland-intelligent-vernetzt.org/wp/expertengruppen/expertengruppe-m2minternet-der-dinge/ [Accessed: 21-Feb-2018].
[16] ESP8266 [Online]. Available:http://www.esp8266.com [Accessed: 21-Feb-2018].
[17] MQTT [Online]. Available: http://mqtt.org [Accessed: 21-Feb-2018].
[18] Measurements and algorithm of results [Online]. Available: https://github.com/dziubany/ML-Localization [Accessed: 30-Mai-2018].