Multimed Tools Appl (2017) 76:20317–20339
DOI 10.1007/s11042-017-4779-6

Indoor localization with a signal tree

Wenchao Jiang[1] · **Zhaozheng Yin**[1]

Received: 27 October 2016 / Revised: 29 March 2017 / Accepted: 28 April 2017 /
Published online: 24 May 2017

Abstract Smartphones embedded with cameras and other sensors offer possibilities to attack the problem of indoor localization where GPS is not reliable. In this paper, a novel tree-based localization system is proposed based on WiFi, inertial and visual signals. There are three levels in the tree: (1) WiFi-based coarse positioning. The WiFi database of a building is clustered into several branches for coarse positioning; (2) Orientation pruning. Images collected in a building are tagged with camera orientations towards which they are taken, so when inferring a user's location by comparing the query image the user takes with the reference image dataset, the image branches tagged with unmatched orientation will not be searched; (3) Fine visual localization. The user's location is accurately determined by matching the query image with the reference image dataset based on a multi-level image description method. Our signal tree based method is compared with other methods in terms of the localization accuracy, localization efficiency and time cost to build the reference database. Experimental results on four large university buildings show that our indoor localization system is efficient and accurate for indoor environments.

Keywords Indoor localization · Multimodal information fusion · Cross-media data analytics

1 Introduction

Indoor localization using consumer electronic devices (e.g., smartphones embedded with various sensors) has many applications such as localizing a patient in a hospital, navigating consumers in a big shopping mall or finding a safety egress during an emergency. The technology of indoor localization of human can also be applied to automatically position

✉ Zhaozheng Yin
yinz@mst.edu

[1] Missouri University of Science and Technology, Rolla, MO, USA

a robot in a building. Finding a smartphone carrier's location based on Global Positioning System (GPS) usually works well outdoors. However, GPS relies on unobstructed signals transmitted between satellites and devices on or near the earth. Walls, ceilings or other physical infrastructures within an indoor environment weaken or cut off GPS signals, making GPS-based indoor localization inaccurate [6]. Furthermore, it is difficult to localize which floor the phone-carrier is on because GPS is more sensitive to the horizontal disparity on the earth surface than the vertical difference.

1.1 Related work

The drawbacks of GPS inspire research ideas on other new indoor localization approaches [7, 9, 16], which can be divided into two categories in general: triangulation-based and fingerprint-based. *Triangulation-based* methods make use of geometric properties of triangles. The device sends signals to three or more stationary receivers and the device's position is estimated by the angle, travel time or strength of arrival signals on the receivers [8, 10, 18, 22, 29]. This method usually requires specific infrastructures and relatively high cost, which limits its application. Moreover, the accuracy of triangulation-based methods will greatly reduce when physical infrastructures, electronic equipment or signal interference influence the transmission of signals. Rather than calculating the geometric relationship between a device and receivers based on signal propagation, *fingerprint-based* methods pre-record "fingerprints" at different indoor positions [12, 20, 25, 32]. The term "fingerprint" means every position inside a building has a unique descriptor to identify it, just like the fingerprints of people. The fingerprint may be represented by different types of sensor signals such as WiFi [32], FM radio signals [20], etc. A device's indoor position is estimated by capturing a fingerprint at a location and matching it with fingerprints in the pre-built reference database. The performance of fingerprint-based methods largely depends on the quality and discriminative capability of signals used. A drawback of fingerprint-based method lies in the reference dataset collection. High accuracy indoor localization usually requires to densely collect reference data in a building.

The method of indoor localization can also be grouped by the sensor signals such as WiFi, Inertial Measure Unit (IMU) and visual signals. WiFi signal is a popular choice used in the indoor localization because the related cyber-physical infrastructures are already available in many buildings nowadays [2, 4, 15, 21]. However, WiFi-based indoor localization needs to conquer several challenges to achieve high accuracy: (1) **Stability**. Figure 1a shows a floor plan of a university building and Fig. 1b shows the Received Signal Strength Indication (RSSI) of a WiFi access point (hotspot), collected over 1000 times at location A, from which we observe that the WiFi signal strength of a hotspot is not a constant at a given location. In Fig. 1b, the RSSI value ranges from -68dBm to -89dBm; (2) **Reliability**. WiFi signal becomes more unreliable when its RSSI value gets lower. For example, it is reported in [19] that one WiFi hotspot becomes limitedly valuable when its RSSI value is below -85dBm; (3) **Discriminative Capability**. Figure 1c–e show the average signal strength of every hotspot at three different locations C, D and E within the same building. Location C is about 10 meters away from location D and location E is about 40 meters away from C and D. From Fig. 1c–e, we observe that locations C and D have similar WiFi signal patterns but they are 10 meters away from each other. (4) **Environment Change.** In practice, it is possible that some hotspots are shut down or the RSSI value of a specific hotspot is changed because of the device update. Although the previous work [2, 4] can mitigate the first three challenges by extracting more sophisticated features from the raw RSSI values, it is still hard to deal with the WiFi environment change.

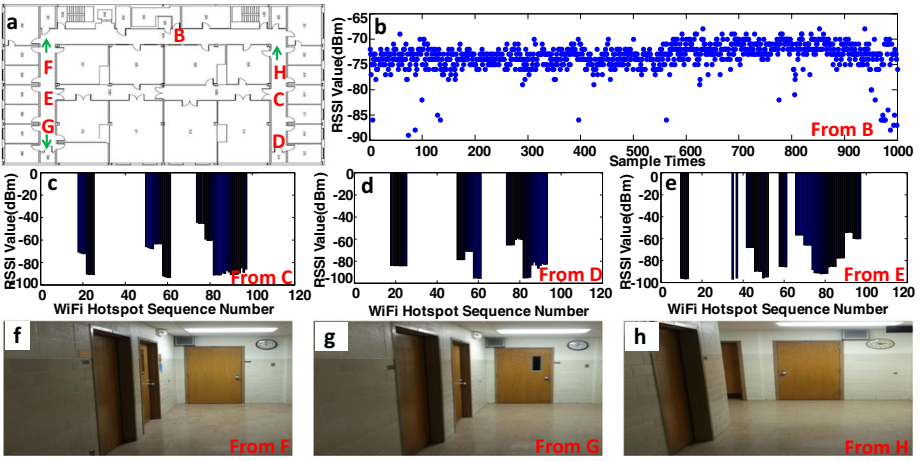


Fig. 1 Challenges for indoor localization. **a** A floor plan of a university building. **b** Received Signal Strength Indication (RSSI) of a WiFi hotspot collected over 1000 times at location B. **c**–**e** RSSI values of all WiFi hotspots at locations C, D and E, respectively. **f**–**h** Images taken at locations F, G and H, respectively, with their orientations specified in (**a**)

Inertial Measure Unit (IMU) is another sensor that can be used in indoor localization. Previous work usually utilizes IMU to perform the step detection, speed estimation and heading direction determination. The three components can be built in the Dead Reckoning framework to obtain the user's trajectory. The trajectory can then be matched with the floor plan to infer the user locations [5, 14, 30]. This method does not need to collect reference data in the building except the floor plan, but Dead Reckoning suffers from cumulative errors, making the trajectory estimation inaccurate.

Visual signal is intuitively useful for indoor localization as people generally know where they are according to what they see. In visual-based indoor localization, an image around a user's location is taken and it is compared with a pre-recorded image database to estimate her/his location [11, 13, 17, 27, 28]. This method needs to be improved in the following aspects: (1) **Computational Cost**. In a common building, thousands of images can be recorded as references and millions of Scale Invariant Feature Transform (SIFT) feature points can be extracted from them. An efficient way to represent and compare image contents is needed for the big image data; (2) **Image Feature Description**. Simple template matching between a query image and reference image dataset will not work well for the indoor localization. Representative and discriminative features must be extracted from images for effective and reliable image matching; (3) **Image Similarity**. A building may have a unified decoration style so similar scenes exist at different positions, which is very hard for a visual-based indoor localization system to classify. Figure 1f–h show three images taken at locations F, G and H, respectively, with their orientations specified in Fig. 1a, and they are very similar to each other.

1.2 Our proposal

Despite these challenges and shortcomings of existing indoor localization algorithms, the pervasiveness of smartphones offers the opportunities to combine heterogeneous signals together and makes full use of their advantages as well as offsets their drawbacks. In this

paper, we propose a novel tree-based indoor localization algorithm in which the WiFi, IMU orientation and visual signals from smartphones are bound into a signal tree: (1) WiFi is used for coarse positioning. The problems of instability, unreliability and low discriminative capacity of WiFi signals is mitigated since WiFi is only used for the coarse localization instead of fine localization. As we cluster WiFi fingerprints into several groups, the impact of WiFi environment change is also mitigated; (2) IMU is not used to estimate the trajectory, but to obtain the orientation towards which the user takes photos. The orientation information helps rule out impossible references in the image database, decreasing the computational cost and increasing the localization accuracy; (3) In the fine localization level, an image taken by the user is compared with the reference image database to search the most confident match. The problem of image similarity is alleviated in our system. For example, although Fig. 1f–h are similar to each other, they are distinguishable according to orientation (e.g., Fig. 1f and g are taken in different orientations) and WiFi fingerprints (e.g., Fig. 1h is far away from Fig. 1f and g, so they may have largely different WiFi fingerprints).

The indoor localization algorithm proposed in this paper is closely related to [1, 3, 23, 33]. Zhang et al. [33] and [23] utilize WiFi signals to rectify the trajectory obtained by IMU sensor. Chen et al. [3] combines Radio Frequency and WiFi to improve the localization accuracy. Azizyan et al. [1] combines more signals (i.e., WiFi, sound, motion, color) to build the localization algorithm. Unlike the previous work which just concatenates several sensors together to improve the localization accuracy, our algorithm hierarchically builds WiFi, IMU orientation and visual signals into a tree, which can not only improve the localization accuracy, but also improve the efficiency and decrease the time used to build the reference database.

In the rest of this paper, we firstly describe how to build the reference signal tree using the WiFi, IMU orientation and image data. Secondly, detailed searching strategies for online localization are introduced. Then, experimental results are presented with thorough comparisons and evaluation.

2 Algorithm overview

The proposed indoor localization algorithm consists of two stages: building the signal tree and online localization (Fig. 2).

Building the signal tree (Fig. 2a) WiFi signals are collected in a building, tagged with hotspots' Received Signal Strength Indication (RSSI) and the positions where the signals

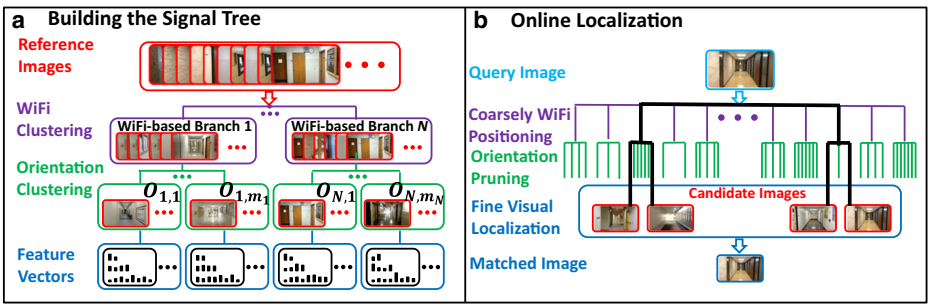


Fig. 2 Overview of the propose indoor localization algorithm

are collected. Reference images are densely captured in a building and labeled with their orientation and location information. Essentially, the construction of a signal tree is the process of clustering and describing reference images with the aid of WiFi and orientation signals. Locations are described by WiFi fingerprints and then all WiFi fingerprints are clustered into branches. All reference images are partitioned into the WiFi branches based on their spatial distance to WiFi fingerprints' positions (purple part in Fig. 2a). Then, images in the same WiFi branch are further classified according to their orientation similarity (green part in Fig. 2a). Images in one leaf node share the same WiFi and orientation labels. Given a leaf node, each image is described by multiple level descriptors (blue part in Fig. 2a).

Online localization (Fig. 2b) When a user takes a photo to localize herself/himself, WiFi and orientation signals are recorded automatically and synchronously. The signal tree is then searched to find the best matched reference image that indicates the user's location. The query WiFi fingerprint coarsely determines which WiFi branches the matched image belongs to. Orientation information further rules out impossible reference images. Then, every searched leaf node gives a candidate image as the best match to the query image. Finally, all these candidate images are compared to decide the final matched image. The matched image's tagged position indicates the user's location.

Our proposed tree-based indoor localization algorithm not only mitigates the aforementioned challenges, but also benefits as follows. (1). It makes full use of the existing infrastructures in a building, such as the WiFi hotspots equipped in libraries and hospitals; (2). It does not require too much work to users. When the user queries her/his indoor position, all needs to be done are taking a picture. The WiFi and orientation signals are automatically recoded by our software in the smartphone. (3). In the online localization, WiFi and orientation can not only offer more context information to refine the matched location, but also rule out impossible reference images, decreasing the computational cost and increasing the localization accuracy. Thus, the proposed method can be applied to scenarios where buildings are equipped with WiFi hotspots and users carry common smartphones which are popular nowadays.

3 Building the WiFi branches

In our signal tree, WiFi signals are sparsely collected in a building. A location is described by the WiFi fingerprint, which is a vector with each dimension equaling to the processed RSSI of a certain hotspot. To better describe the WiFi environment of a building, all fingerprints are clustered into groups. In this section, we firstly introduce how to generate WiFi fingerprints from WiFi signals and then present the clustering algorithm to cluster WiFi fingerprints into WiFi branches in the signal tree.

3.1 WiFi fingerprint

Due to the variation of WiFi signals, we collect the RSSI of each hotspot multiple times at any reference location in a building and compute its mean RSSI:

$$\overline{f}_{i,j} = mean_s \ f_{i,j,s} \tag{1}$$

where $f_{i,j,s}$ is the s_{th} sample of the raw RSSI values of WiFi hotspot j at location i.

It is reported in [19] that WiFi signal gets less reliable when its RSSI value is lower, so we normalize the raw RSSI values by an exponential distribution (i.e., the WiFi signal is less valuable for the WiFi fingerprint generation or comparison if its RSSI value is lower):

$$f^*_{i,j} = \lambda exp\left[\lambda \frac{\overline{f}_{i,j} - f_{min}}{f_{max} - f_{min}}\right] \tag{2}$$

where f_{max} and f_{min} are the maximal and minimal RSSI value in all $\overline{f}_{i,j}$:

$$f_{max} = max_{i,j}\ \overline{f}_{i,j} \tag{3}$$

$$f_{min} = min_{i,j}\ \overline{f}_{i,j} \tag{4}$$

λ is the rate parameter

$$\lambda = \frac{f_{max} - f_{min}}{\overline{f} - f_{min}} \tag{5}$$

where

$$\overline{f} = mean_{i,j}\ \overline{f}_{i,j}. \tag{6}$$

Therefore, the WiFi fingerprint at location i is defined by a feature vector

$$\mathbf{f}_i = \left[f^*_{i,1},\ \ldots\ ,\ f^*_{i,j},\ \ldots,\ f^*_{i,N_h}\right] \tag{7}$$

where N_h denotes the number of WiFi *h*otspots in a building.

Furthermore, we consider the discriminative capability of each WiFi hotspot for the WiFi fingerprint clustering. A WiFi hotspot is less discriminative if many reference locations receive similar RSSI values of this hotspot. On the contrary, a WiFI hotspot is discriminative if different locations receive largely different RSSI values of this hotspot. We define the discriminative capability of WiFi hotspot j by its entropy:

$$w_j = -\sum_r [p_{r,j} \times \log(p_{r,j} + \varepsilon)] \tag{8}$$

where ε is a small value to avoid computing the log of zero (e.g., $\varepsilon = 10^{-6}$). $p_{r,j}$ is the possibility of a location collecting hotspot j's RSSI value that is equal to r (e.g., $r \in [-135dBm, 0dBm]$):

$$p_{r,j} = \frac{\sum_i \delta(\overline{f}_{i,j} = r)}{N_P} \tag{9}$$

where $\delta()$ is the Kronecker delta function. The numerator in (9) is the number of $\overline{f}_{i,j}$ that equals to r for WiFi hotspot j and N_P is the total number of reference *p*ositions that have WiFi signals collected in the WiFi dataset.

If a WiFi hotspot's RSSI values are constant at all locations, the entropy is high (i.e., uncertainty of identifying the location based on this hotspot is high). Contrarily, if a WiFi hotspot's RSSI values are diverse at different locations, the entropy is low (i.e., we are more certain to identify the location based on this hotspot). Hence we define a diagonal weighting matrix for the WiFi clustering to be described in the next subsection:

$$\mathbf{W} = diag(e^{-w_1},\ \ldots,\ e^{-w_j}\ \ \ldots\ e^{-w_{N_h}}). \tag{10}$$

3.2 WiFi clustering

The physical infrastructures and electronic equipment inside a building make the WiFi environment complicated such that the distance between two reference locations is not adequate to describe the similarity of their WiFi fingerprints (i.e., adjacent positions may have

a large discrepancy in WiFi fingerprints while positions with similar fingerprints may be relatively far away from each other). To better partition the WiFi environment in a building and to mitigate the problem that WiFi environment in a building may be changed due to closed or updated hotspots, we cluster WiFi fingerprints into groups based on their **WiFi fingerprint similarity** and **spatial distance**. The clustering procedure is divided into two steps (Fig. 3): Bottom-Up clustering by WiFi fingerprint similarity and Top-Down cutoff by spatial distance:

Bottom-up clustering by WiFi fingerprint similarity (Fig. 3a) Initially each individual WiFi fingerprint is considered as one cluster. Similarities between every pair of clusters are calculated based on the similarity kernel and two clusters which are the most similar are then merged into one bigger cluster in the next higher level. This mergence operation is performed iteratively and the complete WiFi hierarchical tree is built when all WiFi

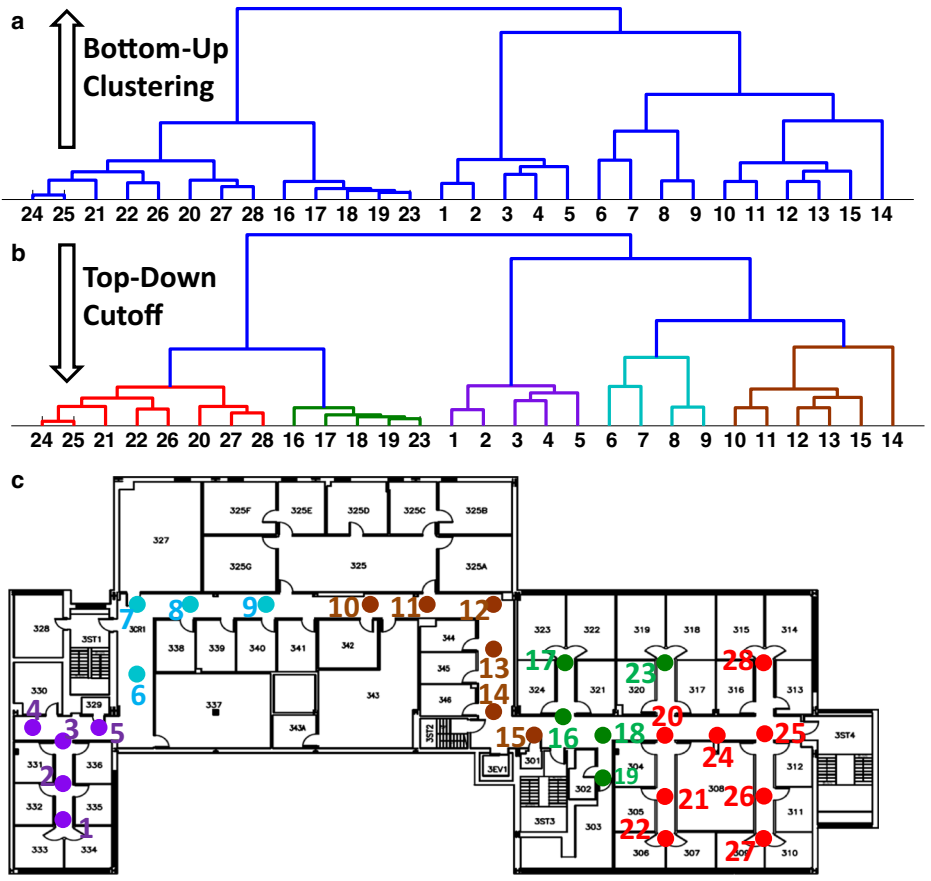


Fig. 3 WiFi clustering. **a** Button-Up WiFi Clustering Dendrogram. The number in the leaves are indices of WiFi fingerprint; **b** Top-Down Cutoff Dendrogram. Leaves sharing the same color belong to the same WiFi cluster; **c** The floor plan where the WiFi fingerprints are collected. Dots indicate where the WiFi signals are collected and surrounding numbers are the corresponding indices of WiFi fingerprints. Dots sharing the same color belong to the same WiFi cluster corresponding to (**b**)

fingerprints are grouped into the single largest cluster. The similarity kernel of two clusters is defined by the Ward's method [31]

$$S(A, B) = \sum_{k \in A \cup B} ||\mathbf{f}_k - \bar{\mathbf{f}}_{A \cup B}||_E - \sum_{k \in A} ||\mathbf{f}_k - \bar{\mathbf{f}}_A||_E - \sum_{k \in B} ||\mathbf{f}_k - \bar{\mathbf{f}}_B||_E \quad (11)$$

where $\mathbf{f}_k$ denotes a WiFi fingerprint. $\bar{\mathbf{f}}_A$, $\bar{\mathbf{f}}_B$ and $\bar{\mathbf{f}}_{A \cup B}$ are the centroids of cluster A, B and $A \cup B$, respectively. $\| \cdot \|_E$ is the weighted *E*uclidean distance defined as

$$||\mathbf{f}_i - \mathbf{f}_j||_E = (\mathbf{f}_i - \mathbf{f}_j)^T \mathbf{W} (\mathbf{f}_i - \mathbf{f}_j) \quad (12)$$

where $\mathbf{f}_i$ and $\mathbf{f}_j$ are any WiFi fingerprints. The weight matrix $\mathbf{W}$ is from (10).

Top-down cutoff by spatial distance (Fig. 3b) The WiFi hierarchical tree in Fig. 3a shows a multi-branch hierarchy rather than a set of WiFi clusters. In this step, we partition the tree into several groups based on WiFi fingerprints' spatial distances. As shown in Fig. 3b, from the top to down of the WiFi hierarchy, every node is checked if the maximal value of the spatial distance between all pairs of WiFi fingerprints belonging to this node is less than a predefined threshold d_{thr}^{WiFi} (we set it as 20 meters here because it is the distance limit that WiFi can coarsely distinguish two positions). When the maximal value is less than d_{thr}^{WiFi}, the WiFi fingerprints belonging to this node will be considered to be within the same group.

Note that the number of WiFi clusters is defined automatically by our algorithm instead of being preset by human. The Bottom-Up clustering and Top-Down cutoff consider the WiFi fingerprint similarity and spatial distance, respectively, thus the clustering result reveals the actual WiFi environment of a building well. Figure 3c shows the final clustering results of WiFi fingerprints in one university building. For example, location 19 is further from 23 than from 20, but the WiFi fingerprint at location 19 is more similar with 23 than with 20, therefore, location 19 is clustered with 23. Location 18 and 20 are near to each other, but they have relatively different WiFi fingerprints so they are grouped in different clusters. In addition, the WiFi environment is represented as groups instead of individual fingerprints, so the tolerance to WiFi environment change becomes higher.

4 Building the orientation branches

Magnetometer (measure the earth's magnetic field) and accelerometer (measure the tri-axis acceleration) are embedded in smartphones hence a phone's acceleration and magnetic field values can be recorded when a user takes photos with the phone. In this section, we firstly introduce the transformation method between a phone's coordinate system and the world coordinate system. Then we introduce how to estimate a phone's orientation on the floor plan when its user takes photos. Finally, we present how to cluster estimated orientations into the orientation branches of a signal tree.

4.1 Transformation between the phone and the world

A *p*hone's coordinate system is shown in Fig. 4a. When a smartphone is held upright in front of a user, x_p axis points right, y_p axis points up and z_p points towards outside the screen.

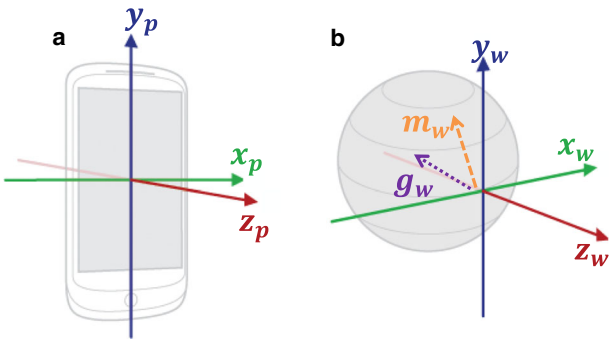


Fig. 4 The coordinate system of a phone (**a**) and the world (**b**) http://www.androidcommunitydocs.com/guide/topics/sensors/index.html. g_w and m_w are the gravity and geomagnetic north pole directions in the world coordinates, respectively

The *w*orld coordinate system is shown in Fig. 4b where y_w axis is tangential to the ground and points to the north. z_w points towards the sky and x_w is just the cross product (right-hand rule) of y_w and z_w. Gravity direction (purple dotted line in Fig. 4b) is parallel to $-z_w$ axis, while geomagnetic north pole direction (croci dashed line in Fig. 4b) has components in both $-z_w$ axis and y_w axis. Nevertheless, the cross product of gravity direction (g_w) and geomagnetic north pole direction (m_w) points to x_w axis. The subscripts p and w denote the phone and world coordinates, respectively.

We describe the coordinate transformation method in the format of quaternion, a four-dimensional normalized vector http://en.wikipedia.org/wiki/Quaternion. We define the transformation quaternion from the phone to the world coordinate system $\mathbf{Q}_{p\to w} = [q_1\ q_2\ q_3\ q_4]$ where $\|\mathbf{Q}_{p\to w}\|_2 = 1$ and $\|\cdot\|_2$ is the 2-norm. A general quaternion in the phone's coordinate system, such as $\mathbf{v}_p = [v_1\ v_2\ v_3\ v_4]$, is transformed to the world coordinate system by

$$\mathbf{v}_w = \mathbf{Q}_{p\to w} \otimes \mathbf{v}_p \otimes \mathbf{Q}^*_{p\to w} \tag{13}$$

where $\otimes$ is the quaternion multiplication defined as

$$\begin{aligned}\mathbf{Q}_{p\to w} \otimes \mathbf{v}_p &= [q_1\ q_2\ q_3\ q_4] \otimes [v_1\ v_2\ v_3\ v_4] \\ &= \begin{pmatrix} q_1v_1 - q_2v_2 - q_3v_3 - q_4v_4 \\ q_1v_2 + q_2v_1 + q_3v_4 - q_4v_3 \\ q_1v_3 - q_2v_4 + q_3v_1 + q_4v_2 \\ q_1v_4 + q_2v_3 - q_3v_2 + q_4v_1 \end{pmatrix}^T .\end{aligned} \tag{14}$$

$\mathbf{Q}^*_{p\to w}$ is the quaternion conjugate of $\mathbf{Q}_{p\to w}$ denoted as

$$\mathbf{Q}^*_{p\to w} = [q_1\ -q_2\ -q_3\ -q_4] = \mathbf{Q}_{w\to p}. \tag{15}$$

In the format of quaternion, any point in the world or phone coordinate is represented as a vector $[0\ x\ y\ z]$, where x, y and z are the values of three axes, respectively. We compute

the coordinate transformation quaternion, $\mathbf{Q}_{p\to w}$, by solving the following optimization problem:

$$\arg\min_{\mathbf{Q}_{p\to w}} \left\{ \| \left(\mathbf{Q}_{p\to w} \otimes \mathbf{a}_p \otimes \mathbf{Q}^*_{p\to w} - \mathbf{g}_w \right) \|_2^2 + \| \left(\mathbf{Q}_{p\to w} \otimes \mathbf{a}_p \otimes \mathbf{Q}^*_{p\to w} \right) \times \left(\mathbf{Q}_{p\to w} \otimes \mathbf{m}_p \otimes \mathbf{Q}^*_{p\to w} \right) - \mathbf{x}_w \|_2^2 \right\} \tag{16}$$

where $\mathbf{a}_p = [0\ a_1\ a_2\ a_3]$ is the normalized (i.e., $\|\mathbf{a}_p\|_2 = 1$) acceleration in the phone coordinate. The acceleration of a phone is measured by its accelerometer. When a user takes photos, we assume the smartphone is relatively stable thus only gravity contributes to a phone's acceleration. The acceleration $\mathbf{a}_p$ is transformed from the phone coordinate system to the world coordinate system by $\mathbf{Q}_{p\to w} \otimes \mathbf{a}_p \otimes \mathbf{Q}^*_{p\to w}$, which should be equal to $\mathbf{g}_w$. $\mathbf{g}_w = [0\ 0\ 0\ -1]$, a constant vector pointing to the gravity in the world coordinate. Therefore, the first term of (16) is the difference between transformed $\mathbf{a}_p$ and $\mathbf{g}_w$.

In the second term of (16), $\mathbf{m}_p = [0\ m_1\ m_2\ m_3]$ is the normalized magnetic field in the phone coordinate, which is measured by the phone's magnetometer. The acceleration $\mathbf{a}_p$ and magnetic field $\mathbf{m}_p$ are transformed from the phone coordinate system to the world coordinate system as $\mathbf{a}_w = \mathbf{Q}_{p\to w} \otimes \mathbf{a}_p \otimes \mathbf{Q}^*_{p\to w}$ and $\mathbf{m}_w = \mathbf{Q}_{p\to w} \otimes \mathbf{m}_p \otimes \mathbf{Q}^*_{p\to w}$, respectively. As shown in Fig. 4b, the normalized cross product ($\times$) of gravity $\mathbf{g}_w$ and $\mathbf{m}_w$ should be equal to $\mathbf{x}_w$ ($\mathbf{x}_w = [0\ 1\ 0\ 0]$, the x axe of the world coordinate system). Therefore, the second term of (16) is the difference between $\mathbf{a}_w \times \mathbf{m}_w$ and $\mathbf{x}_w$.

After some matrix computation, (16) is converted into the following quadratic optimization problem:

$$\arg\min_{\mathbf{Q}_{p\to w}} \left(4 + \mathbf{Q}_{p\to w} \mathbf{A} \mathbf{Q}^T_{p\to w} \right) \tag{17}$$

where $\mathbf{A}$ is a 4-by-4 symmetrical matrix related to $\mathbf{a}_p$ and $\mathbf{m}_p$. This minimization problem is considered as the Rayleigh Quotient and $\mathbf{Q}_{p\to w}$ is solved as the eigenvector corresponding to the smallest eigenvalue of $\mathbf{A}$.

4.2 Orientation estimation

Figure 5 shows the scenario when a user takes a photo. The yellow vector $\mathbf{c}_p = [0\ 0\ 0\ -1]$ (parallel with $-z_p$ axis in the phone's coordinate system) represents the orientation that the camera is towards. $\mathbf{c}_p$ is transformed to the world coordinate by

$$\mathbf{c}_w = \mathbf{Q}_{p\to w} \otimes \mathbf{c}_p \otimes \mathbf{Q}^*_{p\to w} \tag{18}$$

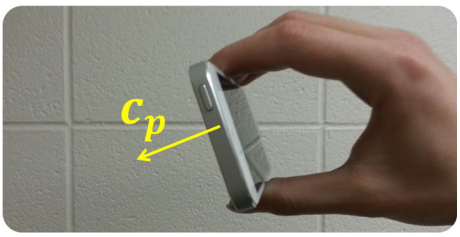


Fig. 5 The scenario when a user takes a photo for localization. $\mathbf{c}_p$ is the constant vector parallel with $-z_p$ axis in the phone's coordinate system

Denoting $\mathbf{c}_w = [0\ c_1\ c_2\ c_3]$, we project the orientation to the horizontal plane in the world coordinate system, i.e., vector $O = [c_1\ c_2]$ is the orientation on the floor plan which the photo is taken towards.

Note that when a user takes photos, the smartphone usually keeps stable, so the acceleration value is relatively accurate without the noise of movement. However, the electronic equipment inside buildings, the metal components and metal structures of smartphones all lead substantial bias to the measurement of earth's magnetic field. The good thing is that this error does not badly affect the localization algorithm considering that the consistent noise results in a constant deviation between the measured earth's magnetic field and ground truth. So different people take photos towards the same direction in the same position, their measured orientations are similar although not perfectly accurate. The similar orientation can still form a useful feature to describe the location.

4.3 Orientation clustering

The orientation information is tagged with reference images. To discuss the orientation clustering, we need to introduce the reference image collection process first. **Compared to the sparse collection on WiFi signals, we collect reference images densely in a building for precise localization by recording continuous videos and sensor information simultaneously.** For example, eight video clips were recorded in a building following the eight routes defined in Fig. 6a. Each frame in the videos is tagged with its corresponding orientation computed by the method in the previous subsection. Each video clip is recorded following the same direction, therefore the computed orientations of all frames in a video are similar, naturally forming a cluster of orientations.

Figure 6b shows the distributions of eight orientation clusters corresponding to the eight routes in Fig. 6a. For example, the red and black distribution curves are largely overlapped and they point to the same orientation in Fig. 6a. The orientation distribution of each video clip is not a constant impulse distribution due to the environmental impacts on magnetic and inertial sensors. We model the orientation distribution of video clip q by a Gaussian distribution $N(\mu_q, \delta_q)$. Suppose the entire floor plan in Fig. 6a is in one WiFi cluster, we can further merge largely overlapped orientation clusters into a bigger cluster. The similarity of two distributions q_1 and q_2 is defined as:

$$S_{q_1,q_2} = \frac{\delta_{q_1}^2 + \delta_{q_2}^2}{|\mu_{q_1} - \mu_{q_2}|} \tag{19}$$

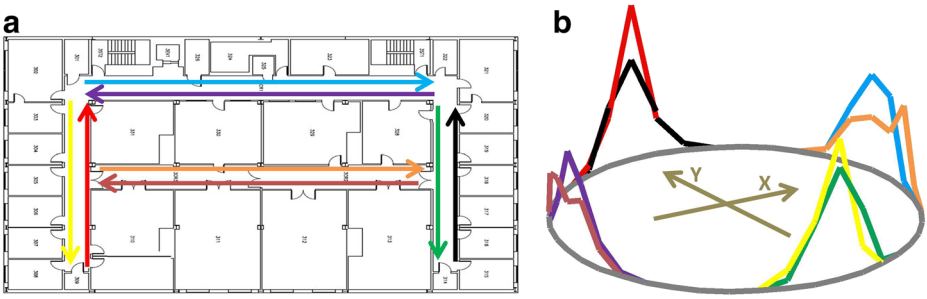


Fig. 6 Orientation clustering. **a** Floor plan of a building with eight routes to record videos and their corresponding sensor information. **b** Orientation distributions calculated by the data collected according to (**a**)

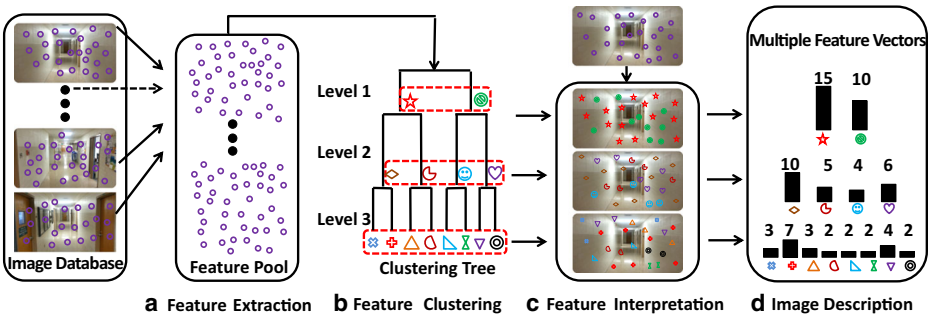


Fig. 7 Flow chart of the Multiple Level Image Descriptions (MLID) method

If the centroid of two distributions are close to each other and their inter-distribution variances are small, then they can be merged into a bigger orientation cluster. In Fig. 6b, the eight orientation distributions can be clustered into four clusters corresponding to north, west, south and east directions. Each of the four orientation clusters is one orientation subbranch within the same WiFi branch.

Without loss of generality, video clips with more orientations can be recorded according to how dense the image database is required to be. The video clips collected towards the same orientation are not necessary to be merged to a cluster if inertial sensors are badly influenced by noise.

5 Building the image leaf nodes

In this paper, we propose a Multiple Level Image Description (MLID) method to describe images in leaf nodes of the signal tree (Fig. 2). MLID is based on Term Frequency Inverse Document Frequency (TF-IDF) [24], but we improve it in three-folds: (1) Dense SIFT key-points are extracted in the low texture areas. (2) Divisive hierarchical clustering is adopted rather than K-means. (3) Each image is described as multiple vectors, thus both global

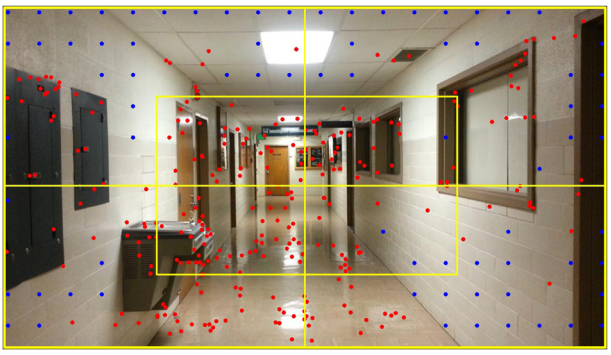


Fig. 8 SIFT points in a user-taken image. The red points are the salient SIFT points and the blue points are the dense SIFT points. The image is equally divided into five subimages

and local information of an image is recorded. As shown in Fig. 7, MLID consists of four steps:

Feature extraction SIFT is ideal to describe local features because it is robust to local affine distortion [26], which means even though the query image has scale or rotation change compared to the images in the dateset, SIFT keypoints remain stable between the corresponding images. Figure 8 shows an example of feature extraction. First, salient SIFT keypoints (red points in Fig. 8) are extracted from an image. However, salient SIFT keypoint extraction ignores the low texture areas such as some parts of the ceiling and walls. Then, dense SIFT keypoints (blue points in Fig. 8) are extracted in the image. Dense SIFT keypoints are uniformly sampled in the area without salient SIFT keypoints. The SIFT features extracted from both salient and dense SIFT keypoints jointly describe the content of an image. All the SIFT features extracted from every reference image in a leaf node of the signal tree are collected into a large feature pool (represented as purple circles in Fig. 7a).

Feature clustering The number of keypoints in the feature pool depends on how large and complex the building is, which ranges between hundreds of millions and hundreds of thousands. Thus it is hard to estimate the proper number of groups. In addition, clustering hundreds of millions of keypoints into millions of groups in one time is time-consuming. In this paper, a top-down clustering method is applied to partition the SIFT keypoints in the feature pool. SIFT features are firstly clustered into t groups ($t = 2$ in Fig. 7b) at *L*evel 1 ($L = 1$) based on 2-norm of feature vector distances. Then each cluster in the first level is clustered into t groups, so that in the second level ($L = 2$), there are t^2 branches. The process is performed repeatedly until every leaf of the feature clustering tree has less than r SIFT feature descriptors (r is set to 100. When there is a small number of features descriptors in a group, no further clustering is needed). The symbol around each node in Fig. 7b represents *the mean of SIFT feature vectors in that subtree* (called visual word) and the visual words at each level forms the visual codebook for that level. In Fig. 7b, the symbols in each dotted rectangle belong to one visual codebook.

Feature interpretation Based on the visual codebooks at different levels, we can interpret the SIFT features in an image into visual words in a corse-to-fine manner. A SIFT feature is interpreted as the visual word of a visual codebook which is the closest to the SIFT feature. The distance between a SIFT feature and visual word is based on the 2-norm of feature vector distance. For example, in Fig. 7c, at level 1, all feature descriptors in one image is interpreted into 2 visual words: 15 SIFT descriptors are close to visual word 1 (red star) and 10 SIFT descriptors are close to visual word 2 (green circle). The interpreted visual words at level 1 are finely interpreted at following levels.

Image description At level L, the number of visual words in the codebook is t^L, and the histogram of visual words at level L is used as the feature description at level L. For example, in Fig. 7d, at $L = 1$, the codebook has 2 elements, so does the feature vector. Because 15 SIFT descriptors belong to visual word 1 (red star) and 10 SIFT descriptors belong to visual word 2 (green circle), the description vector is [15, 10] at level 1 (or [0.6, 0.4] after normalization). The feature descriptors are finely computed in the subsequent levels according to more and more detailed visual vocabulary books.

Spatial information is also considered when formulating the feature description of an image. As the yellow lines in Fig. 8 illustrate, the image is firstly equally divided into four subimages and the fifth subimage is in the center of the image with the same size of other four subimages. Multi-level feature vectors are calculated based on individual subimages and then they are concatenated to form long vectors to describe the whole image. So in level L, there are actually $5t^L$ dimensions in the feature vector.

The proposed MLID algorithm keeps both global and local information of images. At the top level, SIFT descriptors are coarsely clustered and the dimension of feature vector is low, so the global information of the image is reflected. As the feature descriptors are finely clustered, the dimension of feature vector gets larger and more detailed information is recorded. Note that, compared with K-means, there is no need for our algorithm to predefine how many groups we should cluster the SIFT descriptors, which is another advantage of the MLID method to handle different unknown scenes.

In a short summary of this section, reference images in the dataset are partitioned based on their surrounding WiFi environment at the first level of signal tree and then they are further clustered according to their tagged orientation information at the second level. In the third level, the Multiple Level Image Description is generated for every image within a leaf node of the signal tree. Thus, the surrounding sensor environment and image attributes of a position are merged together in the hierarchical signal tree to describe that location.

6 Online localization

When a user takes a photo to localize herself/himself, WiFi and sensor signals are recorded synchronously. This section presents the search strategy to find the best matched reference image to identify a user's location. Online localization consists of three stages: coarsely WiFi positioning, orientation pruning and fine visual localization.

6.1 WiFi-based coarse positioning

Let $\mathbf{f}_0$ be the WiFi fingerprint submitted by the user and can be computed by (7). Assume there are N_{WiFi} WiFi clusters in the signal tree. The centroid of WiFi clusters are denoted as $\mathbf{f}_n (n = 1...N_{WiFi})$. The distance between $\mathbf{f}_0$ and any WiFi cluster $\mathbf{f}_n$ is computed by the weighted Euclidean distance, denoted as $d_{0,n}$.

$$d_{0,n} = (\mathbf{f}_0 - \mathbf{f}_n)^T \mathbf{W} (\mathbf{f}_0 - \mathbf{f}_n). \tag{20}$$

Only the top h WiFi clusters with the smallest distance will be searched in the next level, other WiFi clusters as well as their subbranches are skipped over. In the experiments, h is set to 2 which works well in our campus buildings. If the WiFi environment is complex, h can be larger such that more WiFi branches can be searched. In the following steps, branches are searched independently.

6.2 Orientation pruning

Several hundred of orientation samples can be collected instantly when a user is taking a photo. The query orientations O_0 can be modeled as a Gaussian distribution $N(\mu_0, \sigma_0)$. The similarity between O_0 and any orientation cluster can be computed by (19). Top h orientation clusters with the smallest similarity to O_0 will be searched in the next level,

other subbranches are skipped. As shown in Fig. 2b, the black branches indicate the search routes. Only parts of the leaf nodes need be searched, greatly increasing the efficiency.

Algorithm 1 Search Algorithm in a Leaf Node

Notations:

- B: the totally number of reference images in a leaf node
- L: the number of clustering levels in a leaf node

Input:

- Multiple feature vectors of query image: $V_{0,l}(l = 1...L)$;
- Multiple feature vectors of reference images in a leaf node: $V_{b,l}(b = 1...B, l = 1...L)$;
- codebooks: $M_l(l = 1...L)$;
- A predefined threshold d_{thr}. It is set to 3 meters in this system;
- Comparison Pool (CP): all reference images in a leaf node;

Iteration:

for $l = 1 : L$ **do**

- Compute the similarity between query image and images in CP: $S^l_{0,b} = \frac{V_{0,l} \cdot V_{b,l}}{|V_{0,l}||V_{b,l}|}$
- Compute the average similarity $\overline{S^l_{0,b}} = \frac{\sum_{b \in CP} S^l_{0,b}}{\sum_{b \in CP}}$
- Reference images satisfying $S^l_{0,b} < \overline{S^l_{0,b}}$ are deleted from CP
- Compute the maximum of pairwise spatial distance of images in CP, denoted as d_{max}

  if $d_{max} < d_{thr}$ **then**
    return M_l and reference images with the largest $S^l_{0,b}$
    break
  end if
end for

Output:

M_l and the candidate image which is the reference images with the largest $S^l_{0,b}$ in CP

6.3 Fine visual localization

Within each searched leaf node, the most similar reference image needs to be found. Algorithm 1 shows the search strategy within a leaf node. The best reference image is searched from top to down of multiple vectors. At each level, we discard the reference images having low similarity with the query image. Thus as the level goes deeper, the number of reference images to be compared becomes less and less, which decreases the computational cost. Meanwhile, the dimension of feature vector increases as the level goes deeper, images are compared with more and more local details.

If only one leaf node is searched, the candidate image selected from that leaf node is the final matched reference image. Otherwise, every searched leaf node gives one candidate image, we need to compare which candidate image is the best match. As shown in Fig. 9, without loss of generality, only two candidate images are discussed here. A new visual codebook is built by concatenating the codebooks from the outputs of Algorithm 1. This new codebook is specialized to the two candidate images, therefore it is more discriminative than either of the single codebook. Then, feature vectors of the query image and candidate images are calculated based on the new codebooks. The candidate image that has the largest

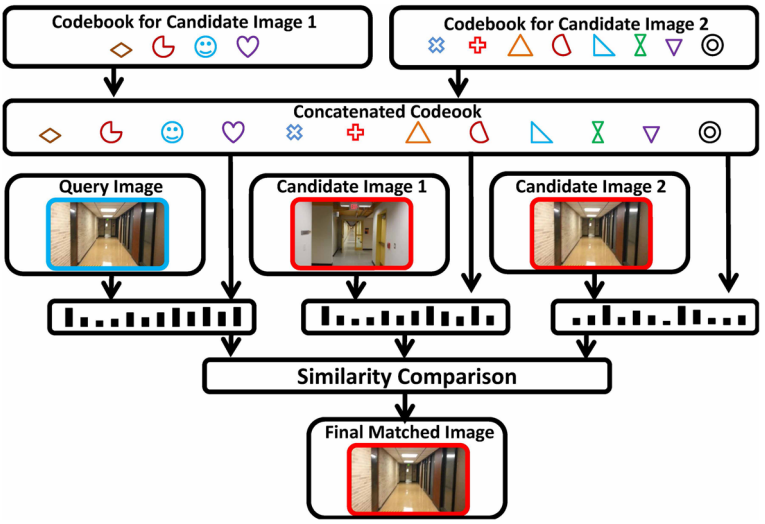


Fig. 9 Determine the final matched image from candidate images

similarity with the query image is considered as the final matched image. The matched image's labeled position is reported as the user's location.

7 Experiments

To validate the effectiveness of our indoor localization algorithm, we developed an App in the platform of Android OS to record the WiFi, inertial sensor and visual signals. Figure 10a is one screenshot of the App with a simple interface. This App is capable of collecting reference signals as well as query signals.

Figure 11 shows the floor plan of a university building based on which we illustrate how we collect reference signals and build the signal tree. The circles in Fig. 11 denote the positions where we collect WiFi signals. The WiFi signals in this building are partitioned

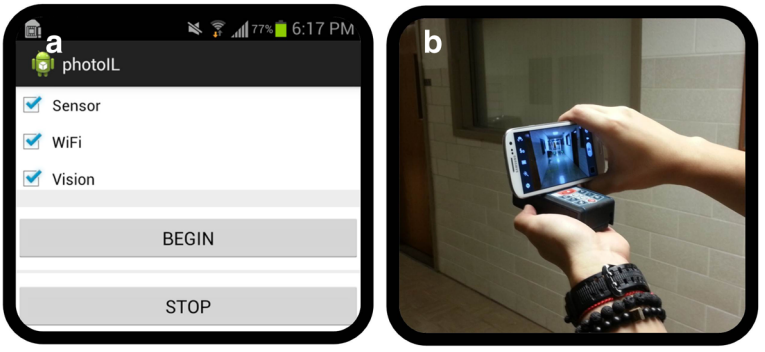


Fig. 10 **a** The data collection app. **b** A laser distance measurer is used to identify the ground truth of a user's position

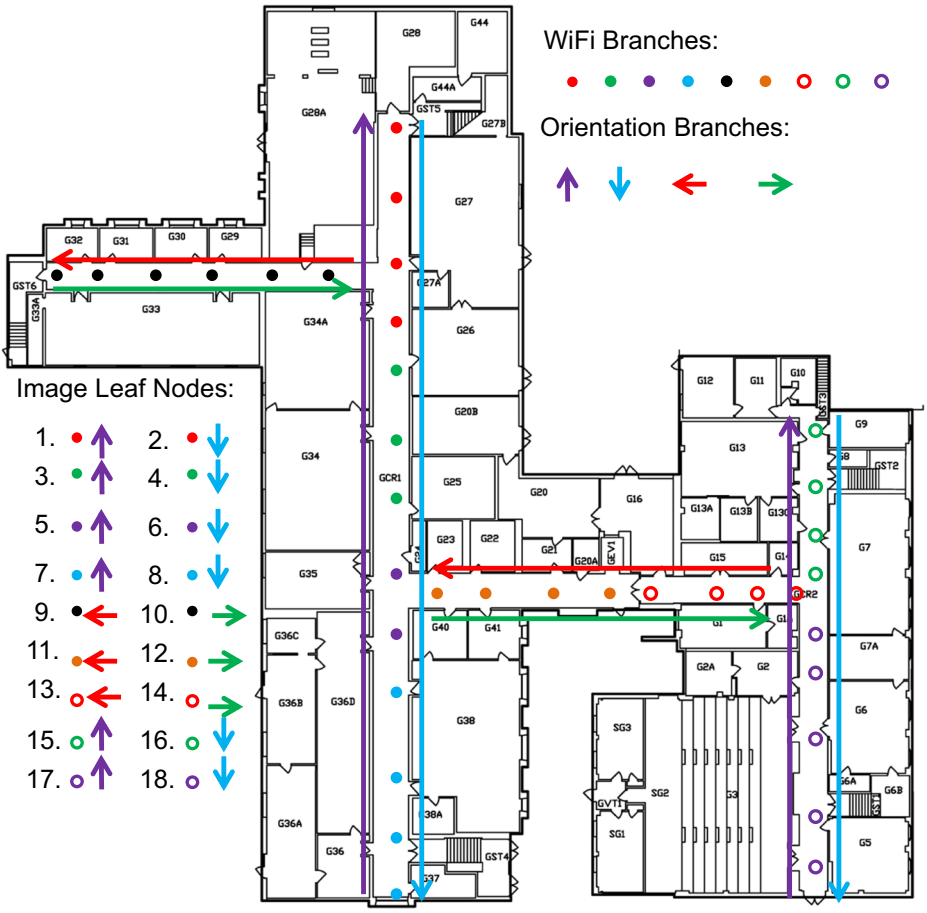


Fig. 11 Floor plan of a building with illustrations on how the signal tree is built. Circles are positions where we collect WIfi signals. Visual signals are collected from the beginning of each arrow to the end while sensor information is recorded simultaneously

into 9 branches by our WiFi clustering algorithm. Generally, WiFi signals are collected uniformly and sparsely in the available regions of a building such as hallway and public lounge. The distance of two adjacent WiFi collection position is about 5 meters. We collect visual signals in the format of videos following the routes in Fig. 11. Simultaneously, the orientation signals are recorded, which are the same as the arrow directions. The frame rate of each video is $30 f/s$. The resolution of images can be calibrated through software method. We keep walking with a constant speed when we record the videos. Thus, the position tagged to each frame can be interpolated by the positions of the start and end of each arrow. Frames in all videos forms the images dataset of the building. In this building, there are 18 image leaf nodes. As shown in Fig. 11, we denote each leaf node by the WiFi and orientation branches it belongs to.

The proposed indoor localization system is tested in 4 campus buildings whose floor plans are shown in Fig. 12. Table 1 summarizes the information of signal trees of the 4 buildings which are used in experimental evaluation.

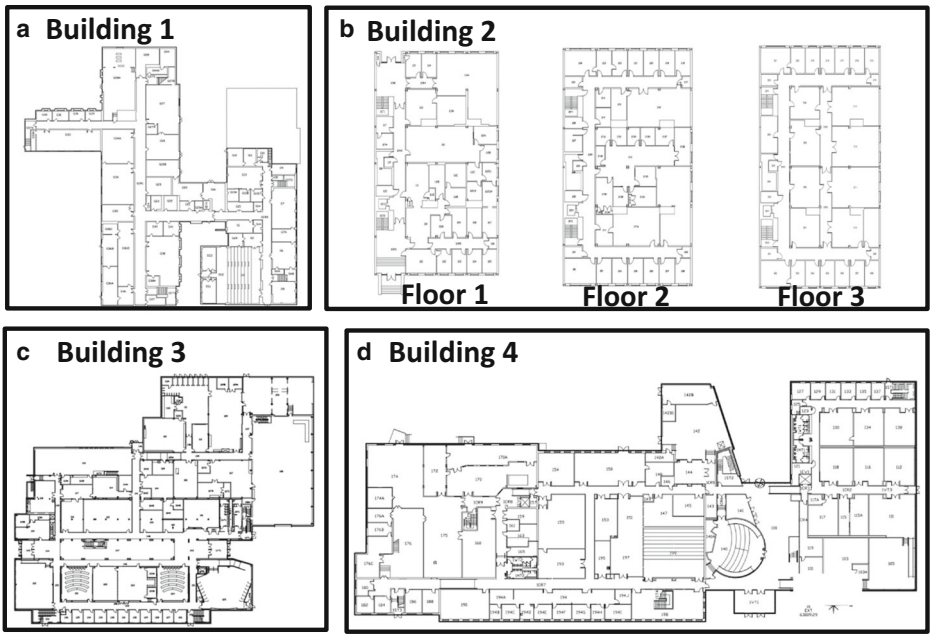


Fig. 12 Floor plans of the test buildings

Figure 13 shows some localization samples of our approach, which demonstrates the proposed localization system is robust to crowded people, scene changes and orientation shifts. Our proposed indoor localization system is compared with three other approaches: (1) Only visual signals are used in the localization. (2) Only WiFi signals are used in the localization. (3) The localization system proposed by Wang et al. [28]. Here the method by Wang et al. [28] is chosen as our benchmark because it is also based on visual information and takes Term Frequency Inverse Document Frequency(TF-IDF) as its landmark. However, Wang et al. [28] does not consider dense SIFT keypoints and multi-level feature vectors, which are parts of the contribution of this paper. The comparison is in terms of localization accuracy, localization efficiency and time used to build the reference database.

7.1 Localization accuracy

Figure 14 summarizes the localization accuracy of 4 approaches in the 4 buildings. Our system achieves the highest accuracy compared to the other 3 methods. The comparison between the approach described in [28] and MLID proposed in this paper shows that it

Table 1 Information about the Signal Trees of 4 Buildings

Building No.	NWB	NOB	NRI	NQI
1	9	4	10117	241
2	6	4	4335	283
3	11	4	19313	202
4	12	4	18825	278

NWB: Number of WiFi Branches; NOB: Number of Orientation Branches; NRI: Number of Reference Images; NQI: Number of Query Images

Fig. 13 Samples of our indoor localization. *Top row*: query images. *Bottom row*: matched reference images. **a** People occlusion. **b** Illumination changes. **c** Orientation shifts. **d** Scene slightly changes. **e** Textureless scene

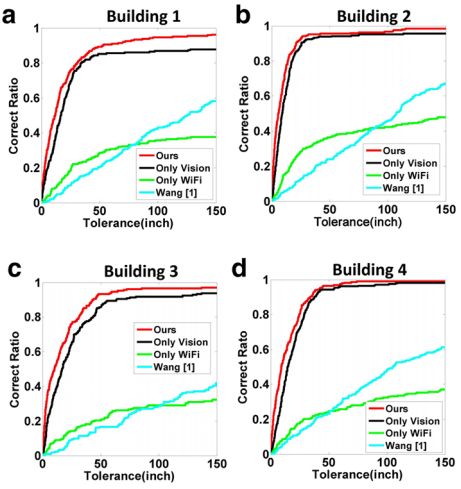


Fig. 14 Accuracy comparison. Horizontal-axis is the distance between ground truth and estimated user's position. Vertical-axis is the proportion of query signals that have the accuracy within the distance labeled in horizontal-axis

Table 2 Average time used for localization (seconds)

Building	Ours	Only image	Wang [28]	Only WiFi
1	5.77 ± 0.59	10.42 ± 0.60	5.48 ± 0.52	0.0094 ± 0.00015
2	5.63 ± 0.52	9.63 ± 0.54	5.22 ± 0.56	0.006 ± 0.00012
3	5.80 ± 0.57	10.79 ± 0.55	5.11 ± 0.59	0.0050 ± 0.00015
4	6.20 ± 0.55	11.23 ± 0.60	4.91 ± 0.51	0.0014 ± 0.00014

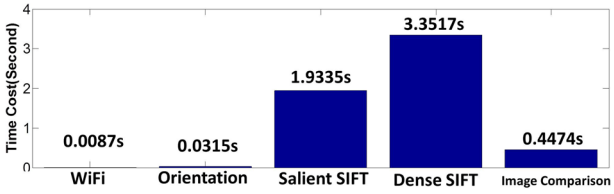


Fig. 15 Time cost of every step of our method in building 1

Table 3 Time used to build the database (hours)

Building	Ours	Only image	Wang [28]	Only WiFi
1	1.75	15.5	12	0.000866
2	2.5	23.75	22	0.001178
3	2	28	27.75	0.000948
4	2.25	27	26	0.00145

is more effective to describe images with multiple-level features descriptors, thus images' global and local information are both recoded and utilized for localization. Note that for all building datasets, 80% of query images have the accuracy within 30 inches and 95% of these have the accuracy within 100 inches, which is acceptable in most scenarios such as schools and hospitals.

7.2 Localization efficiency

Table 2 summarizes the comparison of the average time cost of online localization. During all the experiments, we notice that all query signals can be localized in less than 6.5 seconds with our method. The comparison between column 2 (Our signal tree method) and column 3 (image-only method) proves that WiFi and orientation signals are capable to rule out impossible reference images and largely speed up the online localization.

However, our current method is still slower than Wang et al. [28]. As shown in Fig. 15, we analyzed the average time cost of every step in our method and found out that computing dense SIFT keypoints which is not required in Wang's method consumes 58.06% (about 3.35s) of the total time while searching the signal tree only takes up 7.75% (about 0.45s) in our method. Finding the SIFT keypoints can be further speeded up with GPU parallel computing. We leave this as our future work.

7.3 Time used for building the reference database

Table 3 summarizes the time cost of the 4 approaches to build the reference database. Except WiFi-only method, The proposed signal tree takes the least time to build the database (about one-tenth of the time cost of the image-only method). Note that the advantage of the proposed method will be much greater in a skyscraper. It is very time-consuming to deal with all reference images from a tall building as a whole. However, no matter how big the building is, WiFi and orientation signals pre-cluster reference images into several leaf nodes, thus a complex problem is divided and conquered by small problems.

8 Conclusion and discussion

In this paper, we propose a novel signal-tree based indoor localization algorithm combining WiFi, inertial and visual signals. From the experimental results, our method takes more time to build the reference database compared with the WiFi-only method and our method takes more time for online localization compared with the WiFi-only and Wang's method [28], but the accuracy of our method is much better than the other methods. Considering the evaluation metrics together, our proposed method is competitive and effective in the indoor environment. The biggest contribution of the proposed algorithm is that WiFi and orientation signals simplify the database building process and provide more contextual

information not only to speed up the image query, but also to improve the localization accuracy. In addition, the multiple level description records more comprehensive information about an image. They all make an accurate and efficient indoor localization possible. Other advantages of the proposed algorithm are summarized as follows.

For a fingerprint base algorithm, it is usually painful to collect reference data, especially for the high accuracy requirement. In this paper, because WiFi signal is only used for coarse position, we just uniformly and sparsely collect WiFi fingerprints in a building (about every 5 meters a reference location). In addition, the previous work [11, 13, 27] usually takes thousands of reference images in building and recodes each image's location one by one. However, we collect reference images in the format of videos. Every frame of the videos forms a reference image tagged by WiFi and IMU sensor signals automatically, which largely speed up the collection.

In online localization, the user just need to take a photo with his/her smartphone, the WiFi and orientation information will be automatically recorded. Thus, the workload of the user will not be increased compared with traditional image-based indoor localization system.

One challenge of indoor localization algorithm using the WiFi signal is the WiFi environment change, which may be caused by the shut-down or updating of hotspots. The proposed algorithm deals with this problems in two ways. As discussed in Section 3.2, WiFi fingerprints are clustered into groups, thus the tolerance to WiFi environment change is higher compared to that we treat each WiFi fingerprint individually. In the condition that WiFi environment is largely changed, we can increase the number of search branch h described in Section 6.1 to allow more branches to be searched in the signal tree. In the extreme case, h can be defined as the number of WiFi branches, which means all WiFi branches will be searched. We have tested our algorithm in this extreme condition in building 1 and the average localization errors is only slightly increased from 56 inches to 59 inches.

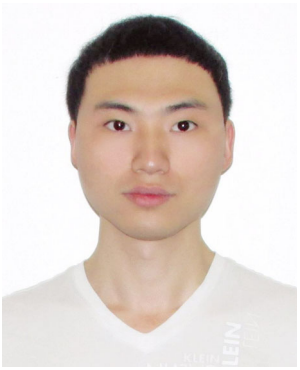
In the future work, we plan to set up an indoor navigation system based on the indoor localization algorithm. Combined with the GPS based navigation outdoors, a persistent navigation system can be built, which is an important part of smart cities.

Acknowledgements This work was supported by NSF grant CNS-1205695 on social intelligent computing and NSF grant CMMI-1646162 on cyber-physical systems, and Intelligent Systems Center at Missouri University of Science and Technology. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

1. Azizyan M, Constandache I, Roy Choudhury R (2009) Surroundsense: mobile phone localization via ambience fingerprinting. In: International conference on mobile computing and networking. ACM, pp 261–272
2. Biswas J, Veloso M (2010) Wifi localization and navigation for autonomous indoor mobile robots. In: International conference on robotics and automation. IEEE, pp 4379–4384
3. Chen Y, Lymberopoulos D, Liu J, Priyantha B (2012) Fm-based indoor localization. In: International conference on mobile systems, applications, and services. ACM, pp 169–182
4. Chintalapudi K, Padmanabha Iyer A, Padmanabhan VN (2010) Indoor localization without the pain. In: Proceedings of the 16th annual international conference on mobile computing and networking. ACM, pp 173–184
5. Constandache I, Choudhury RR, Rhee I (2010) Towards mobile phone localization without war-driving. In: Infocom. IEEE, pp 1–9
6. Corporation G (2001) About gps, in http://www.garmin.com/aboutGPS/

7. Deak G, Curran K, Condell J (2012) A survey of active and passive indoor localisation systems. Comput Commun 35(16):1939–1954
8. Esteves JS, Carvalho A, Couto C (2003) Generalized geometric triangulation algorithm for mobile robot absolute self-localization. In: International symposium on industrial electronics. IEEE, vol 1, pp 346–351
9. Fallah N, Apostolopoulos I, Bekris K, Folmer E (2013) Indoor human navigation systems: a survey. Interact Comput 25(1):21–33
10. Granados-Cruz M, Pomarico-Franquiz J, Shmaliy YS, Morales-Mendoza LJ (2014) Triangulation-based indoor robot localization using extended fir/kalman filtering. In: International conference on electrical engineering computing science and automatic control. IEEE, pp 1–5
11. Jang G, Lee S, Kweon I (2002) Color landmark based self-localization for indoor mobile robots. In: International conference on robotics and automation. IEEE, vol 1, pp 1037–1042
12. Jiang Y, Pan X, Li K, Lv Q, Dick RP, Hannigan M, Shang L (2012) Ariel: Automatic wi-fi based room fingerprinting for indoor localization. In: Conference on ubiquitous computing. ACM, pp 441–450
13. Kim J, Jun H (2008) Vision-based location positioning using augmented reality for indoor navigation. IEEE Trans Consum Electron 54(3):954–962
14. Koide S, Kato M (2005) 3-d human navigation system considering various transition preferences. In: International conference on systems, man and cybernetics. IEEE, vol 1, pp 859–864
15. Koweerawong C, Wipusitwarakun K, Kaemarungsi K (2013) Indoor localization improvement via adaptive rss fingerprinting database. In: International Conference on Information Networking. IEEE, pp 412–416
16. Liu H, Darabi H, Banerjee P, Liu J (2007) Survey of wireless indoor positioning techniques and systems. IEEE Trans Syst Man Cybern Part C Appl Rev 37(6):1067–1080
17. Liu H, Yu X, Yu H (2009) Combining color histogram and gradient orientation histogram for vision based global localization. In: International conference on systems, man and cybernetics. IEEE, pp 4043–4047
18. Liu RP, Hedley M, Yang X (2013) Wlan location service with txop. IEEE Trans Comput 62(3):589–598
19. Martin E, Vinyals O, Friedland G, Bajcsy R (2010) Precise indoor localization using smart phones. In: International conference on multimedia. ACM, pp 787–790
20. Moghtadaiee V, Dempster AG, Lim S (2011) Indoor localization using fm radio signals: A fingerprinting approach. In: International conference on indoor positioning and indoor navigation. IEEE, pp 1–7
21. Park J-g, Charrow B, Curtis D, Battat J, Minkov E, Hicks J, Teller S, Ledlie J (2010) Growing an organic indoor location system. In: International conference on mobile systems, applications, and services. ACM, pp 271–284
22. Pomárico-Franquiz J, Khan SH, Shmaliy YS (2014) Combined extended fir/kalman filtering for indoor robot localization via triangulation. Measurement 50:236–243
23. Rai A, Chintalapudi KK, Padmanabhan VN, Sen R (2012) Zee: zero-effort crowdsourcing for indoor localization. In: International conference on mobile computing and networking. ACM, pp 293–304
24. Sivic J, Zisserman A (2003) Video google: A text retrieval approach to object matching in videos. In: International conference on computer vision. IEEE, pp 1470–1477
25. So J, Lee J-Y, Yoon C-H, Park H (2013) An improved location estimation method for wifi fingerprint-based indoor localization. Inter J Softw Eng Its Appl 7(3):77–86
26. Tuytelaars T, Mikolajczyk K et al (2008) Local invariant feature detectors: a survey. Found Trends® Comput Graph Vis 3(3):177–280
27. Wang J, Cipolla R, Zha H (2005) Vision-based global localization using a visual vocabulary. In: International conference on robotics and automation. IEEE, pp 4230–4235
28. Wang J, Zha H, Cipolla R (2006) Coarse-to-fine vision-based localization by indexing scale-invariant features. IEEE Trans Syst Man Cybern B Cybern 36(2):413–422
29. Wang Y, Yang X, Zhao Y, Liu Y, Cuthbert L (2013) Bluetooth positioning using rssi and triangulation methods. In: Consumer communications and networking conference. IEEE, pp 837–842
30. Wu H, Marshall A, Yu W (2007) Path planning and following algorithms in an indoor navigation model for visually impaired. In: International conference on internet monitoring and protection. IEEE, pp 38–38
31. Xu R, Wunsch D (2008) Clustering, vol 10, Wiley
32. Yang Z, Wu C, Liu Y (2012) Locating in fingerprint space: wireless indoor localization with little human intervention. In: International conference on mobile computing and networking. ACM, pp 269–280
33. Zhang S, Xiong Y, Ma J, Song Z, Wang W (2011) Indoor location based on independent sensors and wifi. In: International conference on computer science and network technology. IEEE, vol 4, pp 2640–2643



Wenchao Jiang obtained his BS degree from University of Electronic Science and Technology of China in 2012. Since then, he had been working in the fields of Biomedical Image Analysis. Computer Vision, Machine Learning and Multimedia, as a Ph.D. student in the Department of Computer Science at Missouri University of Science and technology (Missouri S&T). He obtained his Ph.D. degree in Jan 2017 and joined Google Inc. in March 2017.

Zhaozheng Yin received his Ph.D. degree in Computer Science and Engineering from Penn State in 2009. After his postdoctoral training at Carnegie Mellon University, he joined the Computer Science Department at Missouri S&T as an Assistant Professor since September 2011. He has been the Daniel St. Clair Fellow in the Computer Science department since 2015 and has been the Dean's Scholar in the College of Engineering since 2016. He is working with his research group on Multimedia, Imaging, Learning, and Vision. He has been an Area Chair of MICCAI2015, WACV2016 and CVPR2017.