



Article

Scene Recognition for Indoor Localization Using a Multi-Sensor Fusion Approach

Mengyun Liu ¹ , Ruizhi Chen ^{1,2,*}, Deren Li ^{1,2,*}, Yujin Chen ³, Guangyi Guo ¹, Zhipeng Cao ¹ and Yuanjin Pan ¹ 

¹ State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing (LIESMARS), Wuhan University, Wuhan 430079, China; amylmy@whu.edu.cn (M.L.); guangyi.guo@whu.edu.cn (G.G.); godsay1983@whu.edu.cn (Z.C.); pan_yuanjin@163.com (Y.P.)

² Collaborative Innovation Center of Geospatial Technology, Wuhan University, Wuhan 430079, China

³ School of Geodesy and Geomatics, Wuhan University, Wuhan 430079, China; yujin.chen@whu.edu.cn

* Correspondence: ruizhi.chen@whu.edu.cn (R.C.); drli@whu.edu.cn (D.L.); Tel.: +86-150-7111-9753 (R.C.)

Received: 20 October 2017; Accepted: 28 November 2017; Published: 8 December 2017

Abstract: After decades of research, there is still no solution for indoor localization like the GNSS (Global Navigation Satellite System) solution for outdoor environments. The major reasons for this phenomenon are the complex spatial topology and RF transmission environment. To deal with these problems, an indoor scene constrained method for localization is proposed in this paper, which is inspired by the visual cognition ability of the human brain and the progress in the computer vision field regarding high-level image understanding. Furthermore, a multi-sensor fusion method is implemented on a commercial smartphone including cameras, WiFi and inertial sensors. Compared to former research, the camera on a smartphone is used to “see” which scene the user is in. With this information, a particle filter algorithm constrained by scene information is adopted to determine the final location. For indoor scene recognition, we take advantage of deep learning that has been proven to be highly effective in the computer vision community. For particle filter, both WiFi and magnetic field signals are used to update the weights of particles. Similar to other fingerprinting localization methods, there are two stages in the proposed system, offline training and online localization. In the offline stage, an indoor scene model is trained by Caffe (one of the most popular open source frameworks for deep learning) and a fingerprint database is constructed by user trajectories in different scenes. To reduce the volume requirement of training data for deep learning, a fine-tuned method is adopted for model training. In the online stage, a camera in a smartphone is used to recognize the initial scene. Then a particle filter algorithm is used to fuse the sensor data and determine the final location. To prove the effectiveness of the proposed method, an Android client and a web server are implemented. The Android client is used to collect data and locate a user. The web server is developed for indoor scene model training and communication with an Android client. To evaluate the performance, comparison experiments are conducted and the results demonstrate that a positioning accuracy of 1.32 m at 95% is achievable with the proposed solution. Both positioning accuracy and robustness are enhanced compared to approaches without scene constraint including commercial products such as IndoorAtlas.

Keywords: indoor scene recognition; deep learning; indoor localization; WiFi; magnetic field strength; particle filter; smartphone

1. Introduction

After decades of research, there are still no pervasive products for indoor localization while the demand for indoor localization-based service is increasing rapidly in smart cities [1]. Recent years have witnessed a lot of work on indoor localization. Most of them try to provide a widely used

scheme for indoor localization and achieve satisfying performance like GPS in outdoor environments. Among these [2–6], fingerprinting-based methods are the most popular due to their effectiveness and independence of infrastructure. Generally, fingerprinting-based methods include Wi-Fi and magnetic fingerprinting. Both of them are based on an assumption that each location has a unique signal feature. The system of fingerprinting localization is usually divided into two phases: offline training and online processing. In the offline phase, Wi-Fi-received signal strength (RSS) or magnetic field strength (MFS) at different reference points are collected to build a radio map. In the online phase, users can sample RSS or MFS data at their positions and search similar signal patterns in the database. The corresponding location with the most similar pattern is considered as a positioning result.

However, although this kind of scheme is easy to be established and can achieve fine performance in the first few months, it still is unable to become a widely used solution in the location-based service (LBS) market due to these challenges: (a) Data collection is labor-intensive and time-consuming. Surveyors need to collect enough signal samples at every reference point to build the fingerprint database. (b) Database maintenance is difficult, since signal patterns change over time and are vulnerable to environmental changes. In order to guarantee the accuracy of the indoor localization system, fingerprint database updating is required in an appropriate time interval, which is difficult and time-consuming. (c) Ambiguities in positioning results for the reason that different locations may have same signal patterns, especially for magnetic fingerprints. In most of the time, single source of signal is not enough to guarantee the accuracy of positioning result. (d) Low efficiency in fingerprint database matching. When the positioning area is considerable wide, the fingerprint database may be too large to give a positioning result in real-time on commercial smartphones.

To overcome the drawbacks of traditional fingerprinting-based indoor localization, researchers have proposed a large body of solutions. Among these solutions, a calibration-free mechanism which leverages motion information of users has been formed [4–7]. Whereas, the accuracy and robustness of these systems are not enough for wide use due to the complexity and diversity of indoor environments. To solve these problems, this paper presents a scheme which fully utilizes built-in sensors on a smartphone, including camera, accelerometer, gyroscope, compass, magnetometer and WiFi. Inertial sensors such as accelerometer, gyroscope and compass are used to record users' trajectories by pedestrian dead-reckoning (PDR). Every trajectory is construct by a step vector and every step consist of heading, WiFi RSS vector and magnetic field strength (MFS) vector. These steps are used to replace reference points in former fingerprinting-based systems. That is to say, in the proposed system, a fingerprint database can be built by walking trajectories of a user. A surveyor is not required to stay and wait at a reference point to collect signal samples. In this way, difficulties in database construction and maintenance are greatly decreased. Furthermore, inspired by our own spatial cognition experience in indoor environments, the "eye" (camera) of a smartphone is used to "see" the scene around a user. This procedure will give a priori information for localization. With scene information constrained, the searching range of fingerprint database is narrowed down and signal ambiguities are remarkably decreased as a result. To utilize scene information, every trajectory will be attached with a scene label in the database construction phase. To recognize a scene, an indoor scene model will be trained via fine-tuned deep convolutional neural networks (CNNs). The training data is captured by the camera on a smartphone. In the positioning phase, scene recognition will be first conducted. Then, a matching procedure will be carried out to narrow the fingerprint database to a small area (only contain fingerprints in that scene). Finally, a user's location will be estimated by a particle filter algorithm which fuses WiFi and MFS signals.

The main contributions of this paper can be concluded as follows.

- (1) To imitate the visual cognition ability of human, a scene recognition module implemented with deep learning is adopted to improve the accuracy and robustness of infrastructure-free indoor localization system.
- (2) A practical indoor localization scheme is designed which digging the full potential of built-in sensors on a smartphone. Particle filter algorithm is used in sensor data fusion.

- (3) The database construction in this paper is labor-saving and easily to be extended as a crowdsourcing solution for the reason that data is automatically collected while users' walking.
- (4) The proposed system is implemented on Android platform and performance is evaluated in a challenging indoor environment which includes several glass walls and a patio.

The rest of this paper is organized as follows. Related works are reviewed in Section 2. System architecture and methodology in proposed system are demonstrated in Section 3. Experiments and performance evaluations are presented in Section 4. Sections 5 and 6 are discussion and conclusion of the whole work and give suggestions in future research.

2. Related Work

The literature on fingerprinting-based indoor localization is long and rich. He et al. [8] overviews recent advances on WiFi fingerprinting localization. The survey includes two aspects. One is how to make use of signal patterns, user collaboration and motion sensors. Another is how to reduce labor-intensive database construction, maintain a fingerprint database and calibrate heterogeneous devices. Moreover, Yang et al. [9] supplies a comprehensive literature review on how mobility information enhances smartphone-based indoor localization. These studies indicate that sensor-fusion-based methods are main trends of future pervasive indoor localization services.

Earlier work for sensor-fusion-based indoor localization method can go back to [7], which utilizes commercial smartphones only. Although this system provides a reliable step detection and heading estimation algorithm, it requires users to provide initial location on an indoor map. LiFS systems [5] need little intervention to be deployed since mapping between fingerprint space and stress-free floor plan is created. In this process, mobility information is used to transform a floor plan to a stress-free floor plan and signal patterns are used to extract indoor space features including corridor and room. A more flexible deployment was proposed by Zheng et al. [10], which is a self-deployable system depending on trajectories recorded by guiders. In this system, followers can be navigated to their destination by recorded trajectories. Other similar studies also include [11–14]. Chen et al. [11] combined WiFi, pedestrian dead-reckoning (PDR) and landmarks with a Kalman filter algorithm. Liu et al. [12] employed spatial correlation extracted from user's motion. Chen et al. [13] adopted a maximum likelihood-based fusion algorithm to process WiFi and PDR information. While Shu et al. [14] use local disturbances of a geomagnetic field as features and incorporate WiFi signals to achieve a relatively high accuracy.

In the computer vision community, scene recognition has attracted lots of attention for it is a highly valuable perceptual ability of indoor mobile robots. Previous studies relate to indoor scene recognition including [15–22]. In [22], local and global information are combined to extract indoor scene prototype. In terms of practical training for deep CNNs, references [23–27] give different architectures for different vision tasks. Generally, with a large enough dataset, increasing the capacity of network can improve recognizing accuracy. Krizhevsky et al. [27] designed an eight-layer network to classify 1000 classes in ImageNet with a low error rate. Girshick et al. [23] proposed R-CNN (regions with CNN features) to improve object detection performance and achieved a remarkable enhancement. He et al. [24] allowed CNNs input arbitrary-size images with a spatial pyramid pooling strategy. References [25,26] devoted to understand features in hidden layers. Zhou et al. [25] showed that object detectors emerged inside CNNs trained for scene recognition. Yosinski et al. [26] studied how features in different layers affect the results of transfer learning based on a deep CNN. Considering recognition accuracy and efficiency, an eight-layer architecture is appropriate for our task.

Compared with previous schemes, this paper combines not only WiFi, PDR and geomagnetic features but also cameras to implement a high performance indoor localization system. The camera is adopted to extract the whole semantic information of locating area which noted as indoor scenes. Ref. [28] also combined an optical camera on a smartphone. However, the use of visual information in that work is different from the proposed system. They extracted SIFT features to calculate image matching factors along with orientation sensors. While this paper adopts deep CNNs to recognize

indoor scenes without any feature extraction which needs high-cost computation. To evaluate the performance, fingerprint database construction in this paper is similar to a trajectory-based system which is quite popular in both academia [4–7,10] and industry [29]. The major difference between the proposed system and other trajectory-based systems is that they have not taken visual information into account in localization process. For human experience, visual ability is important for us to locate ourselves and navigate. Applying this useful information to localization systems is helpful to improve the performance. In the proposed system, image data are simultaneously collected with other sensors such as WiFi, magnetometer, accelerometer, gyroscope and compass. Trajectories are collected and organized with scene labels. These scene labels can be used to give a boundary of fingerprint database in the localization phase.

Compared to other indoor scene recognition approaches, scene definitions in this paper are different since the aim of the proposed system is to accurately locate a user. Indoor scenes are labeled combined with localization areas in this paper. While other approaches label them based on their functionalities, such as corridor, bookstore, bedroom, library, gym and so on. In this system, not only category of a scene can be recognized, but also a specific one can be located. Regarding recognition methods of indoor scenes, deep learning is taken into account in this paper. Compared to [22], deep learning is more simplified since it is not necessary to define local or global features. The outputs of hidden layers can be regarded as features which are computed by feedforward neural networks. However, since deep learning is a data-hungry method, transfer learning can be adopted to overcome deficits of training data. In [15], Oquab et al. transferred image representations learned with CNNs on ImageNet dataset to other vision tasks. Besides, Espinace et al. [21] has proven that contextual relations of objects and scenes can facilitate indoor scene recognition. Therefore, a fine-tuned CNN is adopted to train the indoor scene recognition model in this paper, which is based on a pre-trained model with a large-scale dataset for object recognition [30]. In this way, even without a considerable dataset, an indoor scene model can still be trained to recognize a scene with high accuracy.

3. System Overview and Methods

In this section, an overview of the system is presented. Then, key modules and important algorithms are described, which include data acquisition, fingerprints processing, scene model training and location estimation.

3.1. System Overview

The architecture of the proposed system is shown as Figure 1. The whole procedure is similar to traditional fingerprinting localization schemes which contain two phases: offline training and online localization. In the offline phase, not only a fingerprint database should be built, but also an indoor scene recognition model will be trained through deep convolutional neural networks. In the online phase, scene recognition will be executed before location estimation to give a constraint for localization. We define scene labels in areas where it is hard to discriminate WiFi signal patterns or magnetic signal patterns. For instance, WiFi fingerprinting often fails to locate a user correctly in an open area such as a patio or a wide hall. While magnetic fingerprinting usually confuses locations for the reason that there are only three dimensions for magnetic measurements from the magnetometer in smartphones. More specifically, in the offline phase, the modules in the proposed system include data acquisition, fingerprint processing and indoor scene model training. In the online phase, the system includes scene recognition, location space narrowing and location estimation.

3.2. Data Acquisition and Fingerprints Processing

In this part, details of data acquisition and fingerprint processing are described. Both of these procedures are implemented on a smartphone.

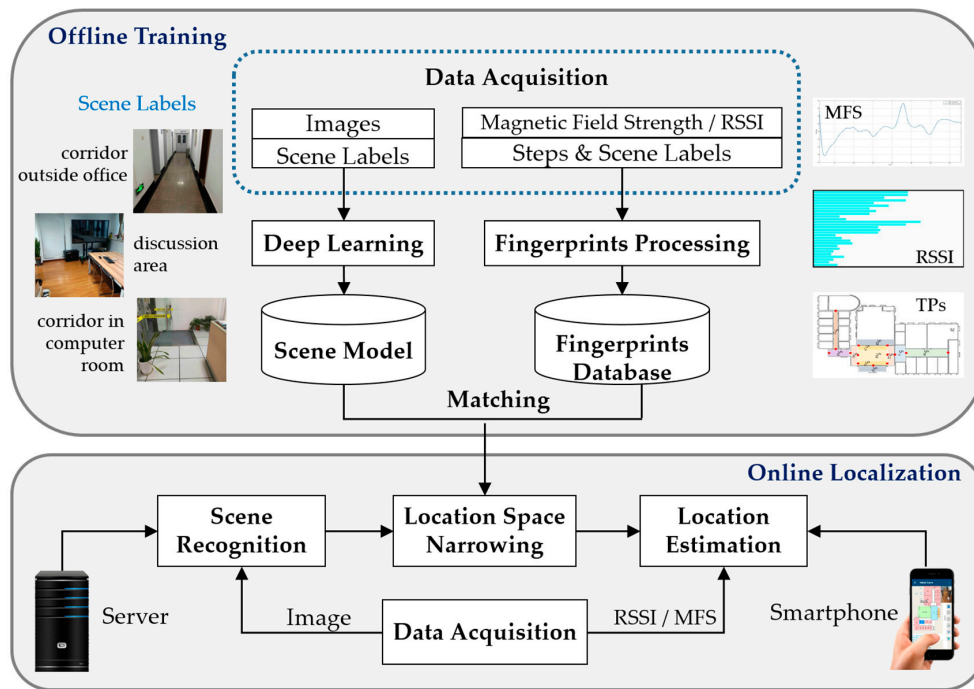


Figure 1. System architecture. MFS, RSSI and TP's refer to convolutional neural networks, magnetic field strength, received signal strength indicator and transition points, respectively.

3.2.1. Data Acquisition

In the stage of data acquisition, images are automatically captured along with the WiFi received signal strength indicator (RSSI) and magnetic field strength (MFS). In the meantime, a step detection module following [7] will be activated and heading of every step will be recorded. The illustration of data acquisition while walking is shown as Figure 2.

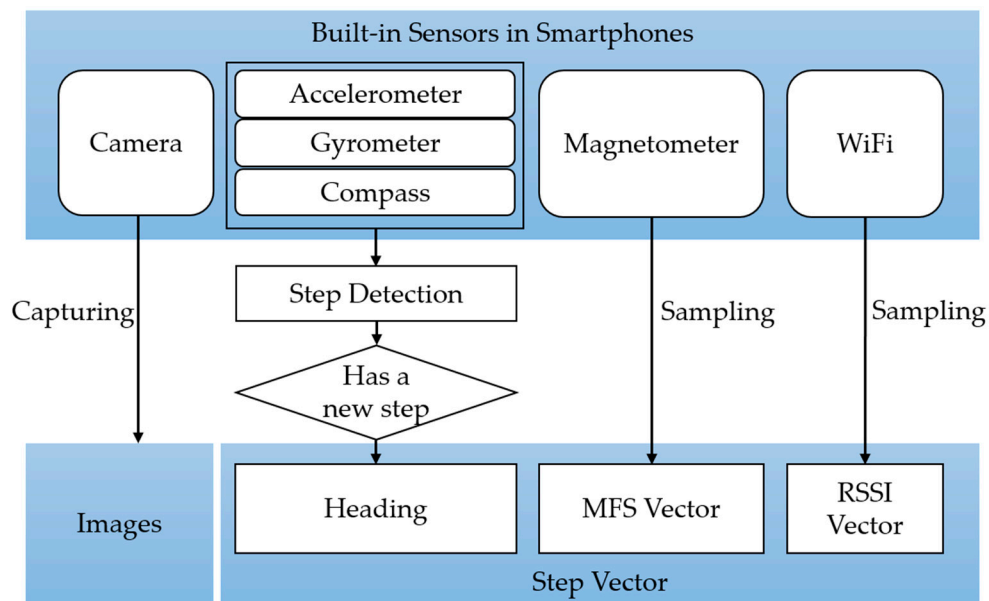


Figure 2. Data acquisition procedure while walking. Note that, WiFi RSSI vector is not required in every step, for its sampling frequency is slower than other sensors. If WiFi is not available, it will be set to null in a step vector.

When a step has been detected in a scene, the position of that point can be computed as:

$$\begin{cases} x_{i+1} = x_i + stepLength * \cos(r_i) \\ y_{i+1} = y_i + stepLength * \sin(r_i) \end{cases} \quad (1)$$

where r_i is the heading of $step_i$ and $stepLength$ can be predicted by an empirical model using Equation (2) proposed by [31].

$$stepLength = \left(0.7 + a(H - 1.75) + \frac{b * (StepFrequency - 1.79) * H}{1.75} \right) * c \quad (2)$$

a , b and c are model parameters for each person and can be calibrated by pre-training.

WiFi and MFS data are stored as vectors which are step-indexed. However, the sampling rate of WiFi is slower than other sensors. Thus, it is not necessary to wait for WiFi data at each step. In other words, if WiFi is not available, set it to null in this step. Since MFS data provided by a smartphone (Android) are raw data (in μT) for each of the three coordinate axes which are shown as Figure 3. The value of each axis is influenced by orientation of a smartphone. So we compute the sum of the three-dimensional raw measurements to represent the MFS.

$$M = \sqrt{m_x^2 + m_y^2 + m_z^2} \quad (3)$$

To guarantee the quality of data, we hold the smartphone in a fixed orientation as far as possible, such as 'in front of chest with 60° '.

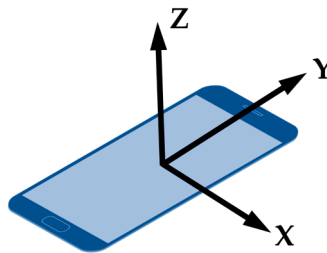


Figure 3. Three axes of magnetometer on smartphone.

To combine scene information with location appropriately, we introduce the definition of transition points (TPs). Most of the transition points are in the boundary of scenes. In the data collection process, transition points are stop stations for scene label change. In most cases, TPs are start points or end points of indoor scenes. As shown in Table 1, (x_0, y_0) is a start point of Scene S_1 , while (x_1, y_1) is a start point of Scene S_2 as well as end point of Scene S_1 , etc. For other data in the table, where $k_1, k_2, \dots, k_{(n-1)}$ are numbers of steps detected in a scene, while $r_1, v_{rssi\ 1}$ and $v_{mfs\ 1}$ are headings, vector of RSSI and vector of MFS at every step, respectively. A scene label can be seen as an attribute for TPs and steps.

Table 1. Information of data storage.

Scene Labels	TPs ¹	Steps	Heading	WiFi RSSI	MFS ¹
S_1	(x_0, y_0)	k_1	$(r_1, r_2, \dots, r_{k_1})$	$(v_{rssi\ 1}, v_{rssi\ 2}, \dots, v_{rssi\ k_1})$	$(v_{mfs\ 1}, v_{mfs\ 2}, \dots, v_{mfs\ k_1})$
S_2	(x_1, y_1)	k_2	$(r_1, r_2, \dots, r_{k_2})$	$(v_{rssi\ 1}, v_{rssi\ 2}, \dots, v_{rssi\ k_2})$	$(v_{mfs\ 1}, v_{mfs\ 2}, \dots, v_{mfs\ k_2})$
...
S_{n-1}	(x_n, y_n)	k_{n-1}	$(r_1, r_2, \dots, r_{k_{n-1}})$	$(v_{rssi\ 1}, v_{rssi\ 2}, \dots, v_{rssi\ k_{n-1}})$	$(v_{mfs\ 1}, v_{mfs\ 2}, \dots, v_{mfs\ k_{n-1}})$

¹ TPs and MFS indicate transition points and magnetic field strength.

Since deep learning is a kind of data-hungry method, a crowdsourcing method is designed to enlarge the training dataset. Images captured in the online phase also will be saved to server. Based on

our recognition task, 20 different scene labels are defined which cover two floors in our laboratory. Examples of these scenes in the third floor of our lab are shown as Figure 4. The whole scene labels will be demonstrated in experiments. There are two guidelines for us to define indoor scenes.

- Firstly, locations with the same scene label must have same semantic significance. For instance, as shown in Figure 4, point 3 belongs to room 313 while point 4 belongs to patio, it is obvious they are in different scenes.
- Secondly, locations with the same semantic significance but in sensitive areas are preferentially separated into different scenes. The sensitive areas here are more likely to achieve high error rate in the positioning phase. For instance, magnetic fingerprinting methods are difficult to distinguish locations at each side of a glass wall or door. As illustrated in Figure 4, the wall between room 313 and patio is a glass one. So the corridor inside room 313 and the corridor on the left of patio are defined as two different scenes to distinguish locations such as point 3 and point 4 even better.

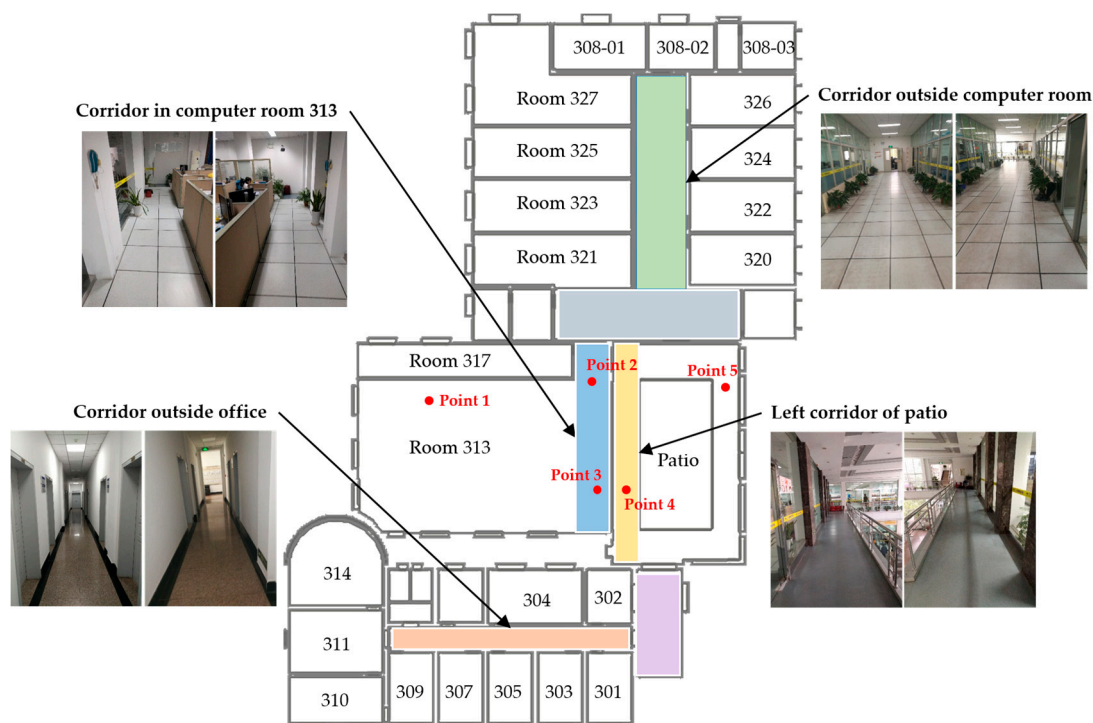


Figure 4. Examples of indoor scenes on the third floor of our laboratory. Two images are given in different walking directions.

3.2.2. Fingerprints Processing

The final database of fingerprints contains two parts, one is TP-indexed, and the other is step-indexed. As described in the previous section, when walking around in a scene, WiFi RSSI and MFS are collected and organized by step. In order to improve the localization accuracy at start points, a user should stop at TPs to collect WiFi signal for at least 30 samples. Therefore, TPs can be seen as robust reference points and used as initial points for pedestrian dead-reckoning (PDR). As illustrated in Figure 5a, we defined nine scenes in the third floor of a laboratory, and the corresponding number of transition points is ten. Thus, we have ten stable reference points in this floor, and other locations are indexed by detected steps. In Figure 5b, change of MFS in a corridor (S_9) is illustrated. It has shown that magnetic signal pattern is distinguishable in a five-step range, and it will be used to compute DTW distance in localization process.

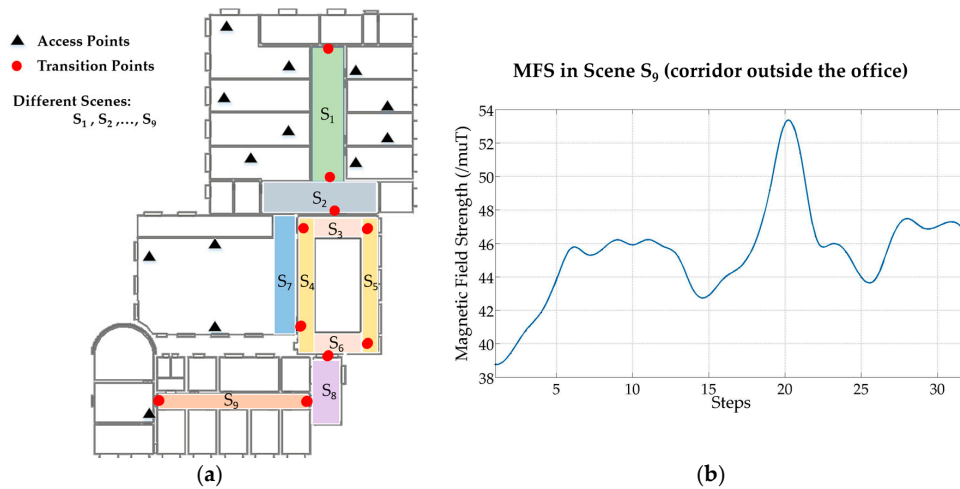


Figure 5. Environment information on the third floor of laboratory. (a) There are 13 access points, ten transition points and nine different scenes on this floor; (b) Change of MFS when walking in scene S_9 .

3.3. Indoor Scene Model Training

In this section, fundamentals of a deep convolutional neural network is described, as well as a fine-tuning method for the indoor scene model with our own dataset. The training is implemented on the server side with GPU. Although methods for training a deep model on mobile devices have been proposed by some researchers [32,33], the capability of a modern smartphone is still not sufficient for this large computation requirement. In this paper, we use Caffe [34], an open source framework of deep learning, to implement model training and scene recognition. This framework is widely used in the computer vision community, and lots of trained Caffe models for different tasks with all kinds of architectures and datasets are provided in Caffe Model Zoo [35].

3.3.1. Deep Convolutional Neural Networks (CNNs)

As illustrated in Figure 6, the architecture of CNNs used in the proposed scheme is similar to CaffeNet [36], which is modified from AlexNet [27] and provided by Berkeley AI Research (BAIR). As we can see, this network consist of five convolutional layers (conv1, conv2, conv3, conv4, conv5) and three fully connected layers (fc6, fc7, fc8), which conv1, conv2 and conv5 are followed with pooling layers (pool1, pool2, pool5) as well as normalization (norm1, norm2, norm5). Note that the outputs of fc8 are 1000, which indicates the number of classes that need to be predicted for ImageNet datasets. For our task, there are 20 scenes in our dataset. Therefore, the output of fc8 is 20 in our networks.

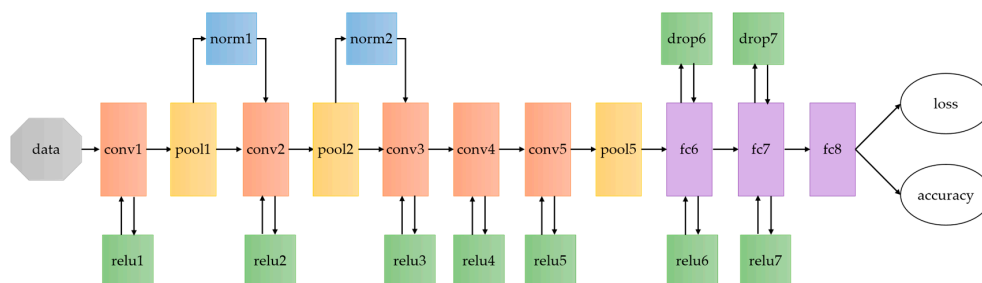


Figure 6. Architecture of CaffeNet. CaffeNet is a deep convolutional neural network modified from AlexNet with two differences: (1) training without the relighting data-augmentation; (2) pooling is switched to be done in front of normalization. CaffeNet is the base network for the proposed indoor scene model training.

For more details, specification of CaffeNet is given as Table 2. A convolution computation [37] can be denoted as Equation (4), where I is a two dimensional image as input, and K is a two-dimensional kernel.

$$S(i, j) = (I, K)(i, j) = \sum_m \sum_n I(i + m, j + n)K(m, n) \quad (4)$$

$$g(G, V, s) = \frac{\partial}{\partial K_{i,j,k,l}} J(V, K) = \sum_{m,n} G_{i,m,n} V_{j,(m-1) \cdot s+k,(n-1) \cdot s+l} \quad (5)$$

In the training process, derivations with respect to the weights in kernel need to be computed. To do so, we can use a function as (5), where s stands for stride in convolution, V is multichannel image, G is a tensor which computed by cost function J . The last layer of CaffeNet is a loss layer, which utilize SoftmaxWithLoss denoted as (6), which z_j stands for j th category in a recognition task.

$$\tilde{l} = -\log \left(\frac{e^{z_y}}{\sum_{j=1}^m e^{z_j}} \right) = \log \left(\sum_{j=1}^m e^{z_j} \right) - z_y \quad (6)$$

Table 2. Specification of CaffeNet.

Layers	Kernel Size	Stride	Pad	Output
conv1	11	4	0	96
pool1	3	3	0	96
con2	5	1	2	256
pool2	3	2	0	256
conv3	3	1	1	384
conv4	3	1	1	384
conv5	3	1	1	256
pool5	3	2	0	256
fc6	/	/	/	4096
fc7	/	/	/	4096
fc8	/	/	/	20

3.3.2. Fine-Tuned Deep CNNs for Indoor Scenes Recognition

As mentioned before, deep learning needs large amounts of training data, which is difficult for a specific vision task like ours. However, for the representation power of CNNs, trained model based on ImageNet, which is a large-scale object dataset, can be used as initial value in our training process. Since high level features are already learned in hidden layers, it is not necessary to train our model from scratch. The fine-tuning process based on our dataset is shown in Figure 7. Features learned from ImageNet are used to be transferred to our classification task.

As demonstrated in Figure 7, outputs of the final layer are redefined to fit the categories of indoor scenes in our task. After getting the pre-trained network on ImageNet, we fine-tune parameters in the internal layers (conv1–fc7) to decrease the loss computed by the last layer (fc_new). Fine-tuning is a back propagation process which updates weights of parameters in hidden layers to decrease the loss. The most important tip for fine-tuning is the initial learning rate which need to be small to avoid over-fitting.

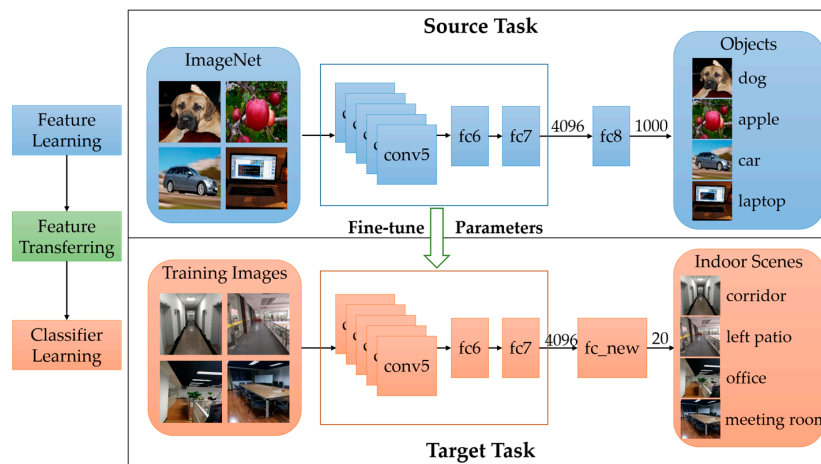


Figure 7. Fine-tuning a pre-trained network for indoor scene recognition. The outputs of final fully-connected layer for indoor scene task are 20 instead of 1000 in CaffeNet.

3.4. Location Estimation

In the online phase of the proposed system, locations can be estimated by three steps: current scene recognition, localization space narrowing and final localization estimation.

3.4.1. Indoor Scene Recognition

After training the indoor scene classifier, it will be deployed on a web server which can communicate with smartphones and receive recognition requests from clients. In our case, images from smartphones are captured automatically without special requirement for users, such as stop walking and deliberate focus on the current scene.

Before recognizing, preprocessing is needed to make test images meet the requirements of CaffeNet. The default configuration of CaffeNet is like this: format of image is BGR, pixel values start in the range of [0, 255] and subtract the mean pixel values of training data from them. The results present as top-5 probabilities with scene names, and we take top-1 as a result of scene recognition. However, it is a big challenge to allow image capturing to run all the time for it is extremely power-consuming. Thus, detection of proper time for capturing is needed. To solve this problem, we utilize TPs to detect potential scene change (Algorithm 1). k-Nearest Neighbor (kNN) algorithm is adopted during this process. Note that, the result of scene change detection is not the only activator for image capturing. It can also be activated by user input when necessary.

3.4.2. Space Narrowing and Localization

In this part, a fusion-based method is adopted to estimate user's location. As described previously, all of the TPs or steps in a fingerprint database that have an attribute of an indoor scene. After the current scene is recognized, we use this attribute to reduce searching space of locations. In other words, users can be located in a sub-fingerprint database. Thus, localization error will decrease in sensitive areas which are hard to distinguish in other indoor localization systems.

The process of localization is shown in Figure 8. When the app starts, it will lock-on to a transition point and start recognition. After the current scene is detected, the fingerprint database is narrowed to this scene. Then, particle filter is used in this sub-fingerprints database. The transition point locked-on is regarded as a start point to spread particles. When a step is detected, the position of each particle is updated by corresponding step length and heading. And each particle is weighted by WiFi signal distance and magnetic dynamic time warping (DTW) distance. The final result is computed by the centroid of weighted particles. In this work, results of particle filter are constrained in a recognized scene. When the scene is changed, it will be locked-on to a new transition point.

Algorithm 1. Transition Point Detection.

```

Input: Samples of WiFi RSSI
Output: TP's position
Begin: minimum distance equals infinite, nearest TP equals TP1
1: for every TP fingerprint in database do
2:   while n < number of APs scanned do
3:     compare mac address of every AP with TP fingerprint
4:     if mac address is matched
5:       number of matched APs ++
6:     end if
7:   end while
8:   if number of matched APs > 3
9:     compute signal distance between samples and TP
10:    if distance < minimum distance
11:      update the minimum distance
12:      update nearest TP
13:    end if
14:  end if
15: end for
16: compare scene label of nearest TP to current scene
17: if not matched
18:   return true
19: else return false
20: end if

```

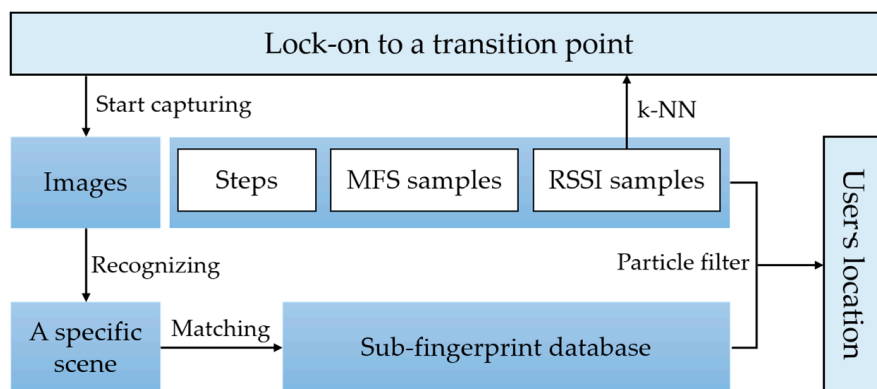


Figure 8. Procedures of localization estimation. Lock-on to a transition point is needed when the localization start. The detection of transition point utilize a k-NN algorithm.

For magnetic signal, two different places may have the same value since MFS is scalar. As can be seen in Figure 5b, MFS sequences are stable and distinguishable. Hence, MFS sequences are taken into account to measure similarities between samples and magnetic fingerprints. We use DTW to compute the similarity of two MFS sequences. DTW is a widely used algorithm in speech recognition which measures similarity between two different waveforms regardless of different speech speeds. Hence, different walking speed of users can be ignored by using the DTW algorithm in this case. However, it is better that length of MFS sequence is not too long or too short considering computation cost and accuracy. In the proposed system, a five-steps length is appropriate for localization considering balance of computation and accuracy. Figure 5b also has shown that a five-step length has good identifiability.

For WiFi signal, we compute the distance between scanned measurements and fingerprints by Equation (7).

$$D_{wifi} = \frac{1}{n} \sum_{i=1}^n |rssi_i^s - rssi_i^f| \quad (7)$$

where D_{wifi} denotes distance of two RSSI vectors, $rssi_i^s$ and $rssi_i^f$ are RSSI of i th AP from scanned measurements and database, respectively.

Then, fusion weight of each particle can be updated by Equation (8), where k_1 and k_2 denote tunable parameters, D_{mag}^i and D_{wifi}^i represent DTW distance and WiFi signal distance respectively.

$$weight^i = e^{\frac{D_{mag}^i}{k_1}} \cdot e^{-\frac{D_{wifi}^i}{k_2}} \quad (8)$$

Due to the noise of the WiFi signal, we set magnetic signal with higher weight than WiFi. In some detected steps, WiFi signal may be absent for the reason that the WiFi sampling rate is slower than other sensors. In that case, we only assign weight via magnetic signals.

4. Performance Evaluation

In this section, experiments of key modules in the proposed system are conducted. We test the accuracy of localization in a complicated indoor environment which contains glass walls and a rectangle patio between the second and third floors. Comparisons with two state-of-art schemes are also conducted. One is based on trajectories which combine PDR, WiFi and Magnetic. The other is IndoorAtlas [29], which is a commercial indoor localization platform based on a hybrid method, which combines PDR, WiFi, Magnetic, iBeacon and waypoints.

4.1. Experiment Setup

The experiments were conducted on the second and third floors of a laboratory building, with a 57.5 m × 41.5 m floor area. In order to verify the effectiveness of the proposed system, areas that may be easy to be confused in other indoor localization system were taken into account in our experiments, such as hall, patio area and corridors on either side of a glass wall, etc. We test 202 points covering the main paths which users visit often. As shown in Figure 9, 109 test points belong to the second floor and 93 test points belong to the third floor. The ground truth is measured by a laser distance meter with a 1.5 mm precision. To evaluate the performance of the proposed system, data collection and localization modules were implemented on an Android platform (version 6.0) and tested with Huawei Honor 8. The interface of application is shown in Figure 10.

The procedure of data collection has been mentioned in Section 3.2. In this experiment, only RSSIs stronger than −96 dBm are considered. Image data are collected in daytime and during a normal walking speed. The resolution of images are set to 240 × 320 which balances the quality and processing overheads. In this experiment, the number of indoor scenes are 20 and corresponding transition points are 24. Figure 11 shows the distribution of these scenes and transition points. The rule for us to divide these scenes is based on their semantic meanings and whether they are sensitive areas for localization. An example of different semantic meanings is that S_7 is a corridor in front of a computer room and S_4 is a corridor in a patio's left side. An example of sensitive areas is that S_3 , S_4 , S_5 and S_6 are all around the patio, however, position errors occurred frequently in these areas due to their low signal discrimination. Without scene constraints, positioning results may bounce between these scenes. This phenomenon is particularly evident with IndoorAtlas and other signal-based methods. Examples of image details in these scenes can be found in Figure 4. For indoor scene recognition model, it was trained on a PC with an NVIDIA Titan X Graphic Card to get a fast training speed. To be noted that, other graphic cards or only with CPU are also supported. The architecture of the network is given in Section 3.3.

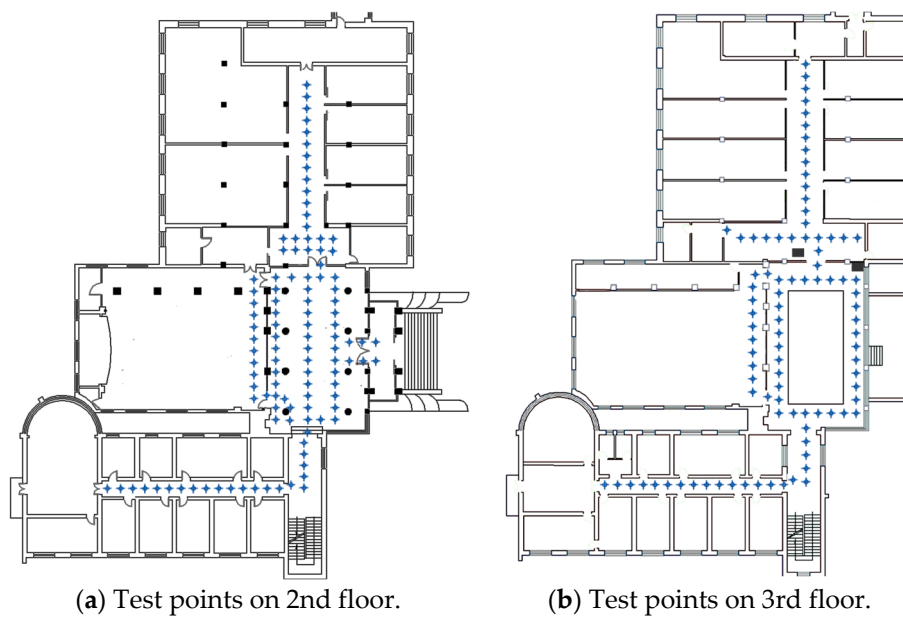


Figure 9. Test environment of the proposed system, which includes second and third floor of a laboratory in Wuhan University. (a) Floorplan of second floor, including 109 test points. (b) Floorplan of third floor, including 93 test points.

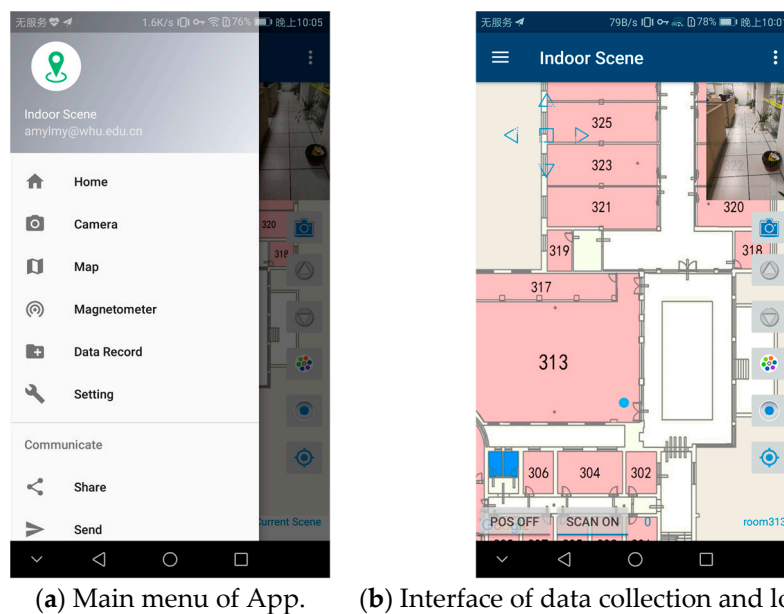


Figure 10. Interface of the proposed system on an Android phone. Before data recording start, we need to switch off the positioning mode, choose a current scene label and pick a start point on the map. Then, after switching on the scan mode, data recording will be activated.

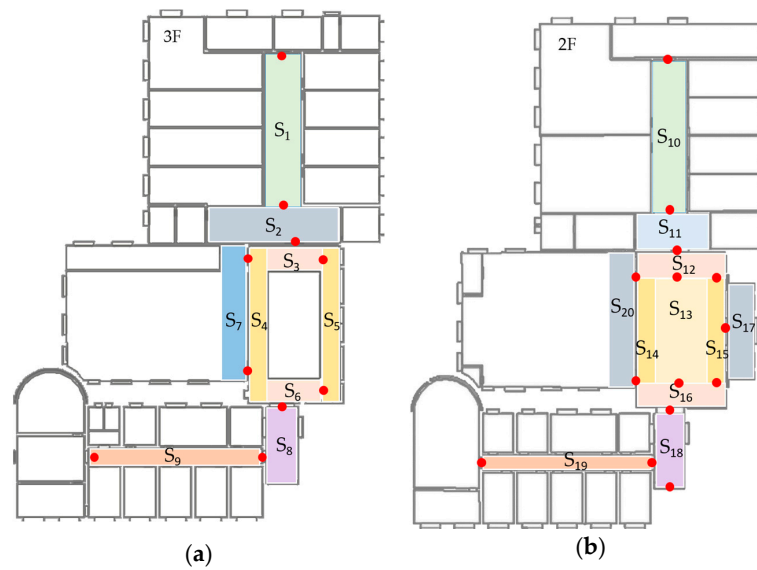


Figure 11. Definition of scene labels and transition points. (a) There are ten TPs and nine scenes in third floor; (b) 14 TPs and 11 scenes in second floor respectively.

4.2. Performance of Indoor Scene Recognition

Training a deep learning model from scratch is time-consuming and the volume of data is hard to be satisfied. Nevertheless, we can still use our own dataset, a relatively small one, to train an effective model for our task. In this experiment, a trained model published by other researchers in Caffe Model Zoo [35] was utilized as initialization in the training process. The specification of this network is provided in Section 3.3.1. Weights of parameters in our model were adapted to fit the training data by back-propagation computing.

In the training process, we gained a 97.5% accuracy after 6000 iterations on our own dataset, which includes 4800 images in total. Among them, 200 are used for training and 40 are used for validation in each scene. The changing trends of accuracy while training is displayed in Figure 12. As it can be seen, the speed of convergence is extremely fast. Testing of the trained model was also conducted in a test dataset. The report accuracy is 89.8% as shown in Table 3. It is important to be noted that in order to verify the generalization ability of this model, some bad-quality images are used in the test process. Figure 13 shows a recognition result on a web server. Both pictures were taken with a smartphone in room 313 of our lab. The first one was well-taken while the second one was heavily blurred for it was taken while walking. Even so, the top-1 result is correct and only it will be considered.

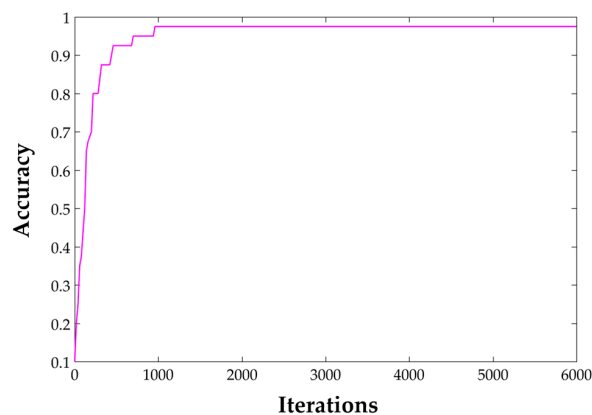


Figure 12. Accuracy change in training process.

Table 3. Score of model for indoor scene recognition.

Model	Iterations	Time Cost	Top-1 (val)	Top-1 (Test)
CaffeNet	6000	23'39"	97.5%	89.8%

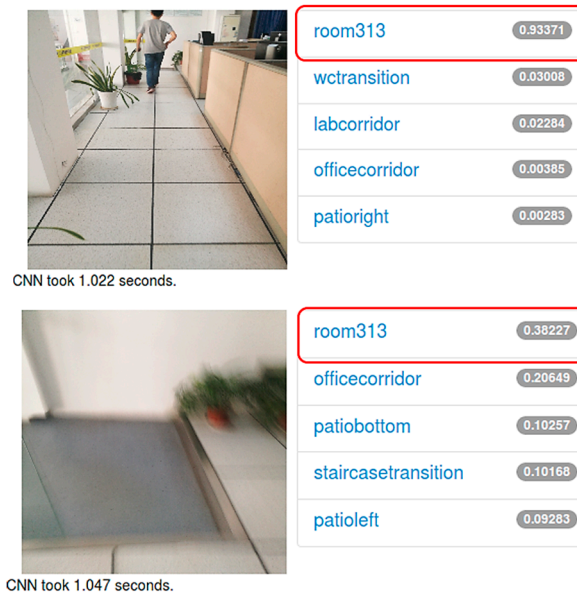


Figure 13. An example of indoor scene recognition with a bad-quality image. Both images were taken in room 313 of our lab. The second one is taken while a walking. The correct probabilities of these images are around 93.37% and 38.23% respectively.

4.3. Localization Results and Analysis

In this part, comparisons of localization results will be demonstrated. We compared our results with two different schemes: (1) a trajectory-based method which involves fusion with PDR, WiFi and magnetic; (2) IndoorAtlas which is a commercial hybrid scheme with PDR, WiFi, magnetic, iBeacon and waypoints.

Before localization, a fingerprint database needs to be built. In the proposed system, we recorded data by scenes while walking normally. The start point of a walking path needs to be a transition point (TP). As depicted in Figure 14, the path segments between different TPs (such as A2–B2, B2–C2, B2–D2 and so on) are recorded with scene labels (S_{19} and S_{18} , etc.). The fingerprints at TPs are recorded by standing still and collecting at least 30 samples. The data format has been discussed in previous sections. For every path segment, step vectors are utilized to organize the fingerprint data. Every step must have a MFS (magnetic field strength) vector and may have a RSSI vector for the sample rate of WiFi is slower than the magnetometer. Headings also are stored in step vectors. Step length in this experiment is 0.66 m which is determined by an offline training.

As mentioned previously, after a scene image is captured by a smartphone, it will be sent to a server for recognizing and the result will be sent back to the smartphone immediately. In our fingerprint database, trajectory segments in a scene can be considered as a sub-fingerprint database which could be indexed by a scene label. After a scene is recognized, the system will estimate locations with corresponding sub-fingerprint database. The start point is determined by transition points. In fact, there are two scenarios in which the detection of transition point needs to be executed. The first one is when the app starts, we need to lock-on the user to a transition point as a starting point. The other is when the scene is changed, it is also necessary to determine the transition point to reset the start point for PDR to avoid accumulative error.

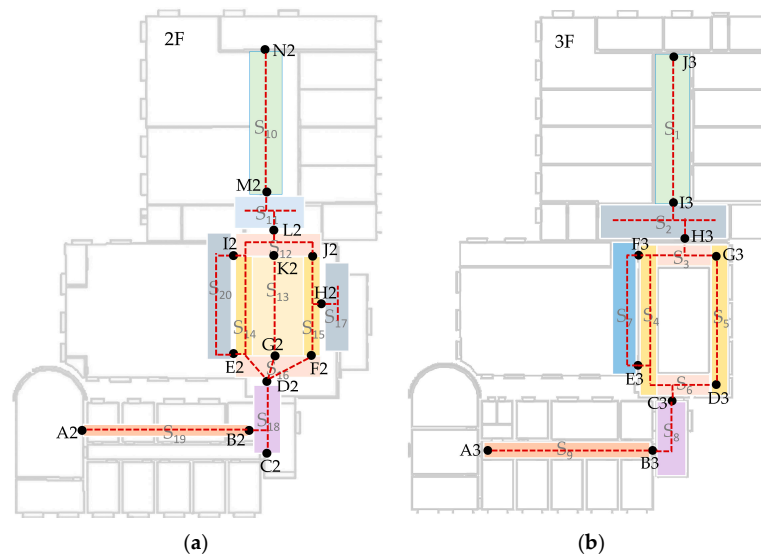


Figure 14. Paths to collect data for localization. (a) A2–N2 are TPs on the second floor and (b) A3–J3 are TPs on the third floor. In test phase, every path segments between two transition points were repeated at least three times.

In order to evaluate the performance of the proposed system, experiments of positioning without scene information were conducted. The first comparison is trajectory-based and involves fusion with multiple sensors such as IMU, WiFi and magnetometer. For comparison purposes, the storage format and collecting method of trajectories are similar to the proposed system. However, it does not contain any scene information. The second comparison is IndoorAtlas. The surveying method of IndoorAtlas is also by recording data while walking. As illustrated in Figure 15, mapping tool of IndoorAtlas requires users to define some waypoints for correction in the surveying phase. When a user arrives at a waypoint, a check-in operation needs to be carried out to tell the actual position of the user. The coverage of mapping is demonstrated with blue color in Figure 15. It covers all test points in this comparison experiment.

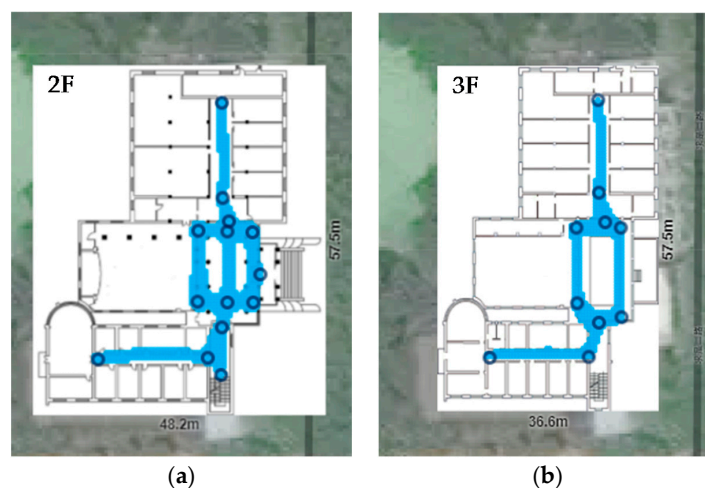


Figure 15. Coverage of site survey on IndoorAtlas. (a,b) are survey maps on the second floor and third floors, respectively. The black circles are waypoints we added in this experiment. A waypoint is pinpointed places on a map, and need to be check-in to define the actual position in data recording process.

Table 4 summarizes the error statistics of these three systems. The mean error of our system is 0.53 m which is better than the trajectory-based method without indoor scene and IndoorAtlas. The 95% accuracy in our system is 1.32 m which also has a significant advantage compared to others. The empirical cumulative distribution functions (CDFs) of these three systems are illustrated as Figure 16. With the constraint of indoor scene, the proposed system decreases the accumulative error of PDR, which is the main error source in trajectory-based systems. Moreover, it can also distinguish areas which have similar signal patterns, such as near a glass wall, patio etc. This is the reason why the proposed system can achieve higher accuracy than IndoorAtlas. Compared to our system, IndoorAtlas also defined waypoints to decrease the error of PDR which is similar to transition points in the proposed system. However, it fails to distinguish most of the positions in scenes S_7 and S_4 (Figure 11), since there is a glass wall between these two scenes. However, with the indoor scene recognition model, positions in these two scenes can be easily distinguished since sub-fingerprint databases corresponding to different scenes that are built in this paper.

Table 4. Comparison of error statistics.

Method	Mean Error	Variance	95% Accuracy
PDR + WiFi + Magnetic ¹	2.36 m	2.84	5.93 m
IndoorAtlas (Hybrid platform)	1.08 m	2.69	3.01 m
PDR + WiFi + Magnetic + Indoor Scene ²	0.53 m	0.16	1.32 m

¹ Trajectory-based method without indoor scene; ² The proposed system.

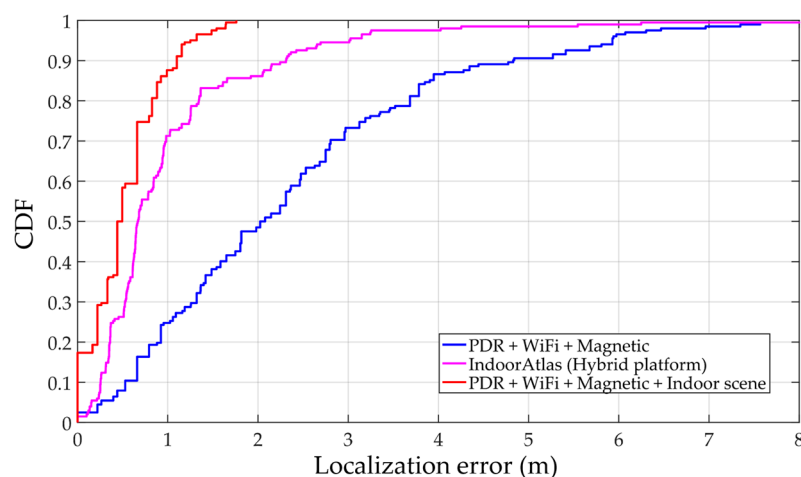


Figure 16. Empirical cumulative distribution function (CDF) of three different solutions. The proposed solution outperformed the conventional approaches without scene constraint.

5. Discussion

In this paper, we presented a multi-sensor fusion approach for indoor localization with scene constraint. A deep-learning method was adopted to recognize the current scene of the user. A recognition accuracy of 89.8% was achieved with our solution. A particle filter algorithm was then developed with constraints of scenes. Experimental results demonstrated that the positioning accuracy of our solution was 1.32 m at 95%, while that for the commercial product IndoorAtlas was 3.01 m in the same experimental environments. It is obvious that our solution outperformed conventional solutions without scene constraints. It is important to note that scene recognition cannot work well in dark environments or other bad visual conditions. Our system allows the user to switch the scene recognition module on or off to make sure it can be used in different kinds of environments.

Our solution has great potential to extend to a crowdsourcing-based method considering the data acquisition scheme. The trajectories from different users can be used to update and

enlarge our fingerprint database, and images captured in the online phase will also be saved in our server for future model training since the volume of training dataset is important for deep neural networks. In the future, the problems that we need to solve for crowdsourcing trajectories are device heterogeneity and data integration. For indoor scene recognition models, a more efficient network architecture and more complicated environments need to be considered in future work.

6. Conclusions

In this paper, a scene-constrained indoor localization scheme has been proposed. The results and comparisons have shown that this scheme has a competitive performance compared to most current systems. We solved indoor localization problems especially in complicated indoor environments. Most systems fail to give a high positioning accuracy with easy deployment. In our system, we do not need to extract features in images compared to other vision-based methods. Deep learning methods only need to assign labeled training data to a pre-defined network; it will learn features automatically.

We are confident that the multi-sensor-fusion-based method may become the “killer” scheme for indoor localization systems without additional infrastructures. Based on the state-of-art study of indoor localization, to the best of our knowledge, this work is the first to adopt deep learning to extract high level semantic information for indoor localization. This method is similar to human brain’s cognitive mode and has great potential in future research.

Acknowledgments: This study is supported by the National Key Research and Development Program of China (grant Nos. 2016YFB0502200 and 2016YFB0502201), NSFC (grant No. 91638203) and the Innovation Fund of Shanghai Aerospace Science and Technology (SAST, 2015014).

Author Contributions: This paper is a collaborative work by all authors. Mengyun Liu proposed the idea, implemented the system, performed the experiments, analyzed the data and wrote the manuscript. Ruizhi Chen and Deren Li helped to propose the idea, give suggestions and revise the rough draft. Yujin Chen, Guangyi Guo, Zhipeng Cao and Yuanjin Pan helped to do some of the experiments.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Li, D.; Shan, J.; Shao, Z.; Zhou, X.; Yao, Y. Geomatics for smart cities-concept, key techniques, and applications. *Geo-Spat. Inf. Sci.* **2013**, *16*, 13–24. [[CrossRef](#)]
2. Bahl, P.; Padmanabhan, V.N. RADAR: An in-building RF-based User Location and Tracking System. In Proceedings of the Nineteenth Joint Conference of the IEEE Computer and Communications Societies, Tel Aviv, Israel, 26–30 March 2000; pp. 775–784.
3. Youssef, M.; Agrawala, A. The Horus WLAN location determination system. In Proceedings of the International Conference on Mobile Systems, Applications, and Services, Seattle, WA, USA, 6–8 June 2005; pp. 205–218.
4. Rai, A.; Chintalapudi, K.K.; Padmanabhan, V.N.; Sen, R. Zee: Zero-effort crowdsourcing for indoor localization. In Proceedings of the 18th Annual International Conference on Mobile Computing and Networking, Istanbul, Turkey, 22–26 August 2012; pp. 293–304.
5. Yang, Z.; Wu, C.; Liu, Y. Locating in fingerprint space: Wireless indoor localization with little human intervention. In Proceedings of the 18th Annual International Conference on Mobile Computing and Networking, Istanbul, Turkey, 22–26 August 2012; pp. 269–280.
6. Yang, S.; Dessai, P.; Verma, M.; Gerla, M. Freeloc: Calibration-free crowdsourced indoor localization. In Proceedings of the 32nd IEEE International Conference on Computer Communications, Turin, Italy, 14–19 April 2013; pp. 2481–2489.
7. Li, F.; Zhao, C.; Ding, G.; Gong, J.; Liu, C.; Zhao, F. A reliable and accurate indoor localization method using phone inertial sensors. In Proceedings of the 14th ACM International Conference on Ubiquitous Computing, Pittsburgh, PA, USA, 5–8 September 2012; pp. 421–430.
8. He, S.; Chan, S.H.G. Wi-Fi Fingerprint-Based Indoor Positioning: Recent Advances and Comparisons. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 466–490. [[CrossRef](#)]

9. Yang, Z.; Wu, C.; Zhou, Z.; Zhang, X.; Wang, X.; Liu, Y. Mobility increases localizability: A survey on wireless indoor localization using inertial sensors. *ACM Comput. Surv.* **2015**, *47*, 1–34. [[CrossRef](#)]
10. Zheng, Y.; Shen, G.; Li, L.; Zhao, C.; Li, M.; Zhao, F. Travi-navi: Self-deployable Indoor Navigation System. In Proceedings of the 20th Annual International Conference on Mobile Computing and Networking, Maui, HI, USA, 7–11 September 2014; pp. 471–482.
11. Chen, Z.; Zou, H.; Jiang, H.; Zhu, Q.; Soh, Y.; Xie, L. Fusion of WiFi, Smartphone Sensors and Landmarks Using the Kalman Filter for Indoor Localization. *Sensors* **2015**, *15*, 715–732. [[CrossRef](#)] [[PubMed](#)]
12. Liu, J.; Chen, R.; Pei, L.; Guinness, R.; Kuusniemi, H. A hybrid smartphone indoor positioning solution for mobile LBS. *Sensors* **2012**, *12*, 17208–17233. [[CrossRef](#)] [[PubMed](#)]
13. Chen, L.H.; Wu, E.H.K.; Jin, M.H.; Chen, G.H. Intelligent fusion of Wi-Fi and inertial sensor-based positioning systems for indoor pedestrian navigation. *IEEE Sens. J.* **2014**, *14*, 4034–4042. [[CrossRef](#)]
14. Shu, Y.; Bo, C.; Shen, G.; Zhao, C.; Li, L.; Zhao, F. Magicol: Indoor localization using pervasive magnetic field and opportunistic WiFi sensing. *IEEE J. Sel. Area. Comm.* **2015**, *33*, 1443–1457. [[CrossRef](#)]
15. Oquab, M.; Bottou, L.; Laptev, I.; Sivic, J. Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 24–27 June 2014; pp. 1717–1724.
16. Khan, S.H.; Hayat, M.; Bennamoun, M.; Togneri, R.; Sohel, F.A. A Discriminative Representation of Convolutional Features for Indoor Scene Recognition. *IEEE Trans. Image Process.* **2016**, *25*, 3372–3383. [[CrossRef](#)] [[PubMed](#)]
17. Yu, J.; Hong, C.; Tao, D.; Wang, M. Semantic embedding for indoor scene recognition by weighted hypergraph learning. *Signal Process.* **2015**, *112*, 129–136. [[CrossRef](#)]
18. Gupta, S.; Arbelaez, P.; Malik, J. Perceptual organization and recognition of indoor scenes from RGB-D images. In Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 25–27 June 2013; pp. 564–571.
19. Kawewong, A.; Pimup, R.; Hasegawa, O. Incremental Learning Framework for Indoor Scene Recognition. In Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, Bellevue, WA, USA, 14–18 July 2013; pp. 496–502.
20. Espinace, P.; Kollar, T.; Roy, N.; Soto, A. Indoor scene recognition by a mobile robot through adaptive object detection. *Robot. Auton. Syst.* **2013**, *61*, 932–947. [[CrossRef](#)]
21. Espinace, P.; Kollar, T.; Soto, A.; Roy, N. Indoor scene recognition through object detection. In Proceedings of the IEEE International Conference on Robotics and Automation, St Paul, MN, USA, 14–18 May 2012; pp. 1406–1413.
22. Quattoni, A.; Torralba, A. Recognizing indoor scenes. In Proceedings of the 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), Miami, FL, USA, 20–25 June 2009; pp. 413–420.
23. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 24–27 June 2014; pp. 580–587.
24. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [[CrossRef](#)] [[PubMed](#)]
25. Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; Torralba, A. Object detectors emerge in deep scene cnns. *arXiv* **2014**, arXiv:1412.6856v2.
26. Yosinski, J.; Clune, J.; Bengio, Y.; Lipson, H. How transferable are features in deep neural networks? In Proceedings of the Neural Information Processing—21st International Conference, ICONIP 2014, Kuching, Malaysia, 3–6 November 2014; pp. 3320–3328.
27. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. In Proceedings of the Neural Information Processing—19th International Conference, ICONIP 2012, Doha, Qatar, 12–15 November 2012; pp. 1097–1105.
28. Chen, W.; Wang, W.; Li, Q.; Chang, Q.; Hou, H. A Crowd-Sourcing Indoor Localization Algorithm via Optical Camera on a Smartphone Assisted by Wi-Fi Fingerprint RSSI. *Sensors* **2016**, *16*, 410. [[CrossRef](#)] [[PubMed](#)]
29. IndoorAtlas. Available online: <http://www.indooratlas.com/> (accessed on 15 October 2017).

30. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Li, F.-F. ImageNet: A Large-Scale Hierarchical Image Database. In Proceedings of the 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
31. Chen, R.; Pei, L.; Chen, Y. A smart phone based PDR solution for indoor navigation. In Proceedings of the 24th International Technical Meeting of the Satellite Division of the Institute of Navigation, Portland, OR, USA, 20–23 September 2010; pp. 1404–1408.
32. Lane, N.D.; Georgiev, P. Can deep learning revolutionize mobile sensing? In Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications, Santa Fe, NM, USA, 12–13 February 2015; pp. 117–122.
33. Latifi Oskouei, S.S.; Golestani, H.; Hashemi, M.; Ghiasi, S. CNNdroid: GPU-Accelerated Execution of Trained Deep Convolutional Neural Networks on Android. In Proceedings of the 2016 ACM on Multimedia Conference, Amsterdam, The Netherlands, 15–19 October 2016; pp. 1201–1205.
34. Caffe|Deep Learning Framework. Available online: <http://caffe.berkeleyvision.org/> (accessed on 15 October 2017).
35. Model Zoo BVLC/Caffe Wiki. Available online: <https://github.com/BVLC/caffe/wiki/Model-Zoo> (accessed on 15 October 2017).
36. Caffe/Models/Bvlc_Reference_Caffenet at Master BVLC/Caffe. Available online: https://github.com/BVLC/caffe/tree/master/models/bvlc_reference_caffenet (accessed on 15 October 2017).
37. Goodfellow, I.; Bengio, Y.; Aaron, C. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016; pp. 120–121, ISBN 0262035618.



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).