# Indoor Localization with a Signal Tree

Wenchao Jiang Missouri University of Science and Technology, MO, USA, 65401 Email: wjm84@mst.edu Zhaozheng Yin Missouri University of Science and Technology, MO, USA, 65401 Email: yinz@mst.edu

Abstract-Indoor localization based on image matching faces the challenges of clustering large amounts of images to build a reference database, costly query when the database is large and indistinctive image features in buildings with unified decoration style. We propose a novel indoor localization algorithm using smartphones where WiFi, orientation and visual signals are fused together to improve the localization performance. The reference database is built as a signal tree with less computational cost as WiFi and orientation signals pre-cluster the reference images. During localization, WiFi and orientation signals not only offer more context information, but also prune impossible reference images, improving the accuracy and efficiency of image matching. In addition, images are described by multiple-level descriptors recording both global and local image information. The proposed method is compared with other methods in terms of localization accuracy, localization efficiency and time cost to build the reference database. Experimental results on four large university buildings show that our algorithm is efficient and accurate for indoor localization.

## I. INTRODUCTION

Indoor geo-location is an important component of smart buildings, which can be divided into two categories: indoor navigation and indoor localization. Indoor navigation provides the route to the user's destination while indoor localization tells the user where she/he is. This paper focuses on indoor localization because it has many daily applications in different scenarios such as hospitals, shopping malls, museums and office towers. In addition, indoor localization lies the basis for further navigation. The technology of indoor localization of human can also be applied to robots in a building.

Visual signal is intuitively useful for indoor localization as people generally know where they are according to what they see. A typical vision based indoor localization algorithm consists of two stages: building a reference database and online localization by image matching. The database is built by the feature representation of geo-tagged images taken within a building. When positioning, a new image around a user's location is taken and it is compared with the database to estimate her/his location.

Although vision based indoor localization has been studied for several years [1][2][3][4][7], there are still several unsolved challenges when practical implementations are considered: (1). In a common building, thousands of images can be recorded as references and millions of visual features can be detected and extracted from the images. An efficient way to build the reference database is needed. (2). In online localization, the query image will be compared with the whole database, which decreases the efficiency when the database is huge. (3). A building may have unified decoration style, so similar scenes exist in different positions, which is hard to be visually classified.

The pervasiveness of smartphones offers the opportunities to assist visual indoor localization with WiFi and orientation signals and mitigates the challenges described above. The WiFi module collects WiFi signals and inertial sensors (e.g., accelerometer and magnetometer) can be used to measure the orientation of a smartphone when its user takes photos. In this paper we fuse the visual signal and other contextual information offered by WiFi and inertial sensors to make the *energy-saving*, *efficient and accurate* indoor localization possible.

## A. Previous Work

WiFi and inertial sensors can be individually applied to indoor localization. [14] and [13] extracted sophisticated features from the raw Received Signal Strength Indication (RSSI) of WiFi signals to describe locations. However, it is possible that some hotspots are shut down or the RSSI value of a specific hotspots is changed because of the device update, which dramatically decreases the localization accuracy of merely WiFibased approaches. [15] and [16] utilized inertial sensors to perform step detection, speed estimation and heading direction determination and 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's location. Inertial sensors based approaches do 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.

For vision-based indoor localization, in [6], local affine invariant points were extracted from images. These points were quantized into visual words by K-means. Each image can be represented as a vector and each dimension of the vector represented a count of the occurrence of a visual word. The feature descriptor in terms of visual words was used for image/object matching. Wang et al. [1] proposed a coarseto-fine localization system where several candidate images were obtained by comparing the similarity of a query vector with reference vectors in the database, then a keypoint voting algorithm was adopted to determine the final matched image. Although online localization is reliable, the database is still computationally costly to be built. Liu et al. [4] considered



Fig. 1. Overview of the propose indoor localization algorithm.

the global features, including a weighted gradient orientation histogram and color histogram to localize people in indoor environment, but merely global information can not distinguish two different locations with similar decoration. To reduce the cost to build database, Chiang et al. [8] improved the traditional K-means by compressing and removing patterns at each iteration which are unlikely to change their membership thereafter. To improve the localization accuracy, Sadeghi et al. [9] adopted epipolar geometry constrains to refine the location.

Previous work also investigated signal fusion methods for indoor localization. [18] and [19] utilized WiFi signal to rectify the trajectory obtained by inertial sensors. [20] combined Radio Frequency and WiFi to improve the localization accuracy. [21] combined more signals (e.g., WiFi, sound, motion, color) to build the localization algorithm. These signal fusion methods just concatenate several sensors together to improve the localization accuracy but the problems of feature selection on signals and how to efficiently combine signals are often overlooked.

## II. PROPOSED METHOD AND ALGORITHM OVERVIEW

In this paper, we propose a novel tree-based indoor localization algorithm in which the WiFi, orientation and visual signals from smartphones are integrated into a signal tree. In the proposed algorithm, WiFi is used for coarse positioning, thus the problem of WiFi environment change is mitigated since WiFi is only used for coarse localization instead of fine localization. Inertial sensor is not used to estimate the trajectory, but to obtain the orientation towards which the user takes photos. The problem of image scenes caused by unified decoration can also be alleviated because similar scenes may have different WiFi and orientation information. This algorithm consists of two stages: building the signal tree and online localization (Fig.1).

**Building the Signal Tree** (Fig.1(a)): WiFi signals are collected in a building, tagged with hotspots' Received Signal Strength Indication (RSSI) and the positions where the signals are collected. Reference images are densely captured in a building and labeled with the 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.1(a)). Then, images in the same WiFi branch are further classified according to their orientation similarity (blue part in Fig.1(a)). 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.1(a)).

**Online Localization** (Fig.1(b)): 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 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 best match to the query image within a leaf node. Finally, 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 does not bring extra work to users. A user only needs to take a photo while the WiFi and orientation signals are automatically recorded. Then, the user can discover her/his location in the building based on the signal tree. In addition, when building the signal tree, images are naturally clustered into groups sharing similar WiFi and orientation environments. Combined with parallel computing, the time needed to build the database can be remarkably reduced. In 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 computational cost and increasing localization accuracy.

In the rest of this paper, building the reference signal tree is described in Section 3. Detailed search strategies for localization are introduced in Section 4. Then, experimental results are presented with comparisons and evaluations.



Fig. 2. WiFi fingerprints clustering. (a) Button-Up WiFi Clustering Dendrogram; The number in the leaves are indices of WiFi fingerprints. (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).

#### III. BUILDING THE SIGNAL TREE

This section presents the algorithm to build the signal tree. The surrounding sensor environment (WiFi and orientation) and image attributes of a position are fused together in the hierarchical signal tree to describe that location.

## A. Building WiFi Branches

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.

It is reported in [10] that WiFi signal gets less reliable when its RSSI is lower, so we normalize the raw RSSI by an exponential distribution

$$f_{i,j}^* = \lambda exp[\lambda \frac{f_{i,j} - f_{min}}{f_{max} - f_{min}}]$$
(1)

$$\lambda = \frac{f_{max} - f_{min}}{f_{mean} - f_{min}} \tag{2}$$

where  $f_{i,j}$  is the raw RSSI of WiFi hotspot j at location i.  $f_{i,j}^*$  is the normalized RSSI.  $f_{max}$ ,  $f_{min}$  and  $f_{mean}$  are the maximal, minimal and average RSSI of all  $f_{i,j}$ .  $\lambda$  is the rate parameter. Then, the WiFi fingerprint at location i,  $\mathbf{f}_i$ , is defined as

$$\mathbf{f}_{i} = [f_{i,1}^{*}, \dots, f_{i,j}^{*}, \dots, f_{i,N_{i}}^{*}]$$
(3)

where  $N_i$  is the number of WiFi hotspots in a building.

1) WiFi Clustering: Treating WiFi fingerprints individually is not robust to environment changes such as shutdown of some hotspots. Thus WiFi fingerprints are clustered into groups based on their WiFi fingerprint similarity and spatial distance. The clustering procedure is divided into two steps (Fig.2): Bottom-Up clustering by WiFi fingerprint similarity and Top-Down cutoff by spatial distance.

As shown in Fig.2(a), WiFi fingerprints are firstly hierarchically clustered from bottom to up. Initially, each WiFi fingerprint is a cluster. Then two clusters most similar to each other are merged into a bigger cluster. This agglomerative mergence operation is performed iteratively and stops when all WiFi fingerprints are in one cluster. The similarity metric of two clusters is defined by Ward's method [11]

$$S(A,B) = \sum_{k \in A \cup B} ||\mathbf{f}_k - \overline{\mathbf{f}}_{A \cup B}|| - \sum_{k \in A} ||\mathbf{f}_k - \overline{\mathbf{f}}_A|| - \sum_{k \in B} ||\mathbf{f}_k - \overline{\mathbf{f}}_B||$$
(4)

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

The WiFi hierarchical tree in Fig.2(a) only shows a multibranch hierarchy rather than a set of clusters. It is partitioned into several groups based on WiFi fingerprints' spatial distances. As shown in Fig.2(b), from top to down of the WiFi hierarchy, every node is checked if the maximal value of spatial distance between all pairs of WiFi fingerprints belonging to this node is less than a predefined threshold  $d_{thr}^{WiFi}$  (e.g.,  $d_{thr}^{WiFi}$ =20 meters). When the maximal value is actually less than  $d_{thr}^{WiFi}$ , the WiFi fingerprints belonging to this node will be considered to be the same group.

Note that the number of WiFi clusters is automatically defined by the fingerprint similarity and spatial distance instead of presetting by human. Fig.2(c) shows the final clustering results of WiFi fingerprints in a university building. The clustering result accurately reveals the actual WiFi environment of this building. Then, each reference image is clustered to the nearest WiFi group based on spatial distance.

### B. Building Orientation Branches

Inertial sensors, including accelerometer and magnetometer, are equipped in most smartphones. When smartphones are stable (taking photos), accelerometer measures the gravity while magnetometer measures the earth's magnetic field. Gravity and magnetic field set up a world coordinate system. Thus, every point in the phone's coordinate system can be converted to the world coordinate system by a transformation matrix.

Let  $\mathbf{Q}_{p \to w}$  denote the transformation matrix from phone coordinate system to the world coordinate system, which can be obtained by the algorithm described in [12]. Fig.3 shows the scenario when a user takes a photo, the yellow vector  $\mathbf{c}_p$  represents the orientation that the camera is towards. Note



Fig. 3. The scenario when a user takes a photo for localization.  $\mathbf{c}_p$  is a constant vector in photo's coordinate system, pointing outside the back of the phone.

that  $\mathbf{c}_p$  is a constant vector in phone's coordinate system.  $\mathbf{c}_p$  is transformed to the world coordinate by

$$\mathbf{c}_w = \mathbf{c}_p \times \mathbf{Q}_{p \to w} \tag{5}$$

Denoting  $\mathbf{c}_w = [c_{wx} \ c_{wy} \ c_{wz}]^T$ , we project the orientation to the horizontal plane in the world coordinate, i.e, vector  $O = [c_{wx} \ c_{wy}]^T$  is the orientation on the floor plan which the photo is taken towards.

1) Orientation Clustering: When building the visual database, previous work [3][4][5][7] mostly took thousands of photos manually, which is pretty time consuming. Instead, we collect continuous videos and orientation information simultaneously. Every frame of these videos is a reference image. Without loss of generality, we make the explanation with a simple floor plan. For example, eight video clips were recorded in a building following the eight routes defined in Fig.4(a). Each frame in the videos is tagged with its corresponding orientation. Each video clip was recorded following the same direction, therefore the orientations of all frames in a video are similar, naturally forming a cluster of orientation.



Fig. 4. Orientation clustering. (a) Floor plan of a building with 8 routes to record videos and sensor information. (b) Orientation distributions calculated by the data collected according to (a).

Fig.4(b) shows the distributions of eight orientation clusters corresponding to the eight routes in Fig.4(a). The orientation distribution of each video clip is not a constant impulse distribution due to noise. The orientation distribution of a video clip q is modeled by a Gaussian distribution  $N(\mu_q, \sigma_q)$ . Suppose the entire floor plan in Fig.4(a) is in one WiFi cluster, overlapped orientation clusters can be further merged into a bigger cluster. The similarity of two distributions  $q_1$  and  $q_2$  is defined as:

$$S_{q_1,q_2} = \frac{\sigma_{q_1}^2 + \sigma_{q_2}^2}{|\mu_{q_1} - \mu_{q_2}|} \tag{6}$$

If the centroid of two distributions are close to each other and their inter-distribution variance ar small, then they can be merged into a bigger cluster. In Fig.4(b), the eight orientation distributions can be clustered into four clusters. Each of the four orientation clusters is one orientation subbranch within the same WiFi branch.



Fig. 5. SIFT points in a user-taken image. The red points are the salient SIFT keypoints and the blue points are the dense SIFT keypoints. The image is equally divided into five subimages.

#### C. Building 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.1). MLID is based on Term Frequency Inverse Document Frequency(TF-IDF) [6], but we improve it in threefolds: (1) Dense Scale Invariant Feature Transform(SIFT) keypoints 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 and local information of an image is recorded. As shown in Fig.6, MLID consists of four steps:

- Feature Extraction: In Fig.5, salient SIFT keypoints (red points) are firstly extracted from an image. Dense SIFT keypoints (blue points) are then extracted in the low texture areas ignored by salient SIFT such as some parts of the ceiling and walls. Features of keypoints extracted from all reference images in a leaf node are collected into a large feature pool (represented as purple circles in Fig.6(a)).
- 2) Feature Clustering: Divisive hierarchical clustering is applied to partition SIFT features in the feature pool. In Fig.6(b), SIFT features are firstly clustered into t groups (t = 2 in Fig.6(b)) at level 1 (l = 1) based on Euclidean distance. Then each cluster in the first level are clustered into t groups. The process is performed repeatedly until every leaf of the feature clustering tree has a small set of SIFT feature descriptors (e.g., less than 100 SIFT features on the leaves). The symbol around each node represents *the mean of SIFT feature vectors*



(a). Feature Extraction (b). Feature Clustering (c). Feature Interpretation (d). Image Description

Fig. 6. Flow chart of the Multiple Level Image Descriptions (MLID) method.

*in that subtree* (called visual word) and the visual words at each level forms the visual codebook for that level. In Fig.6(b), the symbols in each dotted rectangle belong to one visual codebook.

- 3) Feature Interpretation: As shown in Fig.6(c), SIFT features in an image can be interpreted into visual words hierarchically based on the visual codebooks at different levels. A SIFT feature is interpreted as the visual word which is the closest to the SIFT feature based on Euclidean distance. For example, at level 1 of Fig.6(c), 15 SIFT descriptors are close to visual word 1 (red star) and 10 descriptors are close to visual word 2 (green circle). The interpreted visual words at level 1 are finely interpreted at following levels.
- 4) Image Description: Based on the hierarchical feature interpretation, an image can be described by multiple vectors. In each level, the dimension of the vector is the same as the number of visual words and each dimension is the count of the occurrence for corresponding visual word. For example, in level 1 of Fig.6(d), 15 SIFT descriptors belong to visual word 1 (red star) and 10 descriptors belong to visual word 2 (green circle), the description vector is [15, 10], normalized as [0.6, 0.4]. The feature descriptors are finely computed in the subsequent levels according to more and more detailed visual codebooks.

Spatial information is also considered when formulating the feature description of an image. As the yellow lines in Fig.5 illustrate, the image is first 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.

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 descriptors are finely clustered, dimension of feature vector gets larger and more detailed information is recorded. Note that, compared with K-means, there is no need to predefine how many groups we should cluster the SIFT descriptors, which is another advantage of the MLID method to handle different unknown scenes.

## IV. ONLINE LOCALIZATION

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

#### A. Coarsely WiFi Positioning

Let  $\mathbf{f}_0$  be the WiFi fingerprint submitted by the user and can be computed by Eq.3. 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 Euclidean distance, denoted as  $d_{0,n}$ . 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.

## B. Orientation Pruning

Several hundred orientation samples can be collected when a user is taking 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 Eq.6. 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.1(b), the black branches indicates the search routes. Only parts of the leaf nodes need be searched, greatly increasing the efficiency.

## C. Fine Visual Localization

Within each searched leaf node, the most similar reference image needs to be found. Algorithm1 shows the search strategy within a leaf node. The best reference image is searched from top to down of multiple vectors. 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.

#### Notations:

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

#### **Iutput:**

- 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:

  - Compute the average similarity
  - $\overline{S_{0,b}^{l}} = \frac{\sum_{b \in CP} S_{0,b}^{l}}{\sum_{b \in CP}}$

• Reference images satisfying  $S_{0,b}^l < \overline{S_{0,b}^l}$  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  $S_{0,b}^l$ break

end if

end for

## **Output:**

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

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.7, 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 Algorithm1. 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 codebook. The candidate image that has the largest 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.



Fig. 7. Determine final matched image from candidate images.

## V. 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 and visual signals. Fig.8(a) is one screenshot of the App with a simple interface. This App is capable of collecting reference signals as well as query signals.

Fig.8(b) shows how we collect signals. WiFi, orientation and visual signals are collected by the smartphone. A laser distance measurer is utilized to identify the actual location. When we collect reference signals, WiFi signals are collected uniformly and sparsely in the available regions of a building such as the hallway and public lounge. The distance of two adjacent WiFi collection positions is about 5 meters. As described in Section 3.B.1, visual signals are recorded in the format of videos. The frame rate of each video is 30fps. We keep walking with a constant speed when recording the videos. Thus, the position tagged to each frame can be interpolated by the positions of the start and end of each video recording.



Fig. 8. (a) The data collection App. (b) A laser distance measurer is used to identify the ground truth of a user's position.

#### A. Testing Environment

The proposed indoor localization algorithm is tested in 4 campus buildings whose floor plans are shown in Fig.9. Table I summarizes the information of signal trees of the 4 buildings which are used in experiment evaluations.

TABLE I
INFORMATION ABOUT THE SIGNAL TREES OF 4 BUILDINGS. NWB:
NUMBER OF WIFI BRANCHES; NOB: NUMBER OF ORIENTATION
BRANCHES; NRI: NUMBER OF REFERENCE IMAGES; NQI: NUMBER OF
QUERY IMAGES

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

## B. Comparison

Fig.10 shows some localization samples of our approach, which demonstrates the proposed localization algorithm is robust to crowded people, illumination change, scene changes and orientation shifts. Our proposed indoor localization algorithm is compared with three other approaches. (1) Multi-Level Image Description (MLID) method that only uses visual signals in the localization. (2) WiFi-based method. (3) The localization algorithm proposed by Wang et al. [1], which did not consider dense SIFT keypoints and multi-level feature vectors. The comparison is in terms of localization accuracy, localization efficiency and time used to build the reference database.

1) Localization Accuracy: Fig.11 summarizes the localization accuracy of 4 approaches in the 4 buildings. Our approach achieves the highest accuracy compared to the other 3 methods. The comparison of the approach described in [1] and MLID proposed in this paper shows it is more effective to describe images with multiple vectors, thus images' global and local information are both recoded and utilized for localization.

2) Localization Efficiency: Table II 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 of column 2 (Our signal tree method) and column 3 (Multi-Level Image Description, image-only method) proves that WiFi and orientation signals are capable to rule out impossible reference images and largely speed up the online localization.

Our method is slightly slower than Wang et al. [1]. 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 Wangs method consumes 58.06% (about 3.35s) of the total time while searching the signal tree only takes 7.75% (about 0.45s) in our method. The SIFT key detection and extraction can be speeded up with GPU parallel computing. For example, it only needs 0.07 second to detect and extract SIFT keypoints from a  $1024 \times 768$  image by a GPU [17]. We leave this as our future work. The WiFi-only method is the fastest, but its localization accuracy is very low (Fig.11).

3) *Time Used to Build the Database:* Table III 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

 TABLE II

 Average Time Used for Localization (Seconds)

Building	Ours	MLID	Wang [1]	WiFi
1	5.77	10.42	5.48	0.0094
2	5.63	9.63	5.22	0.006
3	5.80	10.79	5.11	0.0050
4	6.20	11.23	4.91	0.0014

the time cost of the image-only(MLID) method). Note that building or updating a reference database including thousands of images for a skyscraper can be a very time-consuming task. However, in our signal tree method, WiFi and orientation signals pre-cluster reference images into several leaf nodes, thus a complex problem is divided and conquered by small problems.

TABLE III Time used to Build the Database (Hours)

Building	Ours	MLID	Wang [1]	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

#### C. Discussion

From the experimental results, our method takes more time to build the database compared to WiFi-only method and our method takes more time for query compared to WiFi-only and Wang et al. [1] methods, but the accuracy of our method is far better than the other methods. Considering the evaluation metrics together, our proposed method is competitive and its effectiveness is multi-folds.

For a fingerprint based algorithm, it is time-consuming to collect a complete reference dataset to satisfy the high accuracy requirement. In this paper, we just uniformly and sparsely collect WiFi fingerprints in a building. We collect reference images in the format of videos (Scetion 3.B.1), which largely speed up the data collection and updating.

The proposed algorithm deals with the problem of WiFi environment change in two ways. As discussed in section 3.A.1, WiFi fingerprints are clustered into groups, thus our tolerance to WiFi environment change is getting higher. In the scenario that WiFi environment is largely changed, we can increase the number of search branch h described in section 4.A to allow more WiFi branches to be searched.

#### VI. CONCLUSION

In this paper, we propose a novel signal-tree based indoor localization algorithm by fusing WiFi, inertial and visual signals. Our proposed algorithm is accurate as well as efficient because it makes full use of the advantages of three signals and finds the matched signal in a hierarchy manner. The proposed Multi-Level Image Description (MLID) method is very effective to describe and compare images with coarse-tofine image descriptors. In our future work, we plan to provide intelligent guidance to the user allowing a second localization in the extremely challenging cases when the first localization is not reliable.



Fig. 10. 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). Low texture scene.



Fig. 11. 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. MLID: Multi-Level Image Description method that only uses visual signals; Ours: signal tree (MLID + WiFi + Inertial sensor)

#### References

- J. Wang et al. "Coarse-to-fine vision-based localization by indexing scale-invariant features". IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics, 36(2):413-422, 2006.
- [2] R. Mautz and S. Tilch. "Survey of optical indoor positioning systems". Indoor Positioning and Indoor Navigation, 2011.
- [3] J. Z. Liang et al. "Image Based Localization in Indoor Environments". Computing for Geospatial Research and Application, 2013.
- [4] H. Liu et al. "Combining color histogram and gradient orientation histogram for vision based global localization". Systems, Man and Cybernetics, 2009.
- [5] M. Werner et al. "Indoor positioning using smartphone camera". Indoor Positioning and Indoor Navigation, 2011.
- [6] J. Sivic and A. Zisserman. "Video Google: A text retrieval approach to object matching in videos". ICCV, 2003.
- [7] D. Nister and H. Stewenius. "Scalable recognition with a vocabulary tree". CVPR, 2006.
- [8] M. C. Chiang et al. "A time-efficient pattern reduction algorithm for K-means clustering". Information Sciences, 181(4): 716-731, 2011.
- [9] H. Sadeghi et al. "A weighted KNN epipolar geometry-based approach for vision-based indoor localization using smartphone cameras". Sensor Array and Multichannel Signal Processing Workshop, 2014.
- [10] E. Martin et al. "Precise indoor localization using smart phones". International Conference on Multimedia, 2010.
- [11] R. Xu and D. Wunsch. "Clustering". IEEE Press, 2009.

- [12] S. Madgwick et al. "Estimation of IMU and MARG orientation using a gradient descent algorithm". ICORR, 2011.
- [13] Krishna Chintalapudi, Anand Padmanabha Iyer, and Venkata N Padmanabhan, "Indoor localization without the pain". International Conference on Mobile Computing and Networking. ACM, 2010.
  [14] Joydeep Biswas and Manuela Veloso, "Wifi localization and navigation
- [14] Joydeep Biswas and Manuela Veloso, "Wifi localization and navigation for autonomous indoor mobile robots". International Conference on Robotics and Automation. 2010.
- [15] Ionut Constandache, Romit Roy Choudhury, and Injong Rhee, "Towards mobile phone localization without war-driving". Infocom, 2010.
- [16] Shohei Koide and Masami Kato, "3-d human navigation system considering various transition preferences". International Conference on Systems, Man and Cybernetics, 2005.
- [17] Changchang Wu et al., "Multicore bundle adjustment". Computer Vision and Pattern Recognition. 2011.
- [18] Shizhe Zhang et al., "Indoor location based on independent sensors and wifi". International Conference on Computer Science and Network Technology. 2011.
- [19] Anshul Rai et al., "Zee: zero-effort crowdsourcing for indoor localization". International Conference on Mobile Computing and Networking. 2012.
- [20] Yin Chen et al., "Fm-based indoor localization". International Conference on Mobile Systems, Applications, and Services. 2012.
- [21] Martin Azizyan, Ionut Constandache, and Romit Roy Choudhury, "Surroundsense: mobile phone localization via ambience fingerprinting". International Conference on Mobile Computing and Networking. 2009.