Taming the Inconsistency of Wi-Fi Fingerprints for Device-Free Passive Indoor Localization

Xi Chen[1], Chen Ma[1], Michel Allegue[2], and Xue Liu[1]
[1]School of Computer Science, McGill University, Canada,
{xi.chen11, chen.ma2}@mail.mcgill.ca, xueliu@cs.mcgill.ca
[2]Aerial, Canada, michel.allegue@aerial.ai

***Abstract*—Device-free Passive (DfP) indoor localization releases the users from the burden of wearing sensors or carrying smartphones. Instead of locating devices, DfP technology directly locates human bodies. This promising technology upgrades and even redefines many services, such as intruder alarm, fire rescue, fall detection, baby monitoring, etc. Using Wi-Fi based fingerprints, DfP approaches can achieve a nearly perfect accuracy with a resolution less than one meter. However, Wi-Fi localization profiles may easily drift with a minor environment change, resulting in an inconsistency between fingerprints and new profiles. This inconsistency issue could lead to large errors, and may quickly ruin the whole system. To address this issue, we propose a approach named AutoFi to automatically calibrate the localization profiles in an unsupervised manner. AutoFi embraces a new technique that online estimates and cancels profile contaminants introduced by environment changes. It applies an autoencoder to preserve critical features of fingerprints, and reproduces them later in new localization profiles. Experiment results demonstrate that AutoFi indeed rescues the Wi-Fi fingerprints from variations in the surrounding. The localization accuracy is improved from 18.8% (before auto-calibration) to 84.9% (after auto-calibration).**

I. Introduction

Indoor localization has the potential to change the way people navigate, work and live indoors. It enables numerous services, including wireless intruder detection, elder and baby monitoring, home automation, etc. To support these services, radio signals, laser beams, magnetic fields, acoustic waves, accelerometer readings, and other sensory information have been utilized. Existing localization approaches can be divided into two categories, device-oriented approaches and device-free approaches. Device-oriented approaches locate the users by positioning their wearable sensors or mobile devices. The resolution of this category is at the decimeter level [1], [2]. However, a person cannot be located, if she/he is not carrying any device (e.g., someone taking a shower) or is not connected to the localization systems (e.g., an intruder).

On the contrary, device-free approaches are able to detect the physical existence of people. It frees the systems and their users from the constraints of wearable sensors and mobile devices, and thus enables Device-free Passive (DfP) localization services. To achieve this, device-free approaches detect and analyze the distortions of electromagnetic fields introduced by human bodies. They then correlate these distortions with different locations to establish localization fingerprints. A number of electromagnetic characteristics have been applied and combined to generate a robust fingerprint. Among them, Wi-Fi Received Signal Strength Indicator (RSSI) and Wi-Fi Channel State Information (CSI) are the most widely adopted ones. Systems based on Wi-Fi characteristics have several distinct benefits including 1) easy deployment given the ubiquitous Wi-Fi infrastructure and 2) small overhead by recycling Wi-Fi traffic already in the air. Currently, approaches utilizing Wi-Fi CSI are able to achieve a resolution at the sub-meter level [3], [4].

However, a major issue of DfP localization remains unsolved. Propagation of wireless signals is sensitive to not only human bodies but also other objects in the environment. A small change, such as opening a window or closing a door, may significantly change the Wi-Fi CSI and other features. In this case, the newly collected localization profiles become inconsistent with previously stored fingerprints, and are considered as contaminated in this paper. This inconsistency problem requires the localization systems to frequently collect new profiles and retrain the localization models, potential leading to large system downtime and maintenance cost.

To address this problem, we develop an auto-calibration approach AutoFi that automatically calibrates the system by purifying the contaminated profiles. This approach first estimates the contaminants brought by environment changes. It then eliminates these contaminants with a linear regression module from newly collected localization profiles. Moreover, an autoencoder [5] is applied after this module to further preserve profile features robust to the varying environment. We conducted experiments in residential apartments to demonstrate the effectiveness of AutoFi. After we changed the environment, the contaminated profiles yield an accuracy as low as 18.8% at the meter level. AutoFi is able to rescue the system by improving the accuracy to 84.9% while maintaining the resolution at the sub-meter level.

The **main contribution** of this paper is two-fold.

- We are the first to study the inconsistency issue of Wi-Fi features in DfP indoor localization. We investigate its major cause - the changes in the physical environment, and illustrate its large impacts on the accuracy of DfP localization systems.
- To address this issue, we develop an auto-calibration approach AutoFi that embraces a novel contaminant removal module and an autoencoder to calibrate the Wi-Fi profiles. Real-life experiments demonstrate that AutoFi is able to save the localization system by improving its accuracy from 18.8% to 84.9% at the sub-meter level.

II. Related Work

In this section, we review the most related work to ours, and discuss how this paper advances the research area.

A. Device-Oriented Approaches

Device-oriented approaches require users to carry certain portable devices (e.g. sensors and smartphones.). The state of the art in this category can achieve a decimeter resolution.

1) RSSI-Based: The pioneering work adopted RSSI of wireless signals to estimate distances between reference devices and client devices. For example, Sugano et al. in [6] estimated the distance between sensor nodes by measuring the RSSI of ZigBee signals. Mazuelas et al. in [7] applied Wi-Fi RSSI to calculate the distance between a mobile station and an Access Point (AP), with an online estimated propagation model. Yet, the resolution of RSSI-based approaches is not satisfactory. Other metrics are adopted to further improve the localization performance.

2) ToF-Based: Compared with RSSI, Time of Flight (ToF) provides a more accurate estimation of distance, and thus is used more recently in device-oriented approaches. For example, Liu et al. in [8] deployed a smartphone-based system, which performs Non-Line-Of-Sight (NLOS) identification to obtain a fine-grained ToF using acoustic beacon signals. Adib et al. in [2] proposed WiTrack2.0, which pinpoints multiple users locations based the ToF of radio signals reflected off human bodies. Vasisht et al. in [9] applied a frequency mitigation technique to utilize Wi-Fi's 2.4 GHz and 5 GHz spectrums as a whole in combating multipath effect, and achieved sub-nanosecond ToF with commodity Wi-Fi cards.

3) AoA-Based: The Angle of Arrival (AoA) based approaches measure the angle from a client device to multiple reference devices, so as to locations the client device by triangulation techniques. For example, Xiong et al. in [10] designed a powerful client-side antenna array, ArrayTrack, which estimates the AoA spectrum of wireless clients for localization and tracking purposes. Gjengset et al. in [11] developed ArrayPhaser that upgrades a multi-antenna commodity WiFi AP to a phased array signal processing platform for AoA estimation. Kumar et al. in [12] presented LTEye which uses the LTE PHY information to locate mobile users through AoA between LTE-enabled phones and cellular stations. To avoid using a complicated antenna array, Kumar et al. in [13] proposed Ubicarse that facilitates handhold Wi-Fi devices to emulate large antenna arrays using a Synthetic Aperture Radar (SAR) technique. Kotaru et al. in [1] developed SpotFi that incorporates an advanced MUSIC-AoA algorithm to thoroughly eliminate multipath components in AoA estimation, and thus achieved a decimeter level resolution.

4) Fingerprint-Based: The fingerprint-based approaches compare the current features of ambient signals with those already recorded in a fingerprint database, so as to correlate the newly collected signals to pre-defined locations. For example, Wu et al. in [14] explored the frequency diversity of the subcarriers in OFDM systems and leveraged the Wi-Fi CSI to build a fingerprinting system. Nandakumar et al. [15] designed Centaur, which fuses Wi-Fi RSSI and acoustic signals to create localization fingerprints for smartphones. To reduce the effort of collecting fingerprints, Yang et al. [16] leveraged a combination of accelerometer readings and Wi-Fi signals to construct fingerprints in a semi-unsupervised manner. Rai et al. [17] developed a crowdsourcing technique Zee to gather Wi-Fi fingerprints in the area of interest.

Essentially, the device-oriented approaches are localizing sensors and devices instead of the users. They are not suitable in scenarios where the user is not with any device or the device is simply not connected to the system. For applications such as fall detection in the bathroom, baby monitoring and intruder detection, we need the help of device-free approaches.

B. Device-Free Approaches

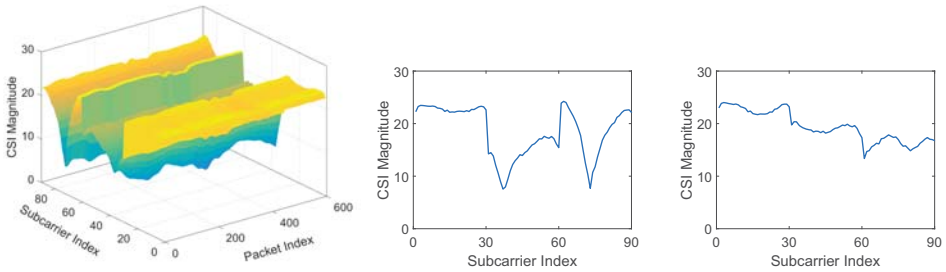
Device-free approaches release the users from the constraint of carrying devices. They passively detect the signal distortions caused by the physical existence of human bodies, and further infer human locations with this information.

1) RFID-Based: RFIDs have also been applied in device-free approaches, mainly due to their cost effectiveness. For example, Han et al. [18] developed a passive localization system leveraging the critical state phenomenon caused by interference among adjacent RF tags. Yang et al. [19] presented Tadar that transforms a group of tags into an antenna array to receive the reflections from people for user tracking. However, deploying RF tags requires a lot of efforts. A more promising way is to adopt the existing infrastructure, such as the Wi-Fi networks.

2) Wi-Fi-Based: A large body of device-free approaches utilizes the ubiquitous Wi-Fi signals. For instance, Moussa et al. in [20] employed Wi-Fi RSSI to detect whether a person enters the area of interest, and presented an maximum likelihood estimator (MLE) algorithm to enhance system robustness. Seifeldin et al. in [21] also utilized Wi-Fi RSSI as fingerprints to localize a person to one of the fingerprinted locations. Xu et al. in [22] modeled indoor human trajectories as a state transition process, and integrated all Wi-Fi RSSI information into a Conditional Random Field (CRF) to simultaneously localize multiple people.

Approaches utilizing Wi-Fi CSI fingerprints generally achieve better performance than those based on RSSI fingerprints. For example, Sen et al. in [3] developed PinLoc that achieves a 90% localization accuracy by utilizing Wi-Fi CSI from multiple OFDM subcarriers as location fingerprints. Xiao et al. in [4] demonstrated that passive CSI fingerprints can outperform RSSI fingerprints by at least 10%. Chapre et al. in [23] utilized a Multiple Input Multiple Output (MIMO) system to collect and fuse a large number of CSI values for improved accuracy and resolution. Wang et al. in [24] presented a novel deep learning approach to further increase the performance of CSI fingerprint based system. Ghourchian et al. in [25] adopted a semi-supervised method for real-time indoor localization with CSI fingerprints.

However, RSSI and CSI values are easily altered by the environment changes. A previously established fingerprint database can become out-of-date with a simple action, e.g., opening a window or moving an office chair. Collecting new fingerprints and retraining the system will cost unacceptable system downtime and a huge amount of supervised work. Therefore, we are still in need of an approach to auto-calibrate the system in the ever-changing environment.



(a) A series of CSI profiles at location L. (b) A CSI fingerprint profile at location L. (c) A contaminated CSI profile at location L when a window is open.

Fig. 1: Example CSI profiles recorded at a residential apartment.

III. Background and Definitions

In this section, we present the background, along with the definition of several important concepts.

A. CSI Profiles and Fingerprints

The CSI describes how the Wi-Fi signals propagate from the transmitter to the receiver through the wireless channel. It is basically a mathematical representation of the channel, reflecting the intertwined effects of fading, scattering, shadowing, power decaying, etc. The Orthogonal Frequency-Division Multiplexing (OFDM) method used by Wi-Fi divides the channel into multiple subcarriers, each of which has its own **CSI value** $h_{i,j}$, where i denotes the subcarrier index, and j denotes the packet index. These CSI values are estimated at the receiver based on the pilots inserted in the each transmitted Wi-Fi packet. A CSI value is a complex number, of which the magnitude is adopted to create CSI profiles and fingerprints in this paper.

***Definition** 1:* A **CSI profile** $G(p)$ is defined as the vector of CSI magnitudes measured via a Wi-Fi packet p, i.e.,

$$G(p) = [|h_{1,p}|, |h_{2,p}|, \cdots, |h_{N_{SC},p}|], \tag{1}$$

where N_{SC} denotes the number of subcarriers.

During the measurement, the system records a series of CSI profiles at the same location to build a robust fingerprint. Figure 1(a) presents examples of CSI profiles collected at a location L in a residential apartment.

***Definition** 2:* A **CSI fingerprint** F_L is a combination of a location label L and a CSI fingerprint profile G_L when a person is at location L, i.e.,

$$F_L = \{L; G_L\}. \tag{2}$$

A **CSI fingerprint profile** is the average of N CSI profiles $\{G_L(p_1), G_L(p_2), \cdots G_L(p_N)\}$ recorded when a person is at location L, i.e.,

$$G_L = \left[\frac{1}{N}\sum_{i=1}^{N}|h_{1,i}|, \frac{1}{N}\sum_{i=1}^{N}|h_{2,i}|, \cdots, \frac{1}{N}\sum_{i=1}^{N}|h_{N_{SC},i}|\right]. \tag{3}$$

Figure 1(b) gives an example CSI fingerprint profile. It is the average of profiles illustrated in Figure 1(a), and is stored in the fingerprint database. The system collects a certain number of CSI fingerprints to establish a fingerprint database for localization.

B. The Inconsistency Issue

However, these collected fingerprints are not valid forever, as the Wi-Fi signal propagation is sensitive to both human bodies and objects in **the area of interest A**. The Wi-Fi features may vary significantly with small environment changes (even if the user stays still). Such changes can happen naturally from time to time in our daily lives, such as opening a window in the living room or closing the door of a bedroom. When these changes happen, the new CSI profiles will become inconsistent with those already stored in the fingerprints, leading to potentially large localization errors. We call this phenomenon as the **inconsistency issue** of Wi-Fi fingerprints, and consider the new profiles as being contaminated.

***Definition** 3:* At a certain location L, a **contaminant** C_L is the difference between the profiles before and after an environment change, i.e.,

$$C_L = G_L^{pre} - G_L^{post}, \tag{4}$$

where G_L^{pre} is the fingerprint profile in the fingerprint database (before the environment change), and G_L^{post} is the contaminated profile (after the environment change).

Figure 1(c) presents an example of contaminated profiles, which is generated by opening a window of the apartment. Comparing Figure 1(b) and Figure 1(c), it is clear that the CSI profile has been changed significantly by the opened window. In other words, the current CSI profile at this location is inconsistent with the one in the fingerprint database. If the system feeds this contaminated profile into the system, the localization result will very likely be wrong.

A straightforward solution is to collect the fingerprints all over again. However, this costs large system downtime, and can be wasted again by another minor environment change, e.g., opening a door.

A better way to solve this issue is to reconstruct the original CSI profiles from the contaminated ones in an unsupervised manner. In this way, the fingerprint database and the trained localization model can still be functional without a re-initialization. This requires an auto-calibration to remove the contaminants.

In order to accomplish this mission, we propose an auto-calibration approach AutoFi in this paper. AutoFi estimates the contaminants by analyzing the variations in CSI profiles at a reference location.

***Definition** 4:* A **reference location** $L0$ is a location used

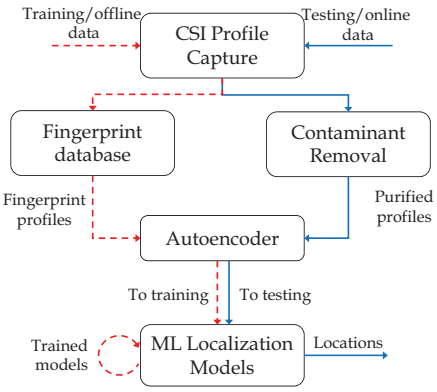


Fig. 2: The architecture of AutoFi.

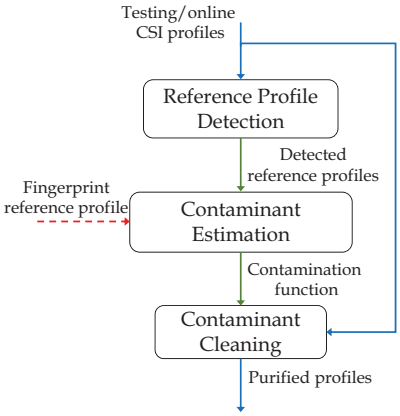


Fig. 3: The contaminant removal process.

to compare CSI profiles G_{L0}^{pre} and G_{L0}^{post} before and after the environment changes. CSI profiles corresponding to a reference location are called **reference profiles**.

If a reference profile is stored in the fingerprint database (e.g., G_{L0}^{pre}), we call it a fingerprint reference profile. If a reference profile is detected during the testing phase (e.g., G_{L0}^{post}), we call it a detected reference profile.

A valid candidate of reference location must be easily localized online in an unsupervised way. As aforementioned, the contaminants are estimated by comparing a fingerprint reference profile and a detected reference profile. Any online profiles will then be purified by removing the estimated contaminants. In Section V, we will discuss how to selection a reference location offline and detect it online. We will also present the estimation and removal of contaminants there. In addition to this contaminant removal, we will also apply an autoencoder to preserve the profile features for better localization accuracy in Section VI.

IV. System Overview

In this section, we present the overall design of AutoFi, and discuss how AutoFi works in both training and testing phases.

A. System Overview

AutoFi consists of a CSI profile capture module, a fingerprint database, a contaminant removal module, a feature-preserving autoencoder, and a localization module with Machine Learning (ML) models, as illustrated in Figure 2. Dash lines denote the data flow during the training phase, while solid lines represent the data flow during the testing phase.

B. Training Process

During the training (offline) phase, raw CSI data sets are collected at different locations, together with location labels $\{L1, L2, \cdots\}$. These data sets are then fed into the CSI profile capture module, which extracts the CSI values and calculates their magnitudes. The outputs of this module are extracted CSI profiles, i.e., $\{G_{L1}(p_1), G_{L1}(p_2), \cdots\}, \{G_{L2}(p_1), G_{L2}(p_2), \cdots\}, \cdots$. These profiles are integrated with their corresponding location labels to build CSI fingerprints, i.e., $F_{L1}, F_{L2}, \cdots$, which are then stored in the fingerprint database. An autoencoder is trained with the stored fingerprints without location labels. In this way, the general features of all CSI profiles (under the same environment condition) are preserved by the autoencoder's neurons. Finally, the localization module utilize the fingerprints to train its ML models. Note that the design of an advanced ML localization algorithm is not the purpose of this paper.

C. Testing Process

During the testing (online) phase, each newly collected CSI data point also goes through the profile capture module first. The output profile $\dot{G}$ is passed to the contaminant removal module for estimating the environment contaminants and purifying the profile. The autoencoder then propagates this purified profile $\hat{G}$ through its neurons. The output of this autoencoder is a reproduced CSI profile $\tilde{G}$, which embeds the critical features of the fingerprint profiles. The localization module then uses this reproduced CSI profile $\tilde{G}$ to estimate the user's location.

The development of CSI profile capture module, fingerprint database, and ML localization module, follows regular practice in the academia and industry (e.g., those in [3], [4], [24]). Thus, we skip the detailed design of them. Next, we focus on the new designs in AutoFi, i.e., the contaminant removal module and the feature-preserving autoencoder.

V. Contaminant Removal

In this section, we present the detailed design of the contaminant removal module.

The whole contaminant removal process is illustrated in Figure 3. The testing/online profiles are the inputs to this module, and the purified profiles are the outputs. The module first detects whether the incoming profile is a reference profile for the reference location $L0$ (as defined in Section III-B). If the answer is affirmative, then this detected reference profile G_{L0}^{D} is compared with a fingerprint reference profile G_{L0}^{F} to estimate the contaminants. A contaminant is represented by a set of contamination functions $W_{L0}(\cdot)$, which describes how G_{L0}^{F} is reshaped into G_{L0}^{D} by the environment changes. $W_{L0}(\cdot)$ is then used to cancel the effect of contaminant from online profiles (including both detected reference profiles and other online profiles). Next, we discuss the details of this contaminant removal module step by step.

A. Reference Profile Detection

1) Selecting a Reference Location Offline: A candidate reference location needs to be detected online with an unsupervised method. There should be no manual calibration nor user inputs. The existence of such a reference location has also been utilized in previous approaches (e.g, [26]). However, there is no guarantee that such a location always exists.

In this paper, we adopt a virtual location $L0$ to represent the case where the area of interest A is empty (i.e., no person in the area). Therefore, this virtual location is guaranteed to exist, and can be used as the reference location. The detection of the reference profile G_{L0} is equivalent to the detection of whether A is empty. The contaminant is then estimated by comparing the profiles of an empty area before and after the environment changes.

2) Detecting a Reference Profile Online: The reference (empty) detection is realized by analyzing the profile variations. This is inspired by the following observation. If a person is in the area A, even she is simply standing, the profile variation will be much larger than the case of empty. An example is illustrated in Figure 4.

The empty detection is described as Algorithm 1. Algorithm 1 confirms the case of empty, if the average variance of all subcarriers' magnitudes is lower than a threshold. This threshold is determined as follows. Given a set of fingerprint reference profiles $\{G_L0(p_1), G_L0(p_2), \cdots\}$, we first calculate their average variances $\{\bar{v}(1), \bar{v}(2), \cdots\}$ of all subcarriers. We then extract their maximum $\bar{v}_{max} = \max\{\bar{v}(1), \bar{v}(2), \cdots\}$, and determine the variance threshold as $v_{th} = (1+\beta)\bar{v}_{max}$, where β denotes a margin for robustness. Empirically, a safe choice of β can be any real number in $[0, 1]$. In this paper, we set $\beta = 0.5$, and thus $v_{th} = 1.5 \cdot \bar{v}_{max}$.

Algorithm 1 Reference Profile (Empty) Detection

Input: A newly collected CSI profile $\dot{G}$.
Output: A boolean variable E denoting whether $\dot{G}$ is a reference (empty) profile.
Constants: A buffer B of length N_B, a variance vector V of length N_{SC}, and a variance threshold v_{th}.

1. Put $\dot{G}$ into B.
2. if $len(B) \geq N_B$
3. for j in $[0, N_{SC}-1]$
4. $g_{temp} = \{|h_{j,1}|, |h_{j,2}|, \cdots, |h_{j,N_B}|\}$.
5. $V[j] = Variance(g_{temp})$.
6. $\bar{v} = Sum(V)/N_{SC}$.
7. if $\bar{v} < v_{th}$
8. $E = True$.
9. else
10. $E = False$.
11. Drop the oldest elements in B so that $len(B) = N_B - 1$.
12. else
13. $E = False$.

B. Contaminant Estimation

If the newly collected CSI profile G is detected as a reference profile, then we denote it as G^D_{L0}. We then compare

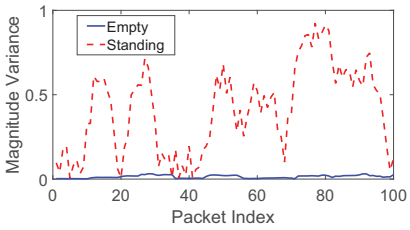


Fig. 4: Magnitude variances, empty area versus one person standing in the area.

G^D_{L0} with the fingerprint reference profile G^F_{L0} to estimate the contaminants. By Eq. (1), we rewrite these two profiles as:

$$G^F_{L0} = \left[|h^F_1|, |h^F_2|, \cdots, |h^F_{N_{SC}}|\right], \quad (5)$$

$$G^D_{L0} = \left[|h^D_1|, |h^D_2|, \cdots, |h^D_{N_{SC}}|\right]. \quad (6)$$

We model the effect of the contaminants as a set of contamination functions $W_{L0} = \{w_1(\cdot), w_2(\cdot), \cdots, w_{N_{SC}}(\cdot)\}$, each of which describes how $|h^F_i|$ is reshaped into $|h^D_i|$ by the environment changes, i.e.,

$$|h^D_i| = w_i\left(|h^F_i|\right), i = 1, 2, \cdots, N_{SC}. \quad (7)$$

We also denote this contamination in an integrated format as

$$G^D_{L0} = W_{L0}\left(G^F_{L0}\right). \quad (8)$$

To determine an appropriate format of each $w_i(\cdot)$, we need to first understand the nature of the magnitude variations caused by environment changes. On each subcarrier frequency, the magnitude of CSI is a combination of signal strength from all multipaths. When the environment has been changed, two kinds of effects may apply on multipaths.

Effect 1: Some new multipaths appear, while some old multipaths disappear.

Effect 2: Some multipaths are enhanced or weaken.

An illustration of these two effects is presented in Figure 5. Two daily actions were taken to change environment, i.e., opening the windows and moving the couch. Path 1 is not affected by the changes, and thus stays the same. Path 2 disappears as the windows used to provide reflections are now opened. The power of path 3 is enhanced (or weaken) as it now propagates through a different part of the couch.

According to our experiments, Effect 1 brings a constant magnitude shift, while Effect 2 introduces a linear scaling on the original magnitude. Therefore, we model each contamination function as a combination of these two effects, and rewrite Eq. (7) as, for $i = 1, 2, \cdots, N_{SC}$,

$$|h^D_i| = \sum_j a_{i,j}|h^F_i| - \sum_k b^{down}_{i,k} + \sum_l b^{up}_{i,l}, \quad (9)$$

where $a_{i,j}$ is the magnitude scaling of an old multipath j, $b^{down}_{i,k}$ is the magnitude down-shift caused by the disappearance of an old multipath k, and $b^{up}_{i,l}$ is the magnitude up-shift brought by the appearance of a new multipath l. We can further summarize Eq. (9) as

$$|h^D_i| = a_i|h^F_i| + b_i. \quad (10)$$

Suppose we have the fingerprint reference profile G^F_{L0}, and

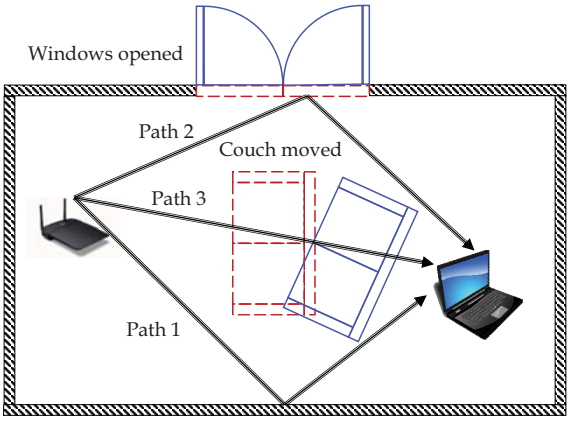


Fig. 5: The impact of environment changes on multipaths. Path 1 remains the same. Path 2 disappears due to the opened windows. Path 3 is enhanced (weaken) due to the moved couch.

N detected reference profiles $\{G^D_{L0}(p_1), \cdots, G^D_{L0}(p_N)\}$. We calculate the least squares estimates of a_i and b_i, for $i = 1, 2, \cdots, N_{SC}$, as

$$(a_i, b_i) = \arg\min_{(a_i,b_i)} \sum_{j=1}^{N} \left(|h^D_{i,j}| - a_i|h^F_i| + b_i\right)^2, \qquad (11)$$

where $|h^D_{i,j}|$ is the magnitude on the ith subcarrier of the jth detected reference profile, and $N > 2$. We adopt a Recursive Least Squares (RLS) method [27] to solve Eq. (11). We skip the details of this RLS method due to limited space. All the estimated contamination functions are saved in the set W_{L0}. Until now, we can estimate the contaminants of the reference location in an unsupervised way.

C. Contaminant Cleaning

We utilize the set of contamination functions W_{L0} to remove contaminants for both detected reference profiles and other online profiles. In fact, for each location Lk, there should be one W_{Lk} capturing the contaminants there. However, during the testing phase, there exist no unsupervised way to directly measure W_{Lk}. To tackle this, we adopt W_{L0} as the approximation for all W_{Lk}, i.e., for $\forall k, k \neq 0$,

$$W_{Lk} \cong W_{L0}. \qquad (12)$$

In this way, we can apply W_{L0} in cancelling the contaminants at other locations. The approximation error in Eq. (12) are to be addressed in Section VI.

Suppose the system online collects a new profile $\dot{G} = [|\dot{h}_1|, \cdots, |\dot{h}_{N_{SC}}|]$, which should be $G = [|h_1|, \cdots, |h_{N_{SC}}|]$ if the environment was not changed. If we can reproduce G, then the previously trained ML models can be used without retraining. To this end, we estimate G by cleaning the contaminants from $\dot{G}$ as, for $i = 1, 2, \cdots, N_{SC}$,

$$|\hat{h}_i| = w_i^{-1}(|\dot{h}_i|), \qquad (13)$$

where $|\hat{h}_i|$ is the purified magnitude, and $w_1^{-1}(\cdot)$ is the inverse function of $w_1(\cdot)$. And we have the purified profile as

$$\hat{G} = \left[|\hat{h}_1|, \cdots, |\hat{h}_{N_{SC}}|\right]. \qquad (14)$$

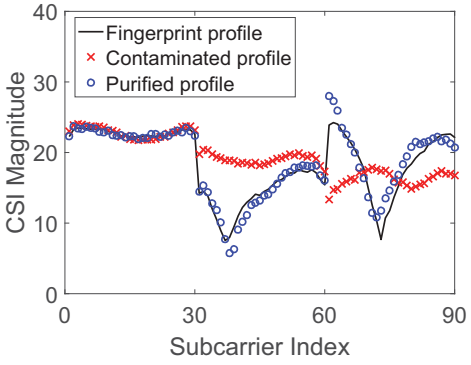


Fig. 6: AutoFi's contaminant removal result for a non-reference location $Lk, k \neq 0$.

An example of purified profiles is presented in Figure 6, together with the fingerprint profile and the contaminated profile. After cleaning up the contaminants, the purified profile is able to mostly reproduce the fingerprint profile. Yet, there still exist some residual errors. Next, we discuss how to further reduce these residual errors.

VI. The Feature-Preserving Autoencoder

To further reduce the residual errors of the contaminant removal module, in this section, we utilize an autoencoder to preserve and reproduce critical features of the fingerprint profiles.

A. Architecture of the Autoencoder

An autoencoder [5], [28] is basically an artificial Neural Network (NN) that learns an efficient coding for the data. Using this encoding capability, a feature-preserving autoencoder saves the critical features of the training data in its neurons. These neurons are later utilized to identify and emphasize the same features in the testing data. The architecture of our feature-preserving autoencoder is illustrated in Figure 7. This autoencoder contains an M_E-layer Deep Belief Network (DBN) for encoding, a feature vector of length N_f, and an M_D-layer DBN for decoding.

B. Training the Autoencoder

During the training phase, all fingerprint profiles are fed into this autoencoder all together without labels. The goal is to memorize current environment characteristics by preserving common features of all CSI profiles in the neurons.

A fingerprint CSI profile G is first input to the encoding DBN, which extracts N_f ($N_f \leq N_{SC}$) representative features from G. This feature extraction can be seen as an $N_{SC} - N_f$ encoding process, which is captured by the neurons of the encoding DBN. The extracted features are recorded as a set of code in the feature vector μ, reflecting the environment characteristics during the training phase.

The decoding DBN then reproduces $\hat{G}$ from the feature vector μ, by learning a decoding function to reverse the

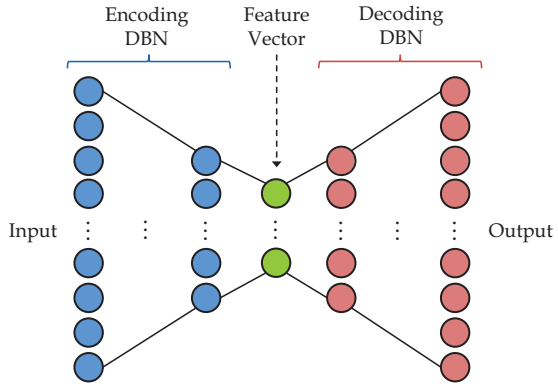


Fig. 7: The structure of the feature-preserving autoencoder in AutoFi.

encoding process. After decoding, the autoencoder outputs a reproduced profile $\tilde{G}$ of length N_{SC}. This reproduced $\tilde{G}$ emphasizes the important features stored in μ, while diluting the impacts of the others. This decoding process is represented by the neurons of the decoding DBN.

Backpropagation in conjunction with gradient descent is used to trained the above encoding and decoding DBNs.

C. Reproducing Features with the Autoencoder

During the testing phase, purified CSI profiles are fed into this autoencoder one by one. A purified CSI profile $\hat{G}$ goes through this feature-preserving autoencoder, and yields a reproduced profile $\tilde{G}$. For each $\tilde{G}$, its features relating to the environment changes are diluted or even removed, as these features are not encoded in the autoencoder's neurons. On the other hand, features relating to the previous environment are identified and enhanced through the encoding and decoding processes. In this way, the influence of environment changes is further reduced, and the localization accuracy is improved.

VII. Experiments

In this section, we conduct real-life experiments to illustrate the effectiveness of AutoFi. We first demonstrate the severe impact of environment changes on the localization accuracy. We then show that AutoFi is able to rescue the contaminated fingerprints and restore the accuracy to a satisfactory level.

A. Experiment Setup

1) The Environment Layout: We conduct the experiments in a residential apartment, of which the layout is presented in Figure 8. This apartment has a living room, a bedroom, a kitchen, a bathroom, and a corridor. We use totally ten locations in the apartment for evaluation. The distance between two locations varies from 0.8 meter (between $L1$ and $L2$) to 7 meters (between $L1$ and $L6$). We connect a laptop in the living room to a Wi-Fi AP in the bedroom, so as to create Wi-Fi traffic for localization.

2) Device Settings: The Wi-Fi AP is a Linksys WRT160N, operating on the 2.4 GHz band with a bandwidth of 20 MHz. The laptop is a Dell latitude e5530 with an Intel 5300 NIC and a CSI collection tool [29]. The whole localization system resides in this laptop. In order to create the localization traffic,

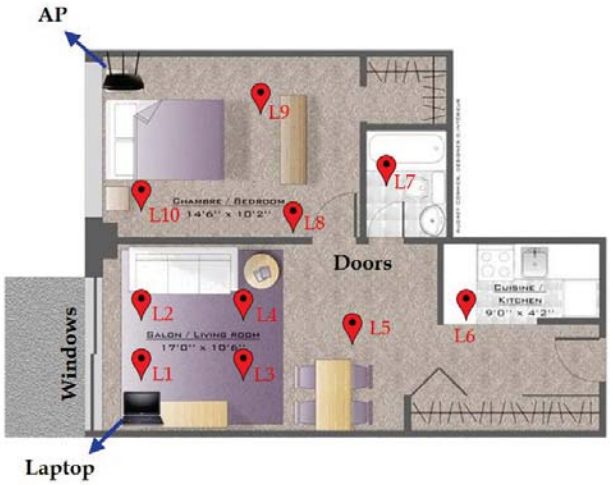


Fig. 8: Device layout and locations for experiments in a residential apartment.

the laptop constantly pings the AP with a frequency of 20 packets per second (pks). The packet size of the pinging is 64 bytes, which yields a 1.28 KB/s traffic. Compared with the bandwidth, this amount of traffic is almost nothing.

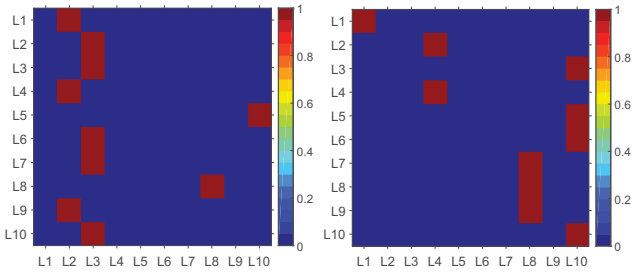
3) Data Recording: We record the CSI profiles while a person is standing at each location, and store these profiles with the location label. Each record lasts for 30 seconds. For each location, we repeat the recording four times while the person is facing four different directions (i.e., north, south, east and west). In addition, we record the CSI profiles when the apartment is empty as the reference profiles for contaminant removal. Three volunteers (two males and one female) are involved in the profile recording. Therefore, in one data set, we have 132 records (from ten physical locations and one reference location, four facing directions, and three people).

4) Environment Changes: In order to evaluate the environment changes, we record multiple data sets before and after different environment changes happen. We record the first data set as the training data set, when the windows and the doors are closed. We then record two other data sets as testing data sets, when the status of the windows and the doors are different. Table I describes these environment changes in data sets.

The training data set is used to 1) create the reference profiles in the contaminant removal module, 2) train the autoencoder, and 3) train the ML localization models. The testing data sets are then passed to the trained system to generate estimated location labels. These estimated labels are compared with the ground truth labels for evaluation.

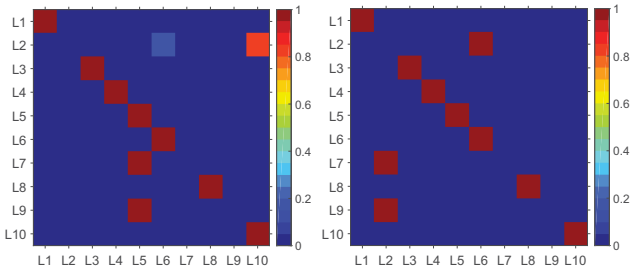
Data Set	Windows	Doors
Training Set	Closed	Closed
Testing Set 1	Open	Closed
Testing Set 2	Closed	Open

TABLE I: Environment changes between the training and testing data sets



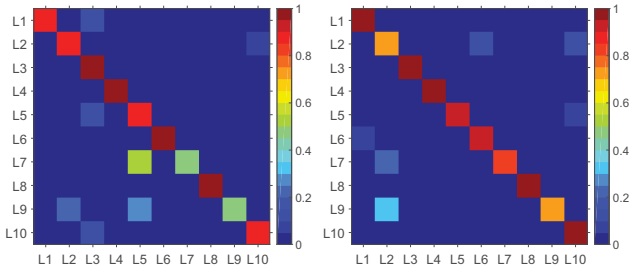
(a) Opening the windows. (b) Opening the doors.

Fig. 9: Confusion matrices of localization after the environment changes, with the Baseline approach.



(a) Opening the windows. (b) Opening the doors.

Fig. 10: Confusion matrices of localization after the environment changes, with the ConRmv approach.



(a) Opening the windows. (b) Opening the doors.

Fig. 11: Confusion matrices of localization after the environment changes, with the AutoFi approach.

B. Approaches Studied

We evaluate the following approaches in this paper.

- The **Baseline** approach reproduces common practice of DfP localization with no auto-calibration.
- The **ConRmv** approach adopts only the contaminant removal module as its auto-calibration technique.
- The **AutoFi** is the proposed approach with both the contaminant removal module and the autoencoder.

We compare the localization accuracies (recalls) of these approaches. A testing CSI profile is accurately localized, if its estimated label matches its ground truth label. We loop through all the testing profiles and collect the average accuracy for each location with all approaches. In this paper, we select the Random Forest (RF) model as the ML model to classify the locations by their CSI profiles. More advanced ML models can be applied. Yet, developing an advanced ML model is not the goal of this paper.

C. Contaminants from the Environment Changes

We first illustrate that a small environment change can indeed cause a large degeneration in the localization accuracy. Concretely, we conduct a ten-fold cross-validation on the training set, and collect the results as the localization accuracies before opening the windows. We then train the system with the training set, and test it with testing sets $1-2$. The testing results are considered as the localization accuracies after different environment changes. We use the baseline approach to show the system performance degradation.

The average accuracy over all locations is summarized in Table II. It is shown that, at first, the system can achieve a nearly perfect localization accuracy. However, once the testing CSI profiles are contaminated by the environment changes, the average localization accuracy degrades drastically. It is also shown that opening the windows introduces a larger degradation than opening the doors. This is because, before the windows are opened, there exist several reflection paths bouncing off the windows. These paths are missing after the windows are opened. Meanwhile, opening the wooden doors strengthens some signal paths and weakens some others. Yet, the number of multipaths is barely changed. Its impact on the CSI profiles is therefore much smaller than that of the missing paths.

Environment changes	Accuracy
None	99.7%
Opening windows	18.8%
Opening doors	41.7%

TABLE II: Accuracy degradation caused by environment changes, with the Baseline approach.

We present the confusion matrices of localization in Figure 9, so as to further investigate the nature of contaminants. It is demonstrated in Figure 9 that, after the environment is changed, the localization results tend to converge to a few locations. For opening the windows, the localization results mostly converge to $L2$ and $L3$. For opening the doors, most results converge to $L8$ and $L10$. Therefore, we have the following observation.

***Observation** 1:* The environment changes contaminate the CSI profiles by making them all look similar to only a few fingerprint profiles.

D. Rescuing the Contaminated Fingerprints

We next apply our auto-calibration approaches to address the contamination caused by environment changes. We first evaluate the ConRmv approach, which employs the contaminant removal module but not the autoencoder. The confusion matrices of the ConRmv approach are presented in Figure 10. We can clearly see from Figure 10 that the localization results are now placed along the diagonals of the matrices, instead of concentrating around a few off-diagonal elements. This suggests that the contaminant removal module is able to reshape the CSI profiles back to their fingerprinted forms to a certain extent. It thus partially addresses the issue summarized as Observation 1. As a result, the average localization

	Opening windows		Opening doors	
Accuracy	Mean	Min	Mean	Min
Baseline	18.8%	0.0%	41.7%	0.0%
ConRmv	69.0%	0.0%	70.0%	0.0%
AutoFi	84.9%	47.6%	90.2%	71.3%

TABLE III: Mean and minimum localization accuracies of different approaches.

accuracies increase to 69.0% (from 18.8%) and 70.0% (from 41.7%) for opening windows and opening doors, respectively.

However, the removal module may sometimes overly compensate the contaminants. For example, in the case of opening the windows, some profiles lean towards $L5$ and $L10$, which is not seen before contaminant removal.

To correct this residual error, the AutoFi approach further employs a feature-preserving autoencoder. The confusion matrices of the AutoFi approach are illustrated in Figure 11. It is shown that, with the help of the autoencoder, the localization results are more concentrated at the diagonals of the confusion matrices. The over-compensation of the contaminant removal is largely relieved. Consequently, the average localization accuracies further increase to 84.9% and 90.2% for opening windows and opening doors, respectively.

We summarize the improvement in localization accuracy in Table III. Compared with the other two approaches, the AutoFi approach not only improves the average accuracy, but also raises the worst-case accuracy from 0.0% up to 71.3%. While the other two approaches still have some dead spots, the AutoFi approach manages to cover all the locations.

VIII. Conclusion

Wi-Fi profiles have been widely adopted as fingerprints by device-free passive indoor localization. Yet, these Wi-Fi fingerprints are easily contaminated by small environment changes. The contaminants tend to make CSI profiles at different locations look similar to each other. Consequently, the fingerprints become inconsistent over time, which ruins the whole localization system as illustrated in this paper. To address this inconsistency issue, we propose an auto-calibration approach AutoFi to automatically reconstruct the fingerprints after environment changes. AutoFi incorporates a contaminant removal module and a feature-preserving autoencoder. Experiments in a residential apartment illustrate that AutoFi is able to rescue the localization system from environment changes. The localization accuracy is raised from 18.8% to 84.9% with a meter-level resolution.

References

[1] M. Kotaru, K. R. Joshi, D. Bharadia, and S. Katti, "Spotfi: Decimeter level localization using wifi," in *SIGCOMM*. ACM, 2015, pp. 269–282.

[2] F. Adib, Z. Kabelac, and D. Katabi, "Multi-person localization via RF body reflections," in *NSDI*. USENIX Association, 2015, pp. 279–292.

[3] S. Sen, B. Radunovic, R. R. Choudhury, and T. Minka, "You are facing the mona lisa: spot localization using PHY layer information," in *MobiSys*. ACM, 2012, pp. 183–196.

[4] J. Xiao, K. Wu, Y. Yi, L. Wang, and L. M. Ni, "Pilot: Passive device-free indoor localization using channel state information," in *ICDCS*. IEEE Computer Society, 2013, pp. 236–245.

[5] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[6] M. Sugano, T. Kawazoe, Y. Ohta, and M. Murata, "Indoor localization system using RSSI measurement of wireless sensor network based on zigbee standard," in *Wireless and Optical Communications*. IASTED/ACTA Press, 2006.

[7] S. Mazuelas, A. Bahillo, R. M. Lorenzo, P. Fernández, F. A. Lago, E. Garcia, J. Blas, and E. J. Abril, "Robust indoor positioning provided by real-time RSSI values in unmodified WLAN networks," *J. Sel. Topics Signal Processing*, vol. 3, no. 5, pp. 821–831, 2009.

[8] K. Liu, X. Liu, and X. Li, "Guoguo: enabling fine-grained indoor localization via smartphone," in *MobiSys*. ACM, 2013, pp. 235–248.

[9] D. Vasisht, S. Kumar, and D. Katabi, "Sub-nanosecond time of flight on commercial wi-fi cards," in *SIGCOMM*. ACM, 2015, pp. 121–122.

[10] J. Xiong and K. Jamieson, "Arraytrack: A fine-grained indoor location system," in *NSDI*. USENIX Association, 2013, pp. 71–84.

[11] J. Gjengset, J. Xiong, G. McPhillips, and K. Jamieson, "Phaser: enabling phased array signal processing on commodity wifi access points," in *MobiCom*. ACM, 2014, pp. 153–164.

[12] S. Kumar, E. Hamed, D. Katabi, and L. E. Li, "LTE radio analytics made easy and accessible," in *SIGCOMM*. ACM, 2014, pp. 211–222.

[13] S. Kumar, S. Gil, D. Katabi, and D. Rus, "Accurate indoor localization with zero start-up cost," in *MobiCom*. ACM, 2014, pp. 483–494.

[14] K. Wu, J. Xiao, Y. Yi, D. Chen, X. Luo, and L. M. Ni, "Csi-based indoor localization," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 7, pp. 1300–1309, 2013.

[15] R. Nandakumar, K. K. Chintalapudi, and V. N. Padmanabhan, "Centaur: locating devices in an office environment," in *MOBICOM*. ACM, 2012, pp. 281–292.

[16] Z. Yang, C. Wu, and Y. Liu, "Locating in fingerprint space: wireless indoor localization with little human intervention," in *MOBICOM*. ACM, 2012, pp. 269–280.

[17] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, "Zee: zero-effort crowdsourcing for indoor localization," in *MOBICOM*. ACM, 2012, pp. 293–304.

[18] J. Han, C. Qian, X. Wang, D. Ma, J. Zhao, P. Zhang, W. Xi, and Z. Jiang, "Twins: Device-free object tracking using passive tags," in *INFOCOM*. IEEE, 2014, pp. 469–476.

[19] L. Yang, Q. Lin, X. Li, T. Liu, and Y. Liu, "See through walls with COTS RFID system!" in *MobiCom*. ACM, 2015, pp. 487–499.

[20] M. Moussa and M. Youssef, "Smart devices for smart environments: Device-free passive detection in real environments," in *PerCom Workshops*. IEEE Computer Society, 2009, pp. 1–6.

[21] M. Seifeldin, A. Saeed, A. E. Kosba, A. El-Keyi, and M. Youssef, "Nuzzer: A large-scale device-free passive localization system for wireless environments," *IEEE Trans. Mob. Comput.*, vol. 12, no. 7, pp. 1321–1334, 2013.

[22] C. Xu, B. Firner, R. S. Moore, Y. Zhang, W. Trappe, R. E. Howard, F. Zhang, and N. An, "SCPL: indoor device-free multi-subject counting and localization using radio signal strength," in *IPSN*. ACM, 2013, pp. 79–90.

[23] Y. Chapre, A. Ignjatovic, A. Seneviratne, and S. Jha, "CSI-MIMO: an efficient wi-fi fingerprinting using channel state information with MIMO," *Pervasive and Mobile Computing*, vol. 23, pp. 89–103, 2015.

[24] X. Wang, L. Gao, S. Mao, and S. Pandey, "Csi-based fingerprinting for indoor localization: A deep learning approach," *IEEE Trans. Veh. Technol.*, vol. PP, no. 99, pp. 1–1, 2016.

[25] N. Ghourchian, M. Allegue, and D. Presup, "Real-time indoor localization in smart homes using semi-supervised learning," in *AAAI*, pp. 1–6.

[26] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury, "No need to war-drive: unsupervised indoor localization," in *MobiSys*. ACM, 2012, pp. 197–210.

[27] K. J. Astrom and B. Wittenmark, *Adaptive Control, 2013 edition*. Courier Dover Publications, 1994.

[28] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, 2010.

[29] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "802.11 with multiple antennas for dummies," *Comput. Commun. Rev.*, vol. 40, no. 1, pp. 19–25, 2010.