A Generic Fusion Framework for Multi-modal Localization

Various signals have been studied for localization, such as GPS, WiFi, Bluetooth, geomagnetism and INS. As these signals have their own strengths and limitations, recent works have been focusing on how to combine them to achieve high performance. Despite promising results, previously proposed schemes are often highly engineered and specialized for few (two or three) signals whose data have to be fully available at localization time. They can be hardly extended to support flexible combination of multi-modal signals with different sampling rates, and arbitrary addition and removal of signals at any time as one roams between indoor and outdoor areas.

We propose SiFu, a highly accurate and generic localization framework to fuse a dynamic number and combination of heterogeneous signals. SiFu unifies multi-modal signals into a probabilistic platform. Using a generic likelihood computation module, signal value is converted to location likelihood on pre-defined grid points in the feasible area of the map. For RSSI vectors, we present a novel and highly accurate likelihood computation module which employs a denoising autoencoder to learn latent representation of the signals. The likelihoods, independently computed for each signal, are then fused with a novel weighted likelihood scheme to achieve high accuracy. The weighted output is then integrated with inertial sensor measurements using a particle filter. SiFu is simple to implement. It is incrementally extensible to new signals without the need for retraining the whole system. We conduct extensive experiments in three markedly different and representative sites, and show that SiFu achieves significantly better performance as compared to other state-of-the-art approaches, cutting the localization error by more than 30% in our experiments.

CCS Concepts: • **Networks → Location based services**; • **Human-centered computing → Ubiquitous and mobile computing systems and tools**.

Additional Key Words and Phrases: fusion localization, signal fusion, denoising autoencoder, weighted likelihood

ACM Reference Format:
. 2020. A Generic Fusion Framework for Multi-modal Localization. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 0, 0, Article 0 (2020), 20 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Localization technology has wide application in navigation, location-based marketing, geo-fencing, etc. For open outdoor environment, GPS usually provides acceptable solution. However, in urban-canyon, semi-indoor or deep indoor setting, its accuracy deteriorates due to weak or unavailable signals. For complex indoor cases, fingerprinting emerges as a promising technique due to its applicability and accuracy. In fingerprinting, the site is first surveyed to label the signal values with their locations, the so-called "fingperprints." Given the fingerprints, user location can be estimated based on the signals he/she samples. Various fingerprinting signals have been considered, such as WiFi, bluetooth, magnetic field, etc. To further enhance localization accuracy and user experience, inertial navigation system (INS) may be added.

Every signal has its own strengths and limitations. For example, GPS works reasonably well outdoor but not so indoor. Radio-frequency (RF) signal such as WiFi or Bluetooth are pervasive and differentiable over long range, leading to its deployability in indoor environment [9, 20]. However, due to multi-path and fading effects, it

Author's address:

2474-9567/2020/0-ART0 $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

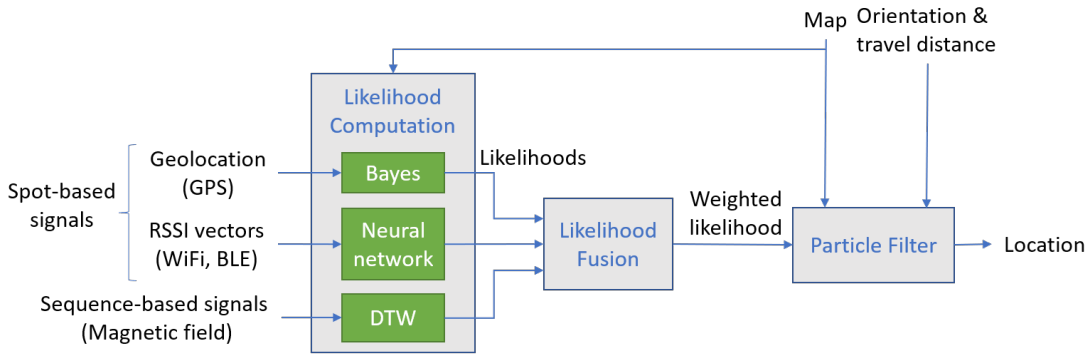


Fig. 1. SiFu framework.

generally suffers from relatively high noise. The localization accuracy also depends much on the strength and density of the signals. For some signal such as WiFi, its sampling rate in a mobile phone may be sparse (once every tens of seconds), which adversely affects user experience. Geomagnetism is another signal explored in literature. Its strengths lie on its omnipresence (no additional infrastructure), fast sampling rate (tens of samples per second) and low noise [12]. Its drawback is global ambiguity where a certain geomagnetic sequence may be matched to multiple places in the area. Inertial sensors are available in almost every mobile phone, and may be used to provide user movement information. However, it provides only relative location information and error may accumulate and diverge over time. As a result, INS is often combined with other signals to enhance localization accuracy.

In view of the complementarity of these signals, recent research has been focusing on fusing different signals to combine their strengths while mitigating their weaknesses [5, 8, 11, 13, 14, 25, 26]. These works have considered only a few (two or three) signals combined in a highly specialized and customized manner according to the specific signal characteristics. As a result, extending them to include other signals is often not straightforward or possible. All the signals are often assumed fully available at the time of localization decision, while in reality due to widely different signal sampling rates, missing signal values result. The remedy may be to use the last measured ones, make prediction based on history, or reduce the localization frequency to the slowest sampling rate. All these are clearly not ideal or satisfactory.

In this work, we propose and study SiFu, a generic localization *framework* for heterogeneous **si**gnal **fu**sion that accommodates arbitrary combination of many multi-modal data. Our framework supports dynamic set of signals in terms of type and number, enabling seamless roaming over a wide range of signal environment. Consider a user walks from one place to another with markedly different sets of available signals, which may involve arbitrary combination of WiFi, Bluetooth, magnetic fields, GPS, etc. If algorithm were designed highly customized for some certain set of signals, the user needs to switch among a plethora of algorithms according to his/her current available signals. This is tedious, if not infeasible, to support large-scale deployment. SiFu overcomes this problem by supporting signal addition and removal at any time. It is a platform, incrementally extensible to new signals without the need for retraining the whole system.

SiFu adopts a probabilistic framework as illustrated in Fig. 1. It considers broadly two classes of signals depending on how they are processed for localization. Sequence-based where localization is based on a group of consecutive signal samples (e.g., magnetic field), and spot-based where location is based on single sample. Spot-based signals are further sub-divided into whether the input already is in the form of global latitude and

longitude coordinates (e.g., GPS or Beidou), or a vector of received signal strength indicators (RSSIs) for signals such as WiFi and Bluetooth.

A signal is fed into a likelihood computation module, which independently converts the signal value into a location likelihood on pre-defined grids in the feasible area of the map. BAYES (Bayesian) module is to convert the global coordinates and accuracy to the likelihood on grids. For a signal strength vector, NEURAL NETWORK module returns the likelihood by vector comparison using a neural network. For sequence-based signals, the module is DTW (Dynamic Time Warping) which estimates the likelihood through matching the input sequence against the fingerprints on the grid points. Note that the likelihood computation modules are generic, i.e., they are applicable to different signal types within a class.

The independently computed likelihoods are then fused in a weighted fashion in the likelihood fusion module, which assigns weights according to the individual model performance at each grid points in the training process of the fingerprints. The weighted likelihood is then integrated with user movement information (as obtained from INS) using a particle filter, with feasible areas of the map as a constraint.

As compared with other fusion works, SiFu advances in the following ways:

- *A probabilistic framework fusing arbitrary combination of heterogeneous signals:* Our framework is a platform unifying and fusing heterogeneous signals. Because of the probabilistic formulation in the localization process, it can accommodate the presence or absence of signals due to different signal sampling rate, signal addition, signal removal, missed samples, etc. It hence achieve high elasticity and scalability in signal combination without the need for retraining and redesigning the system.
- *Novel likelihood computation for spot-based RSSI vectors:* In RSSI vector comparison, how to deal with missing or non-overlapped signal values between two vectors is a critical challenge and has determinant impact on accuracy. In our NEURAL NETWORK module, we present a novel machine learning technique which employs a denoising autoencoder to learn the latent representation of the signals in the interested area. As opposed to the conventional approaches based on raw signal readings (e.g., cosine similarity or euclidean distance), SiFu estimates likelihood in the latent space. Because the deep features are learnt from all fingerprint signals, this overcomes the missing values problem, achieving higher robustness and accuracy.
- *A weighted approach to achieve high accuracy:* As it is nearly impossible to find a flawless localization model, signals may give different localization accuracy at different locations in the interested area. It is very likely that bad performance of the system components will undermine the performance of the whole fusion localization system. Some of the previous works have attempted to address this issue, but they are usually customized for specific signals and hence, cannot embrace new signals into the system. In SiFu, we employ the theory of weighted likelihood in Bayesian analysis to fuse multi-modal signals. Through a location-dependent weighting for each signal, it combines signals more smartly to achieve high accuracy.

SiFu is simple to implement. We have implemented it in Android phones and have conducted extensive experiments to validate its performance. For brevity and concreteness, we select four signals of widely different characteristics in our experiments, WiFi, magnetic field, GPS and Google Fused Location API (which includes GPS, if any). Using these signals as examples, we demonstrate how they can be applied in our framework for multi-modal localization. (Note that our results are by no means limited to these signals, and may be equally and simply extended to other signals.) We conduct our experiments in three areas, namely a corridor, an indoor open-space area and a semi-indoor area in our university. We show that SiFu is highly accurate in localization, improving substantially the accuracy as compared with other state-of-the-art schemes (by more than 30% in our experiments). By demonstrating an indoor and outdoor transition, we further validate the ability of SiFu to handle arbitrary combination of signals to achieve seamless roaming without switching algorithms between outdoor and indoor.

The rest of this paper is structured as follows. We discuss in Section 2 the related work. Section 3, 4 and 5 presents the likelihood computation, likelihood fusion and particle filter, respectively. We evaluate SiFu performance and present its illustrative experimental results in Section 6, followed by conclusion in Section 7.

2 RELATED WORK

As GPS signal cannot penetrate into indoor environments, researchers have been studying alternative signals for indoor localization. Traditionally, radio-frequency (RF) signals such as Wi-Fi [1, 4, 23] and Bluetooth [7] are popular for fingerprint-based localization. Recently, several other fingerprinting signals have also been studied, including geomagnetic field [18, 21], vision [22], light [24], etc. While encouraging results have been reported, due to the limitations of signals, the performance of these single-modal systems cannot be generalized to every scenario.

To overcome that, researchers have recently focused on signal fusion. The fusion approaches can be generally classified into probabilistic and non-probabilistic one. Probabilistic approaches model localization problem using a probability and statistical framework, while non-probabilistic approaches do not.

Regarding probabilistic approaches, user location is estimated by finding the probability that the user is at a location. Previous works take advantages of conditional independence among different signals, hence they can incorporate new signals or remove a signal straightforwardly. An issue of it is the highly customized nature of the algorithm. WaP [10] calculates probability through room and entrance detection in a customized way, and cannot be extended beyond WiFi and inertial sensors to other signals. Magicol [18] fuse WiFi, magnetic field and INS signals. However, the approach is heavily based on WiFi, which is not generic or flexible enough to exclude it or fit in some other new signals. In contrast, SiFu is a generic platform which supports arbitrary combinations of signals with efficient addition and removal of signals.

Non-probabilistic approaches often employ the coarse-to-fine method. This method aims to reduce the search space gradually by considering multiple signals. A typical example is [3], which constructs heterogeneous filters for WiFi, sound and motion, which are sequentially applied to prune away candidate location points. The work in [13] works in a similar manner where intermediate dead reckoning and WiFi results are used to constrain the matching space of magnetic field. Some other works do not adopt a sequential structure as coarse-to-fine method. The work in [25] hybrids WiFi and magnetic field fingerprint in the input level. It stacks WiFi and magnetic field in a single fingerprint vector and estimates user location using kernel discriminant analysis with the KNN algorithm. WiMag [8] first processes WiFi, magnetic and inertial signals independently, and then selects the ones according the phone status. The work [26] formulates an optimization problem to fuse vision with inertial sensors. While these works are impressive, they are clearly designed for a specific combination of signals and are difficult to embrace additional new signals. Furthermore, all signals have to be available at the localization time as the fusion module either depends on the other components or jointly process all signals at the same time.

In [5], an F-score weight is proposed to rate the importance of WiFi and magnetic signals in location prediction. However, the weight setting method relies on an F-score metric in classification problem, which may be inappropriate in localization decision where unavoidable intrinsic errors exist. In SiFu, we propose a location-dependent weight for different signals such that the negative impacts of the poorly performed submodule are mitigated, leading to a more accurate signal fusion.

Motion-assisted localization techniques are often used to enhance location estimation [8, 13, 14, 18]. Using Pedestrian Dead Reckoning (PDR), user movement information such as step length and orientation can be extracted from inertial sensors including gyroscope, accelerometer and magnetometer. Such information plays an important role in localization because local displacement helps calibrate location estimation and reduce localization error, leading to its popularity in localization systems. Algorithms including particle filter and Kalman filter are often used to incorporate INS information. While some of the works cited above makes improvements

Table 1. Major symbols in SiFu.

Notation	Definition
$\boldsymbol{S}$	A vector containing all observed signals
s	An observation of a particular signal type
x	The states in the interested area
d	Dimension of WiFi latent representation
L	Number of layers in WiFi encoder/decoder
σ_w	The observation model parameter σ for WiFi
σ_m	The observation model parameter σ for magnetic field
r	Magnetic matching range
w_{ij}	The weight at RP i for signal j
N	Number of particles in particle filter

by making modifications on these algorithms according to the signals, we use only a typical filter in our work, so that our work is generic that can suit any type of signals in our consideration.

3 LIKELIHOOD COMPUTATION

Each sampled signal in SiFu framework is independently converted to a likelihood at the grid points in the venue. In this section, we discuss the module design of likelihood computation in SiFu. After discussing preliminaries in Section 3.1, we present the likelihood computation modules for different classes of signals. We introduce the BAYES module for geolocation coordinate data in Section 3.2, followed by the NEURAL NETWORK module for spot-based signals in the form of RSSI vectors in Section 3.3. We end in Section 3.4 the DTW module for sequence-based signals.

3.1 Preliminaries

The interested area is discretized into grids of reference points (RP). Then likelihood is computed at each RP in the grid. Note that for signals using fingerprinting approach, a fingerprint signal should be obtained at each RP.

Mathematically, we are interested in finding the probability that the user locates at different locations given the observed signals, denoted as $p(x|\boldsymbol{S})$, where x denotes the states (locations) in the interested area and $\boldsymbol{S} = (s_1, ..., s_n)^T$ is a vector containing n observed signals.

A particle filter, which will be discussed in the later section, is used to estimate this distribution. As such, we are required to find the measurement likelihood $p(\boldsymbol{S}|x)$ for updating the weights of particles in the particle filter. Since different signals are sampled by different sensors, we can assume conditional independence among different signals. So the likelihood $p(\boldsymbol{S}|x)$ is given by

$$p(\boldsymbol{S}|x) = p(s_1, ..., s_n|x) = \prod_{i=1}^{n} p(s_i|x). \tag{1}$$

In other words, for every signal s_i, we independently compute a measurement likelihood $p(s_i|x)$, which is the probability of observing the sampled signal s_i given the user is located at different RPs. This serves as the foundation for our generic fusion localization framework.

3.2 Geolocation of spot-based signals

Geolocation data provide the global coordinates of the user in the format of $[latitude, longitude, accuracy]$. GPS and Google Fused Location API are two examples of this class of signal. While GPS is only available outdoors,

Google API can locate users virtually everywhere with Internet connection. In the two examples mentioned above, the reported accuracy is defined as the radius of 68% confidence, suggesting that there is a 68% probability that the true location is inside the circle centered at the reported location with radius equal to the reported accuracy. Also, note that alignment between map coordinate systems may be required here.

We model this class of signal with a bivariate Normal distribution characterized by a mean $\boldsymbol{\mu}$ and covariance Σ. The mean vector is constructed using the reported location as $\boldsymbol{\mu} = (\tilde{x}, \tilde{y})^T$, where $(\tilde{x}, \tilde{y})$ are the reported location in the local map coordinate system. For the covariance matrix Σ of the distribution, it is approximated from the reported accuracy. With no prior knowledge, it is reasonable to assume that both dimensions of the Gaussian distribution are uncorrelated and have the same variance. Note that the $(1-\alpha)$ interval for a multivariate Gaussian distribution is given by

$$(\boldsymbol{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu}) \leq \chi_p^2(\alpha), \tag{2}$$

where $\boldsymbol{\mu}$ and Σ are the mean vector and covariance matrix of the Gaussian distribution respectively and $\chi_p^2(\alpha)$ is the α quantile of χ_p^2.

As $(\boldsymbol{x} - \boldsymbol{\mu})$ represents the difference between a point $\boldsymbol{x}$ and the mean $\boldsymbol{\mu}$, using the localization accuracy, we can obtain $\sigma_x = \sigma_y = a/\sqrt{2.27}$, where a is the accuracy reported. Hence, the covariance matrix $\Sigma = \begin{bmatrix} \sigma_x & 0 \\ 0 & \sigma_y \end{bmatrix}$.

Then the measurement likelihood $p(s|x)$ is defined as given that locating at x, the probability of the mean being equal to the reported location. By Bayes' theorem,

$$p(s|x) = \frac{p(s)p(x|s)}{p(x)}. \tag{3}$$

We place a uniform prior for the mean as it is equally probable everywhere, i.e., $p(s)$ is constant. As $p(x|s)$ is the density function of Gaussian distribution, which can be computed with a known mean and covariance matrix, the likelihood is given by

$$p(s|x) \propto p(x|s) = \frac{1}{2\pi\sqrt{|\Sigma|}} \exp((\boldsymbol{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu})), \tag{4}$$

where $\boldsymbol{x}$ is the vector representing the location where the likelihood is computed.

3.3 RSSI vector of spot-based signals

For spot-based RSSI vectors, the idea for likelihood computation at RP x_i in the localization step is to obtain a similarity measure $\delta(s, t)$ between the sampled signal s and the fingerprint signal t at RP x_i. By assuming a normally-distributed observation model on the similarity measure, we obtain the probability of observing the RSSI vector s at the RP x_i as

$$p(s|x_i) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{\delta(s,t)}{\sigma^2}}, \tag{5}$$

where σ is the standard deviation of the normal distribution, which is a parameter for the sensitivity of the observation model.

The similarity measure can sometimes be obtained by comparing raw readings. However, it is not good enough for signals of higher dimension where signal noise and missing features may severely mislead the similarity computation, and eventually the likelihood computation. Therefore, we propose to make use of the latent representation of signals to compute similarity measure. The intuition is that the latent representation is the deep features learnt from all collected fingerprint signals in the interested area. Signal noise and missing measurements will be implicitly considered in the latent representation generation. Using such features is expected to give a more precise and robust similarity measure. Comparison between two RSSI vectors on a representation with a

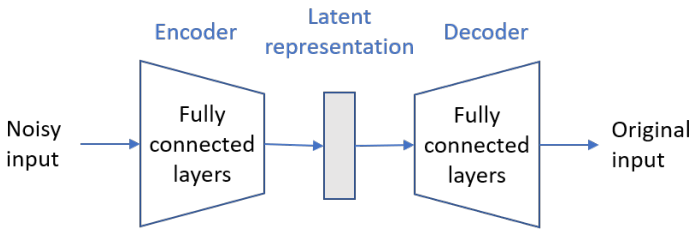


Fig. 2. The denoising autoencoder for learning latent representation of signals.

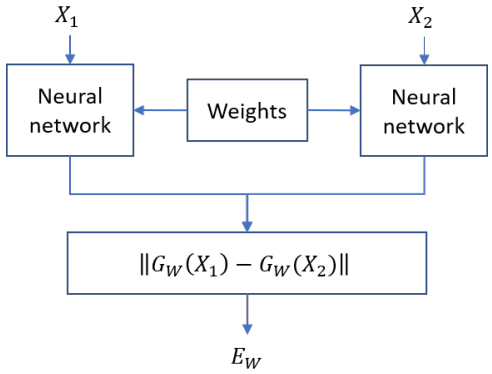


Fig. 3. A Siamese network architecture for similarity measure computation.

fixed dimension is also fairer. For localization purpose, we expect that if the physical locations of two signals are close, their latent representations will also be similar.

Machine learning techniques are used to learn the signal latent representation. We first train a denoising autoencoder. Fig. 2 illustrates the structure of the autoencoder. The input is a noisy version of the fingerprint signals collected in the offline survey phase. Noise can be injected in the following two ways: (1) randomly masking features, which makes the model robust to the situation when some features are missing, and (2) adding Gaussian noise to simulate local measurement errors. The autoencoder consists of an encoder that learns to encode the noisy signal into a latent representation of dimension d, as well as a decoder that recovers the original clean signal from the representation. In our autoencoder, both encoders and decoders are made up of L fully connected layers.

Compared with the work in [1], we use only one autoencoder to learn from all fingerprints in the interested region, instead of one autoencoder specifically for one RP. On the one hand, our method requires less efforts on data collection and model training. On the other hand, through learning to encode and decode the signals, similar signals in the whole localization region will generate similar latent representation, making it useful for signal differentiation for localization.

With the denoising autoencoder, we use a Siamese network demonstrated in Fig. 3 to compute the similarity measure. In the Siamese architecture, two networks share the same weight W on two different inputs X_1 and X_2. Given the the sampled signal s and fingerprint signal t, the similarity measure is compute as

$$\delta(s,t) = ||G_W(s) - G_W(t)||, \tag{6}$$

where G_W denotes the neural network with weight W. Here, the networks are essentially the encoder part of the autoencoder and the similarity measure computes the Eucliedean distance between the latent representations of two RSSI vectors.

In our experiment, WiFi signal is taken as an example for this category of signal. WiFi RSSI readings are widely used in many fingerprint-based indoor localization systems. Because of signal attenuation during propagation, RSSI from each AP reflects how far signal propagates in physical space from that AP. A WiFi RSSI vector, which consists of RSSI values from all detected APs at the moment, contains information rich enough to give a rough estimation of the user location. However, due to multipath and fading effects, WiFi signals are noisy in indoor environment and sometimes a few APs may not be detected. Here, our denoising autoencoder addresses these problems and helps generate a better similarity measure. For preprocessing, we utilize the WiFi signal preprocessor in [6].

3.4 Sequence-based signals

Due to difference in sensor sampling rate and user walking speed, two sequences may not be aligned, making comparing two temporal sequences very difficult. To mitigate this issue, we consider the Dynamic Time Warping (DTW) algorithm which can stretch or compress the time dimension of sequences for optimal matching. The DTW algorithm finds an optimal warping path for two sequences s and t. In the warp path, data points in sequence s are associated with data points in sequence t. Finally, it returns the distance of the warp path as the sum of difference between data points in the warp path as

$$\delta(s,t) = \sum_{k=1}^{n} |s_{ik} - t_{jk}|, \tag{7}$$

where s_{ik} and s_{jk} are the $k-th$ data points on the warp path from the sequences s and t respectively.

The distance itself is a similarity measure, thus we obtain the likelihood based on a normally-distributed observation model as

$$p(s|x_i) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{\delta(s,t)}{\sigma^2}}. \tag{8}$$

Since geomagnetic field reading is only a 3-dimensional vector, usually it cannot provide sufficient information for localization. For example, ambiguity is common where the magnetic field values at two very different locations are similar and this cannot be resolved easily. Hence, it is often necessary to consider multiple magnetic field readings within a period of time to predict user location. Therefore, we select magnetic field as the representative of sequence-based signals in our experiment.

As the complexity of the DTW algorithm is $O(MN)$ where M and N are the lengths of the sequences respectively, we further improve the efficiency of our system by using fastDTW [17], a fast approximation of DTW that has linear time complexity. In addition, we use the last user location to limit the search space to speed up the matching process. The magnetic matching range denoted as r determines how large the magnetic matching space will be. Only the magnetic field fingerprints within r m from the last predicted location will be matched against the sampled signal for likelihood computation.

4 LIKELIHOOD FUSION

Given the independently computed likelihoods, in this section we discuss how to combine them into a joint likelihood in order to update particle weight in the particle filter. Normally, this can be done using Eq.(1). Given the likelihoods, it only computes their product directly. However, these likelihoods may be inaccurate because of the model defects, producing an unreliable final result. To further address the vulnerability of the individual models, we introduce the theory of weighted likelihood [16].

In general, weighted likelihood $\prod \mathcal{L}^{w_{ij}}$ is an approximation of Bayesian inference $\prod \mathcal{L}$. It is believed that weighted likelihood provides good approximation to the posterior distribution with properly selected weights. In localization, signals may suffer from ambiguous readings and noise, thus it is possible to receive similar readings at two locations with far physical distance. They can seriously affect the model training phase and consequently affect the performance of the model and the shape of likelihood function. In [2], weighted likelihood is adopted to provide outlier-resistant solution. In our work, we adopt weighted likelihood to avoid negative influence of bad model performance by spatially controlling the contribution of the signal to the likelihood. With a small weight, the signal will contribute less since low likelihood will be pushed closer to the high likelihood end. Discrimination between signals will be reduced. In the extreme case where the weight is 0, the weighted likelihood will become 1 no matter what the original likelihood is. In this case, the contribution of the signal is removed and the decision power is shifted to the other signals. On the contrary, signal with a high weight will have likelihood values useful for differentiating the location.

In this weighted scheme, weights, which indicate the wellness of the models, play an important role. Therefore, selection of weights is crucial for SiFu. Intuitively, signals with more distinguishable features should be assigned with higher weight, and vice versa. Some work determines signal weights through result-oriented approach [5]. They infer distinguishability from prediction performance on separated testing dataset with large size. In SiFu, weights are inferred from the performance of models employing a fingerprint approach all over the interested area. For signals that do not use a fingerprint approach, we assign the weight to 1 so that the original likelihood function is preserved.

In our work, weights are location-specific because signal distinguishability may vary among the interested site. Let w_{ij} be the weight of signal type j at RP x_i and V_{ij} be the validation dataset consisting of the signals of type j collected there. To find w_{ij}, we test the localization performance using single signal source signal j on the validation dataset V_{ij}. We compute the localization error as the euclidean distance between ground truth and the predicted location. Then the weight is computed through

$$w_{ij} = 1 - \tilde{D}_i, \tag{9}$$

where $\tilde{D}_i$ is normalized mean error at RP x_i.

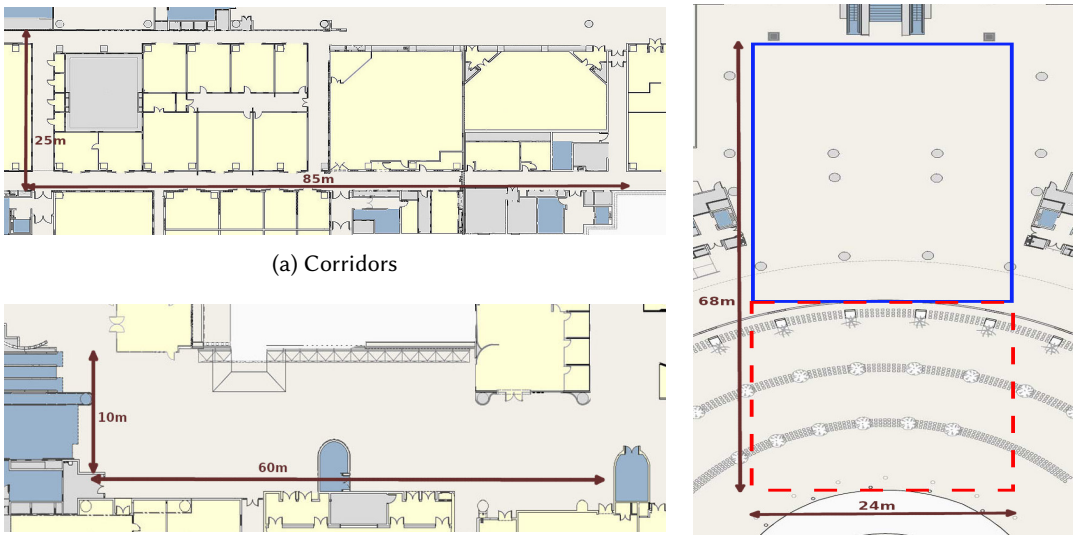
Hence, the measurement likelihood that we are interested in is approximated by

$$p(s_1, ..., s_m|x_i) = \prod_{j=1}^{m} p(s_j|x_i)^{w_{ij}}. \tag{10}$$

5 PARTICLE FILTER

Upon obtaining the weighted likelihood, a particle filter is used to combine it with user movement information obtained from INS. Orientation and travel distance of the user are estimated from the readings of gyroscope, magnetometer and accelerometer (e.g., based on the works in [15, 19] or others). We consider a typical particle filter consisting of the following four steps: (1) Particle prediction, (2) Weight update, (3) Location estimation, and (4) Resampling.

The *particle prediction* step relies on the estimated orientation and travel distance information to propagate the particle. In the *weight update* stage, the weighted likelihood computed in the previous section will be used. Note that we compute likelihoods at the RPs in our computation grid. While particles do not necessarily fall exactly on the RPs, they will take the likelihood value of its closest RP. Furthermore, if the particle violates the map constraints such as moving across a wall, we set the weight to 0. The weight is then normalized.



(a) Corridors

(b) Indoor open space

(c) Semi-outdoor environment

Fig. 4. Site maps in our experiments.

Given a set of N particles $((x_i, y_i), w_i)$, the user *location estimation* step is to compute user location by the weighted average of the positions of the particles as

$$(\hat{x}, \hat{y}) = \frac{1}{N} \sum_{j=1}^{N} w_i(x_i, y_i). \tag{11}$$

Finally, the *resampling* step corrects the set of samples based on the evidence. While more particles will be resampled in a region that has a higher probability density, we also add some diversity to the samples.

6 ILLUSTRATIVE EXPERIMENTAL RESULTS

We have conducted extensive experiments in various sites to validate our design. In this section, we first describe our experiment settings, comparison schemes and evaluation metrics in Section 6.1. Then we present illustrative results in Section 6.2.

6.1 Experimental settings and performance metrics

We have evaluated our work in three testbeds in our university. They cover a corridor area as shown in Fig. 4a as well as two larger testbeds including a indoor open space shown in Fig. 4b and an atrium area shown in Fig. 4c which is a semi-outdoor environment. While there are 91 APs in the corridor area covering a few long corridors, we find 93 APs in the $60 \times 10m^2$ indoor open space area and 103 APs in the $68 \times 24m^2$ atrium area.

In the atrium area, we test the ability of handling arbitrary combinations of signals of our framework. The blue solid box in the map indicates the WiFi and magnetic fingerprint region and the red dashed box shows where GPS is available. In the experiment, we walk from fingerprint region to non-fingerprint region such that the set of signals used switch from WiFi, magnetic and Google API to GPS and Google API. Here we train a In/Out Region Detection module as described in [6] to detect whether a WiFi signal is observed within the fingerprint region in the online phase. If so, we use it for location estimation. Otherwise, we discard it.

Table 2. Default parameters.

Parameter	Default value
Grid size	$1.5m$
N	$1,000$
d	32
L	3
r	$9m$
σ_w	5
σ_m	$10,000$

The parameter settings in our experiments is shown in Table 2. We test our systems on Android devices. The sampling frequency is around one sample every few seconds for WiFi sensor and tens of samples every second for magnetic field and other INS sensors. Furthermore, Google Fused Location API returns a location every few seconds while the sampling rate of GPS is generally faster.

In our experiments, we compare our work with another two state-of-the-art localization algorithms and Google API as well as a naive version of our system. The algorithm and parameters settings are referred to their works.

- WiDeep[1]: The scheme applies stacked denoising autoencoders to denoise WiFi RSSI. For every reference point, an autoencoder is trained correspondingly. User location is estimated by selecting the point whose autoencoder recovers WiFi RSSI reading the best. Noisy training readings are simulated by noise injector which assumes Gaussian distribution on environment noise.
- F-Score-Weighted (FSW)[5]: The scheme measures weight of WiFi and magnetic field through F-score. Location is estimated by minimizing weighted sum of log likelihoods of two signals. Signal likelihood is calculated by Gaussian probability density function.
- Naive version of SiFu: It leverages the same set of signals as SiFu, namely, WiFi, magnetic field, GPS and Google to compute likelihoods, and then fuses with INS signals using a particle filter. In the naive version, signals are not weighted. The joint measurement likelihood is computed only by multiplying all the likelihoods together.
- Google Fused Location API: It is a fusion localization algorithm created by Google that intelligently combines signals including WiFi, Bluetooth and GPS.

We use the following performance metrics for comparison:

- *Localization error:* Localization error is the Euclidean distance between the estimated location and the ground truth. It is the most direct and common metric to measure the overall performance of localization systems. Besides, error distributions will be examined for a more comprehensive evaluation.
- *Computation time:* Computation efficiency is of great importance for localization systems, especially for real-time systems. In our experiments, we measure the average time required to compute the location after signals have been received.

6.2 Illustrative results

We first validate our likelihood computation module for spot-based RSSI vectors. We compare our model which computes based on the distance in the latent space with the other schemes that is based on different metrics, such as Euclidean distance and cosine similarity, in the raw signal readings. Figure 5 shows the comparison in different sites. It is clear that our proposed method is better in all three sites. In the corridor area, the three comparison schemes perform almost the same, probably because WiFi signals there are stronger and more stable. But when we test in open areas like the indoor open space and atrium, the multi-path fading effect can become

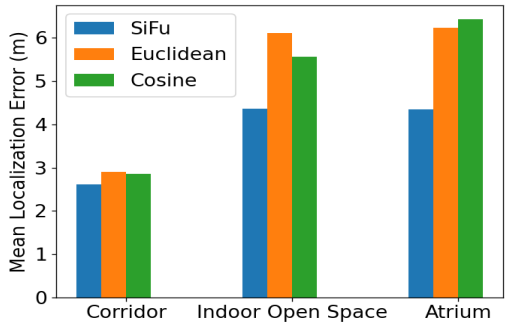


Fig. 5. Mean errors of different schemes for spot-based RSSI vectors in different sites.

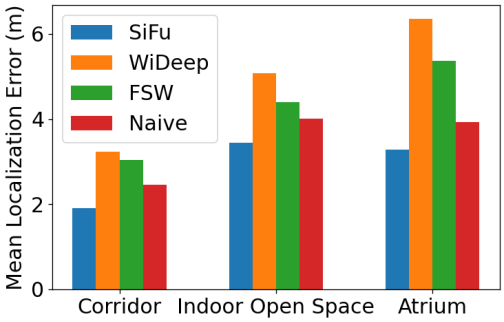
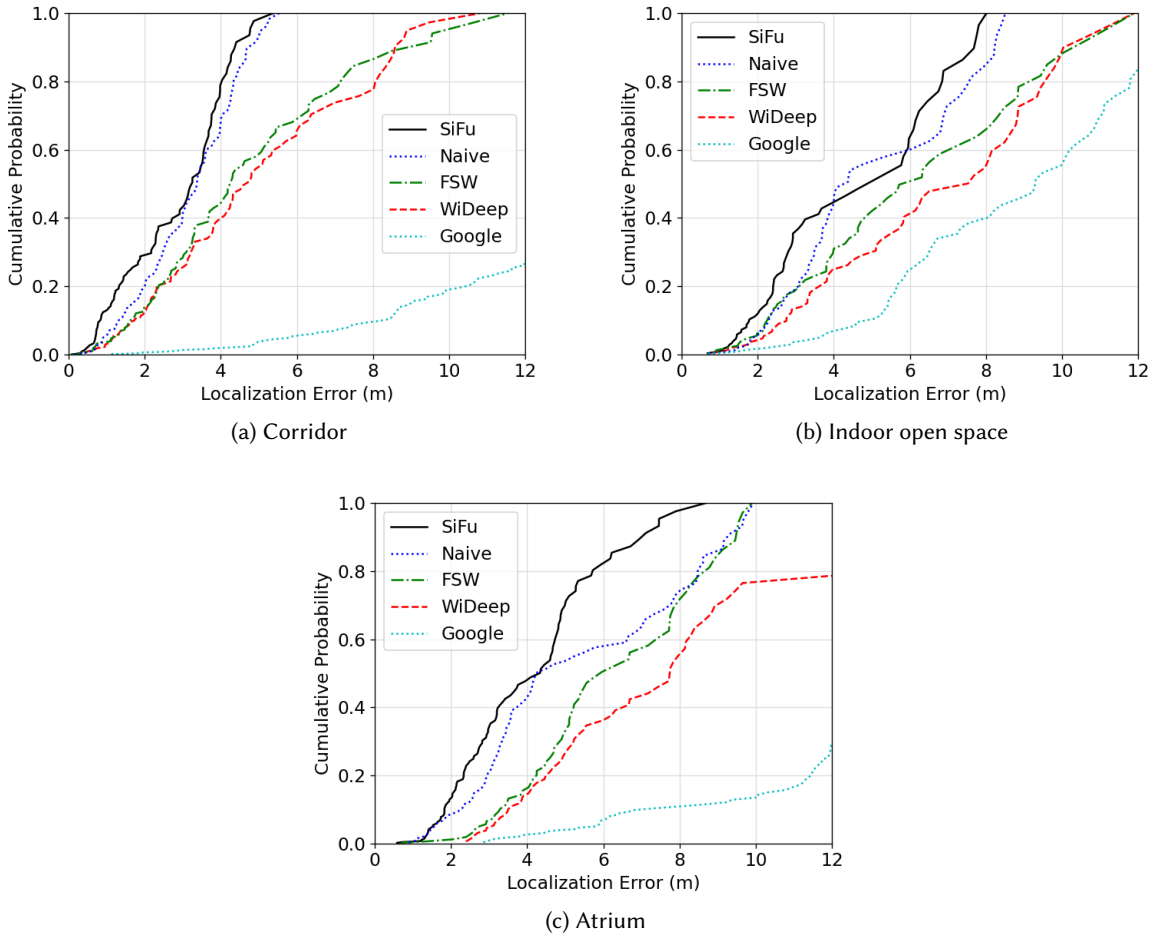


Fig. 6. Mean errors of different schemes in different sites.

much stronger and WiFi signals may fluctuate greatly. Naive methods such as comparing the Euclidean distance and cosine similarity in the raw signal space are evidently affected. Instead, our method that has already learnt to adapt the signal noise and missing values in the training phase is thus more robust and achieves superior accuracy over the other methods.

For fusion localization, we plot in Fig. 6 the mean localization error of SiFu and the compared schemes over three test sites. We notice notable improvements of SiFu over the other state-of-the-art algorithms in all three sites. For WiDeep, the performance is indeed reasonable in corridor area but in indoor open space and atrium regions, its accuracy and stability suffers critically from the noisy WiFi measurements in open areas. As WiDeep only considers single signal for localization, it does not perform well in regions with weak and unstable WiFi signals. In contrast, FSW fuses geomagnetism with WiFi. The negative effects of poor WiFi signals in open areas may be mitigated by the magnetic field, thus it generally works better than WiDeep. However, its fusion method is not ideal as the F-score it introduces does not consider the intrinsic localization errors in complex environments. SiFu improves by fusing more signals and adopting a better fusion approach. Fusing more signals allows us to have more information to make decision and further introducing INS signals and particle filter stabilizes the localization results. Furthermore, compared with the naive version of our system, SiFu takes advantage of the weighted likelihood scheme that alleviates the impacts of bad model performance in some areas, resulting in a even lower localization error. Our proposed method achieves the best mean localization error among the comparison schemes. The mean errors are cut by more than 30% as compared to other state-of-the-art methods.

Figure 7 further plots the cumulative distribution function of the localization errors in different test sites. It is clear that SiFu achieves the most stable and robust performance among all compared schemes. The maximum errors are reduced significantly and a huge portion of the results have acceptable localization errors in all three sites. In Fig. 7a, we observe that the performance of SiFu and its naive version is close. This suggests that the weighted fusion method does not play an important role in the corridor region. This may be because individual models have already worked well and the weight only has limited effects on the shape of the likelihood function. This can also explain why FSW and WiDeep can achieve reasonable accuracy in this site. Figure 7b shows some slight impacts of the weighted likelihood. The maximum error is shrinked while a couple more results have lower localization error than in the naive version. Furthermore, as shown in Fig. 7c, the impact of our proposed likelihood fusion method is the most significant. While there are still a number of results with average accuracy using the naive version, SiFu substantially reduces the error. In atrium, we also notice that WiFi signals may not be ideal for localization as both the localization error and maximum error are large for WiDeep.



(a) Corridor

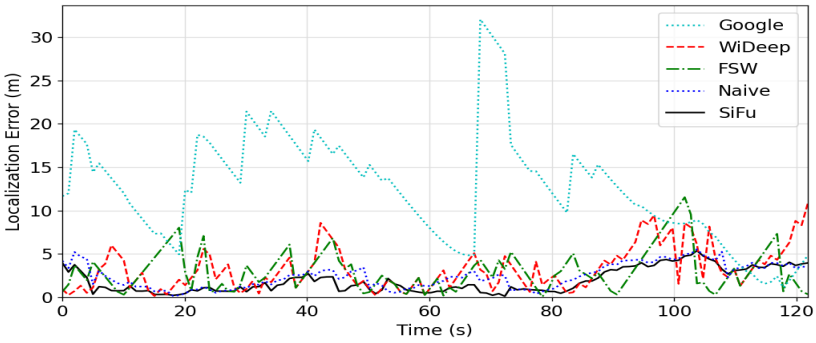
(b) Indoor open space

(c) Atrium

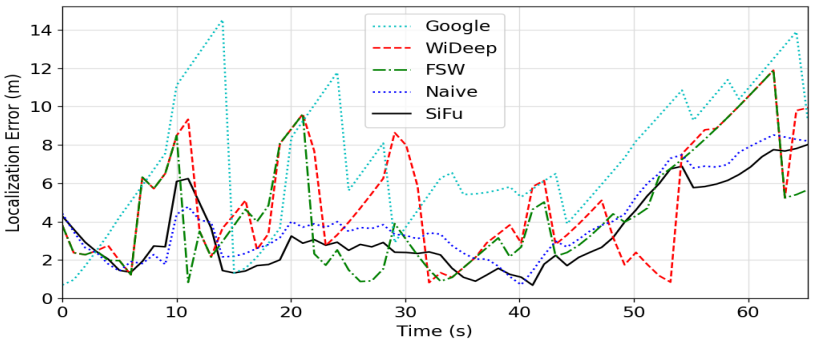
Fig. 7. Cumulative distribution of localization errors in different sites.

In Fig. 8, we also plot the real-time performance of the compared schemes in different sites. Our proposed scheme, which is drawn in solid line, generally achieves stable and satisfactory performance. While the error keeps constantly low, it does not exhibit huge fluctuations either. Being a general localization solution, the Google Fused Location API is not specifically engineered for our testing sites. Hence it is expected that the localization performance is not as good as the other compared schemes requiring site survey. Its result also jumps seriously which undermines user experience. Figure 8a indicates close performance from the state-of-the-art methods in the corridor region. All methods attain generally low errors with acceptable fluctuations. In indoor open space, the results are relatively worse. The localization errors increase considerably near the end of the walk. However, compared with other works, SiFu still achieves the most accurate and stable result.

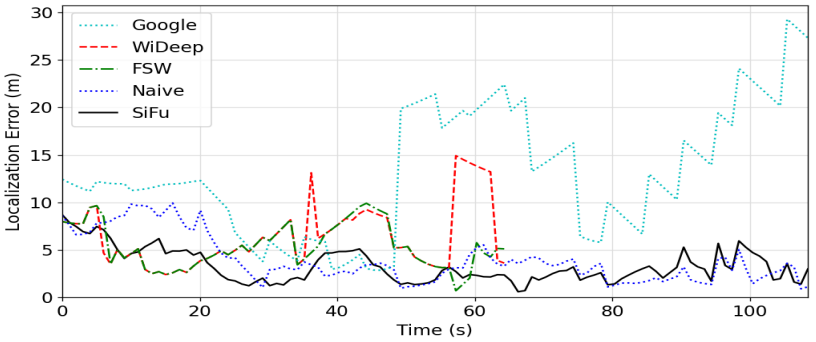
Additionally, we observe a smooth transition from fingerprint region to non-fingerprint region in the atrium area as shown in Fig. 8c. Despite the fact that the signals used for localization change, the localization performance



(a) Corridor



(b) Indoor Open Space



(c) Atrium

Fig. 8. Real-time localization errors in different sites.

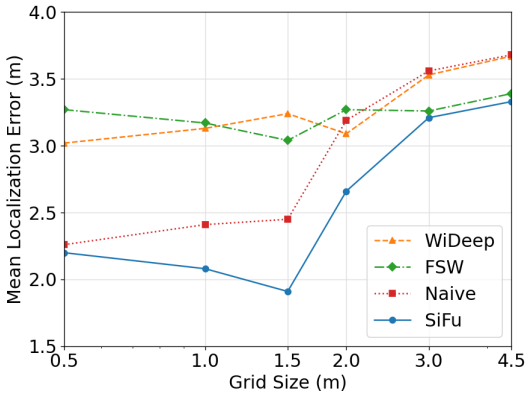


Fig. 9. Mean localization errors with different grid sizes in corridor region.

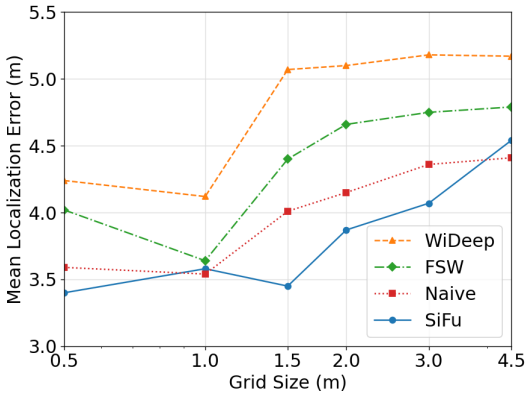


Fig. 10. Mean localization errors with different grid sizes in indoor open space.

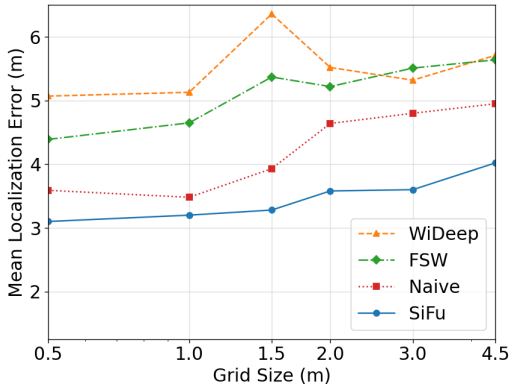


Fig. 11. Mean localization errors with different grid sizes in atrium.

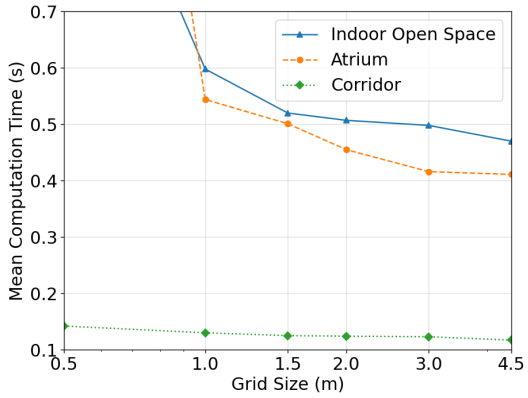


Fig. 12. Computation time with different grid sizes.

is steady that this transition is unnoticeable. This substantiates the ability of SiFu to handle arbitrary signal combinations.

Furthermore, we investigate the impacts of different parameters in our system on the localization performance and computation efficiency. The size of the grid used for computing likelihoods is defined as the distance between adjacent grid points. Figure 9, Figure 10 and Figure 11, plots the mean localization errors with different grid sizes in corridor region, indoor open space and atrium respectively. Overall, we find that the mean localization increases as the grid size increases. Each reference point in the computation grid is associated with a fingerprint signal. The likelihood computed with the fingerprint signal aims to cover the neighbouring region. When the grid size is large, the fingerprint signals may not accurately reflect the signals that can be sampled within the neighbourhood of the grid. Eventually, the likelihood computation would be made incorrect. On the contrary, a denser grid contains more reference points associated with fingerprint signals. This possibly allows better signal

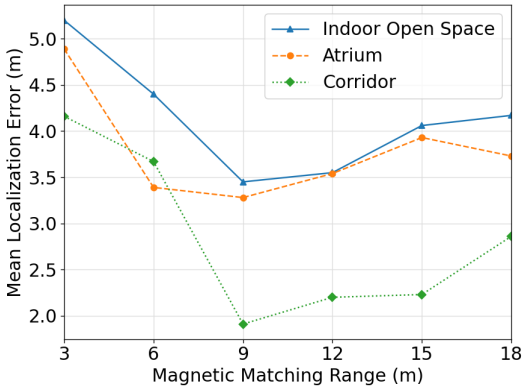


Fig. 13. Mean localization errors in different sites with different magnetic field matching range.

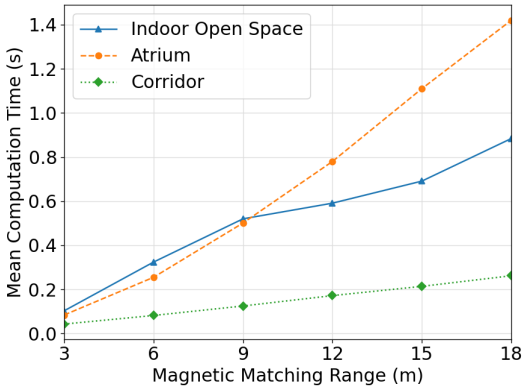


Fig. 14. Computation time with different magnetic field matching range.

differentiation. For example, in the particle filter, particles nearby can have distinct likelihoods when the grid size is small, leading to a more precise distribution.

We also plot the effect of grid size on the computation efficiency in Fig. 12. We observe a decreasing trend of computation time with increasing grid size. It is reasonable because a sparse grid means that we have to compute likelihoods at fewer reference points, and hence less computation is required. Hence, higher accuracy with smaller grid size comes with a trade-off, which is computation efficiency. Furthermore, the result shows the computation complexity of our system is not high, so our system can be easily deployed as a real-time localization system. Even in larger sites, the computation time will not grow significantly as we limit the magnetic field matching space and each module does not require expensive computation.

In Fig. 13, the mean localization errors in different sites with different magnetic matching ranges are shown. A V-shape can be observed from the plot where the localization error becomes larger when the range is too small or too large. As the magnetic matching space is limited by the last predicted location, which is merely an approximate of the true last user location, the true current locations may not be covered in the matching space when the range is too small. Then the magnetic field at the true location will be ignored in the sequence matching procedure and the result will not be valid. On the other hand, if the range is too large, ambiguity in magnetic sequences within the matching region may exist, which makes the likelihood computation inaccurate. In our experiments, the optimal range is around $9m$, which can accommodate the localization error while limiting ambiguity in the search space.

Figure 14 plots the mean computation time with different magnetic field matching range. We see a linearly increasing trend in computation time as the matching range increases. It suggests that the DTW algorithm for magnetic field localization in our system is rather expensive. When the matching space is large and we need to match against more magnetic field fingerprints, the computation time increases drastically. To make our system real-time enough, the range should be limited appropriately. Considering the localization accuracy, the default is set to $9m$.

Figure 15 plots the mean localization errors in different sites with different σ_m. Here we observe a V-shape. The sensitivity parameter σ_m for the observation model for magnetic field decides how much noise can be tolerated. If it is too large, the likelihood cannot effectively discriminate locations because dissimilar signals will still get a rather high likelihood. If it is too small, the likelihood will be low even if the sampled signal deviates a bit from

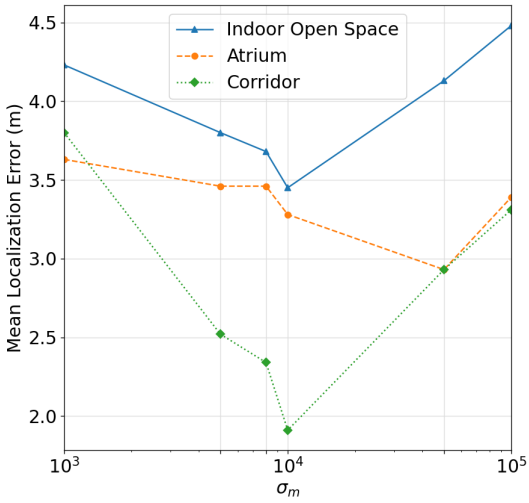


Fig. 15. Mean localization errors in different sites with different σ_m.

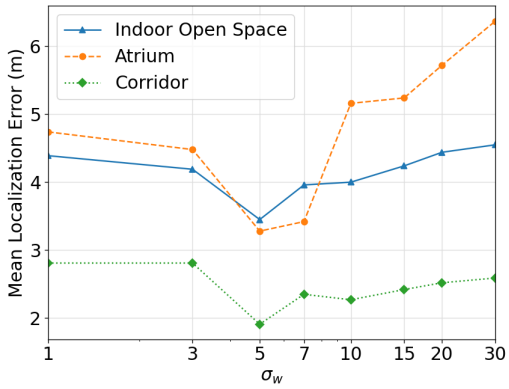


Fig. 16. Mean localization errors in different sites with different σ_w.

the fingerprint signal. As the noise tolerance is low, signal noise cannot be handled properly. This implies that the sensitivity parameter σ_m cannot be too large or too small in order to obtain a legit location estimate.

Also, the impact of σ_w on the localization performance in different sites is demonstrated in Fig. 16. A similar V-shape can be observed. This reinforces the results in the experiment for σ_m. Indeed, σ_m and σ_w are two important parameters in our system and our experimental results explain our default settings of these two parameters.

Finally, we experiment on the localization performance with sparse WiFi sampling. Figure 17 shows the mean errors of SiFu and the compared scheme in different test sites. The results are consistent with the dense WiFi sampling situation. With sparser WiFi sampling, we observe that the mean errors generally increase compared to the situation where dense WiFi signals are available. In the meantime, the increase of the error of SiFu is much lower than that of the compared schemes. It remains the best localization system among the compared schemes. Furthermore, we notice again that the weighted scheme in SiFu boosts the localization performance over the naive version of SiFu.

Figure 18, Figure 19 and Figure 20 plot the cumulative distribution of the localization errors in different sites in the sparse WiFi sampling case. WiDeep that uses only WiFi performs poorly as WiFi signals are less frequently available for localization. FSW, despite considering magnetic field as well, only performs slightly better than WiDeep. This is probably because it does not work when some signals, such as WiFi in this case, are unavailable. When WiFi signal is unavailable, only using magnetic field may not give a desirable performance due to the ambiguity of magnetic field. In constrast, SiFu fuses more signals and integrate INS signals. User location can still be approximated accurately even though some signals are missing. Similar to the dense WiFi sampling case, a majority of the results have satisfactory localization errors.

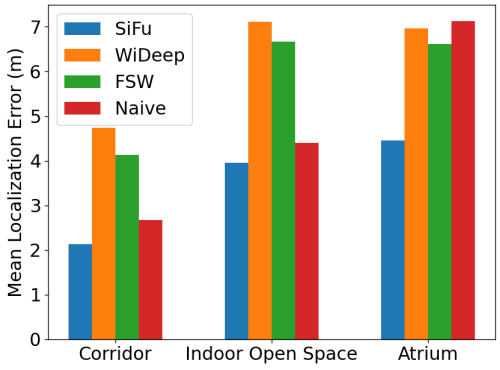


Fig. 17. Mean error of different schemes in different sites with with sparse WiFi sampling.

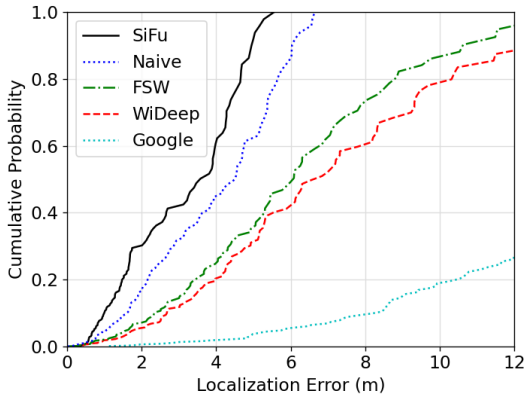


Fig. 18. Cumulative distribution of localization errors in corridor region with sparse WiFi sampling.

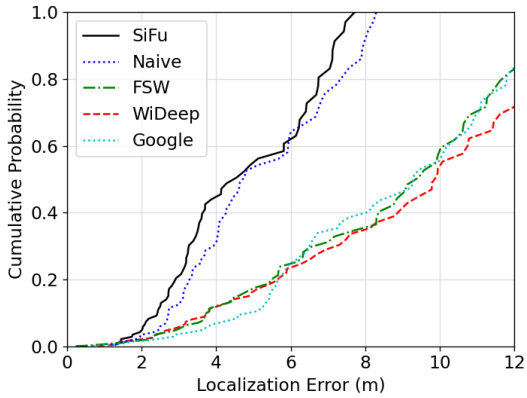


Fig. 19. Cumulative distribution of localization errors in indoor open space with sparse WiFi sampling.

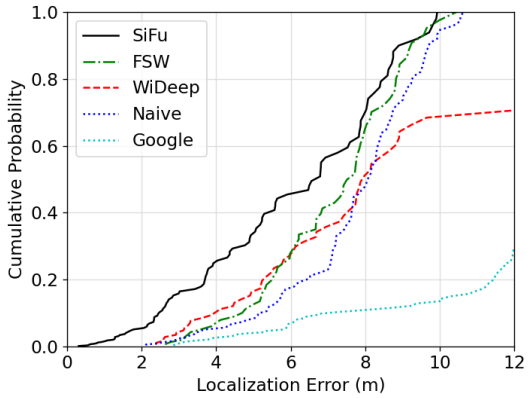


Fig. 20. Cumulative distribution of localization errors in atrium with sparse WiFi sampling.

7 CONCLUSION

Signal fusion is able to achieve higher localization accuracy by mitigating the weaknesses of individual signals. Previous works in the area are often highly customized to the types and few number (two or three) of signals. To overcome this, we propose SiFu, a generic, probabilistic and accurate signal fusion framework to arbitrarily combine many multi-modal signals of heterogeneous sampling rate. Signals may be dynamically added or removed in SiFu, hence supporting seamless roaming of users between markedly different indoor and outdoor environments. SiFu is a platform which can embrace new signals at any time without retraining the whole system.

In SiFu, each signal is processed independently through its corresponding likelihood computation module to calculate likelihood at each grid point in the map. We have presented a novel likelihood convertor for RSSI vectors based on latent representation of the signals learnt from a denoising autoencoder. The likelihoods of the signals at each grid point are then fused together using a novel method based on the weighted likelihood framework in

Bayesian analysis, where the weight at a grid point of each signal is assigned according to the accuracy in the training process. The result is further combined with INS signals through particle filter. Compared with other state-of-the-art localization algorithms, SiFu achieves much higher elasticity in signal composition and deployability in different environments. Extensive experimental studies show that SiFu outperforms state-of-the-art schemes in terms of localization accuracy by a large margin (more than 30% on average in our experiments).

REFERENCES

[1] Moustafa Abbas, Moustafa Elhamshary, Hamada Rizk, Marwan Torki, and Moustafa Youssef. 2019. WiDeep: WiFi-based accurate and robust indoor localization system using deep learning. In *IEEE PerCom*.

[2] Claudio Agostinelli and Luca Greco. 2013. A weighted strategy to handle likelihood uncertainty in Bayesian inference. *Computational Statistics* 28, 1 (2013), 319–339.

[3] Martin Azizyan, Ionut Constandache, and Romit Roy Choudhury. 2009. SurroundSense: mobile phone localization via ambience fingerprinting. In *Proceedings of the 15th annual international conference on Mobile computing and networking*. ACM, 261–272.

[4] P Bahl and VN Padmanabhan. 2000. RADAR: an in-building RF-based user location and tracking system. In *Proceedings of the Nineteenth Annual Joint Conference of IEEE Computer and Communications Societies (INFOCOM)*. IEEE, 775–784.

[5] Sinem Bozkurt Keser, Ahmet Yazici, and Serkan Gunal. 2018. An F-Score-Weighted Indoor Positioning Algorithm Integrating WiFi and Magnetic Field Fingerprints. *Mobile Information Systems* 2018 (2018).

[6] K. Chow, S. He, J. Tan, and S. . G. Chan. 2019. Efficient Locality Classification for Indoor Fingerprint-Based Systems. *IEEE Transactions on Mobile Computing* 18, 2 (2019), 290–304.

[7] Gunter Fischer, Burkhart Dietrich, and Frank Winkler. 2004. Bluetooth indoor localization system. In *Proceedings of the 1st Workshop on Positioning, Navigation and Communication*. 147–156.

[8] Xumeng Guo, Wenhua Shao, Fang Zhao, Qu Wang, Dongmeng Li, and Haiyong Luo. 2016. WiMag: Multimode fusion localization system based on Magnetic/WiFi/PDR. In *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 1–8.

[9] Suining He and S-H Gary Chan. 2015. Wi-Fi fingerprint-based indoor positioning: Recent advances and comparisons. *IEEE Communications Surveys & Tutorials* 18, 1 (2015), 466–490.

[10] F. Hong, Y. Zhang, Z. Zhang, M. Wei, Y. Feng, and Z. Guo. 2014. WaP: Indoor localization and tracking using WiFi-Assisted Particle filter. In *39th Annual IEEE Conference on Local Computer Networks*. 210–217.

[11] Sinem Bozkurt Keser, Ahmet Yazici, and Serkan Günal. 2017. A hybrid fingerprint based indoor positioning with extreme learning machine. In *2017 25th Signal Processing and Communications Applications Conference (SIU)*. IEEE, 1–4.

[12] Binghao Li, Thomas Gallagher, Andrew G Dempster, and Chris Rizos. 2012. How feasible is the use of magnetic field alone for indoor positioning?. In *2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 1–9.

[13] You Li, Yuan Zhuang, Peng Zhang, Haiyu Lan, Xiaoji Niu, and Naser El-Sheimy. 2017. An improved inertial/wifi/magnetic fusion structure for indoor navigation. *Information Fusion* 34 (2017), 101–119.

[14] Zhenguang Liu, Luming Zhang, Qi Liu, Yifang Yin, Li Cheng, and Roger Zimmermann. 2016. Fusion of magnetic and visual sensors for indoor localization: Infrastructure-free and more effective. *IEEE Transactions on Multimedia* 19, 4 (2016), 874–888.

[15] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan. 2011. Estimation of IMU and MARG orientation using a gradient descent algorithm. In *2011 IEEE International Conference on Rehabilitation Robotics*. 1–7.

[16] Michael A Newton and Adrian E Raftery. 1994. Approximate Bayesian inference with the weighted likelihood bootstrap. *Journal of the Royal Statistical Society: Series B (Methodological)* 56, 1 (1994), 3–26.

[17] Stan Salvador and Philip Chan. 2007. Toward Accurate Dynamic Time Warping in Linear Time and Space. *Intelligent Data Analysis* 11, 5 (2007), 561–580.

[18] Yuanchao Shu, Cheng Bo, Guobin Shen, Chunshui Zhao, Liqun Li, and Feng Zhao. 2015. Magicol: Indoor localization using pervasive magnetic field and opportunistic WiFi sensing. *IEEE Journal on Selected Areas in Communications* 33, 7 (2015), 1443–1457.

[19] Q. Tian, Z. Salcic, K. I. Wang, and Y. Pan. 2016. A Multi-Mode Dead Reckoning System for Pedestrian Tracking Using Smartphones. *IEEE Sensors Journal* 16, 7 (2016), 2079–2093.

[20] Quoc Duy Vo and Pradipta De. 2015. A survey of fingerprint-based outdoor localization. *IEEE Communications Surveys & Tutorials* 18, 1 (2015), 491–506.

[21] Hang Wu, Suining He, and S-H Gary Chan. 2017. Efficient sequence matching and path construction for geomagnetic indoor localization. In *Proceedings of the 2017 International Conference on Embedded Wireless Systems and Networks*. Junction Publishing, 156–167.

[22] Han Xu, Zheng Yang, Zimu Zhou, Longfei Shangguan, Ke Yi, and Yunhao Liu. 2016. Indoor localization via multi-modal sensing on smartphones. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 208–219.

[23] Moustafa Youssef and Ashok Agrawala. 2005. The Horus WLAN location determination system. In *Proceedings of the 3rd international conference on Mobile systems, applications, and services*. ACM, 205–218.

[24] Chi Zhang and Xinyu Zhang. 2016. LiTell: robust indoor localization using unmodified light fixtures. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*. ACM, 230–242.
[25] Mei Zhang, Wenbo Shen, and Jinhui Zhu. 2016. WiFi and magnetic fingerprint positioning algorithm based on KDA-KNN. In *2016 Chinese Control and Decision Conference (CCDC)*. IEEE, 5409–5415.
[26] Ran Zhang, Wen-De Zhong, Dehao Wu, and Kemao Qian. 2016. A novel sensor fusion based indoor visible light positioning system. In *2016 IEEE globecom workshops (GC wkshps)*. IEEE, 1–6.