1
Big Data: A Revolution of the Location based Service and Wireless Sensing

1.1
Introduction

Sensing applications are becoming ubiquitous smart applications. Beyond smartphones, tablets and wearable devices, and sensing ability will eventually be embed and integrated in our living space. The recent surge in sensing applications is characterized by distributed collection of data by either self-selected or recruited participants for the purpose of sharing local conditions, increasing global awareness of issues of interest, computing community statistics, or mapping physical and social phenomena.

Most of these applications are location-based services (LBS), which is increasingly important to the user experience. In the big data era, the information provided by LBS enables meaningful and reasonable analysis or mining on the collected data. Some of the most common LBS applications include local news, navigation, points of interest, parking assistance, fleet management, emergency, asset tracking, location-sensitive building, and local advertisement.

To specify the mobile user's location, an intuitive solution depends on the mobile phone system. For example, the current cell phone ID can be used for identifying the base transceiver station (BTS) that the phone is communicating with. Once it is determined, the only thing left is to pinpoint the location of the BTS.

Other solutions use GPS technique. GPS is much more accurate than BTS based solution. Currently, it is very common that contains built-in GPS receivers. However, GPS cannot work well in indoor environments. An alternative common method is to use short-range positioning beacons. It is an ideal solution for existing utilizing WiFi or Bluetooth technologies and are ideal for indoor LBS applications.

Big data spurs the revolution of smart sensing and LBS, which in turn enriches big data in terms of data volume and application scale. Every digital process and social media exchange produces big data. Besides physical locations, velocity, direction, acceleration are all involved in the big data application. On the other hand, application for big data are becoming selective, because the explosive increase of sensed data will easily overwhelm the computing resource that people have. Therefore, sensing should proceed by extracting meaningful value from big data, requiring

optimal processing power, analysis capabilities and skills.

We are aiming at finding effective ways to estimate the location and sense the human behaviors purely by leveraging huge amount of wireless transmission metadata, including signal strength, transmission successful rate, and so on. The transmission metadata from all clients is collected from internet of things (IoT) devices and we use it to automatically establish the mapping between the spatial location spot and wireless signal strength distribution. When the sensing data obtained from each use reach to a certain amount, the features of locations could be extracted through crowdsourcing. All of these elements accord well with the needs of crowdsourcing and big data, where little can be assumed about users, and explicit input or other action from users is best avoided.

As a result, we achieve this goal without any explicit human participation. We expanded this methodology to estimate the number and behavior of people in indoor environments. There are two kinds of our sensing approaches based on big data: device-based sensing and device-free sensing. Device-based sensing approaches allow user to carry around a mobile phone, while device-free sensing approaches allow sensing the target without attaching any electronic tags in wireless environments.

We introduce the big data technology to excavate potential relation between wireless signal variation and the behavior of people, which efficiently reduces the training costs, extends the wireless sensing ability, and improves sensing accuracy.

1.2 Device-based Sensing using Big Data

1.2.1 Introduction

A critical challenge of LBS is to find the accurate location of mobile devices. In outdoor environment, GPS-based localization can provide sufficiently high accuracy for every GPS-enabled device [1]. However, in indoor environments, the existing solutions [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12] are neither pervasive nor interoperable.

Just like the "traffic delay" for each packet transferred in wired network which may act as an indicator the logic distance, each packet transferred in wireless network has a "received signal strength" [2] can also indicate the distance relationship between APs and client. Everyday TBs of data are silently transferred through the wireless infrastructures. Hence, a very large amount of transmission meta data are collected. *Could we discover the location information from these meta data?*

Turning this problem into a more practical way, we have the question: "*Can we automatically establish the mapping between geo-location and surrounding Wi-Fi signal fingerprint*?" In the following text, we introduce a crowdsensing based indoor localization scheme, SENIL. It could extract the fingerprint map from the mere knowledge of indoor floor plan and large amount of RSS samples passively collected from AP-end. In this way, any devices with wireless connection are continuously participating in the crowdsensing, while no exclusive APP is required to be installed.

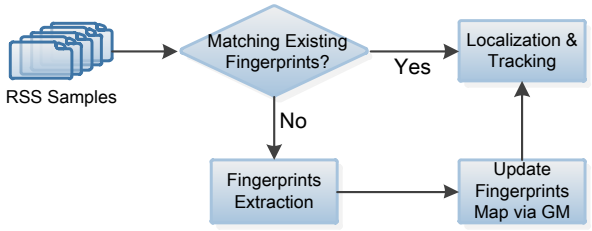


Figure 1.1 SENIL System architecture.

Our solution is based on a key observation that: although the location coordinates and wireless fingerprints are so disparate, they are all generated according to the same indoor manifold and therefore they share a highly similar *intrinsic structure*. If we recognize this shared structure embedded in these two disparate data sets, we will directly extract the fingerprints-to-coordinates mapping without indoor space measurement.

1.2.2 Approach Overview

Since RSS can be estimated by both sides, SENIL can be deployed in either AP-end or client-end. Whenever it is deployed at either end, the core of the system remains same. We should mention that the positions of Wi-Fi APs are not required to be known in either case. In the rest of the paper, we assume SENIL is deployed at AP-end, which mean APs will also work as sniffer to overhear the Wi-Fi traffic and conduct the necessary computation.

Fig. 1.1 demonstrates our system architecture. When a user enters the building, we assume that the device carried along the user will establish wireless connections. The Wi-Fi APs will record the RSS of the traffic sent from the client device and forward the data to a localization server. The RSS values will be used to build a matching with the fingerprint RSS values stored in the fingerprint database, which itself is populated using crowdsourcing techniques. The estimated moving distance, together with the estimated geodesic distance of different fingerprint locations in the map, will be used to further improve the quality of the matching, and thus, the accuracy of the localization.

1.2.3 Trajectories Matching

RSS fingerprint are the most representative RSS point extracted from many samples for a given position, and the error of the fingerprint will directly affect the mapping and localization accuracy. Besides the strong noise in RSS measurement, there are two main type of errors. The first is the well-known device-based measurement off-

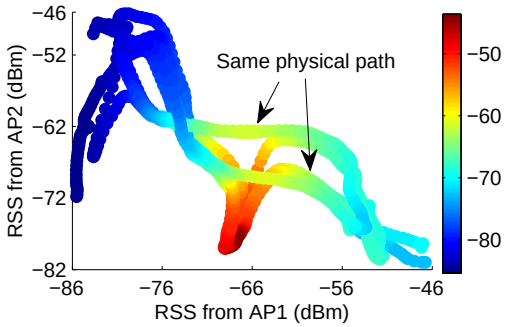


Figure 1.2 4 RSS trajectories measured from 3 APs [1]. The color denotes the value for 3rd AP: deeper color denotes higher RSS for the 3rd AP.

set, which is mainly caused by the variation of device antenna. We observe another measurement error, called *directional shadowing problem*, which has minor effect on traditional site-survey but has strong interference on crowdsensing based approach.

1.2.3.1 The Directional Shadowing Problem

Human body is a strong electromagnetic energy absorption object, which may cause directional shadowing effect to wireless signal. Therefore, when the same device is placed at different position near human body, *e.g.* shirt pocket or back pocket, the RSS measurements will be obviously deviated. This effect has been exploited in previous works [4, 13] to achieve Direction-of-Arrival (DoA) detection, however, it may cause severe error to unsupervised fingerprints-to-floor plan mapping, because there may be multiple parallel RSS trajectories corresponding to the same physical path. Fig. 1.2 presents an example of directional shadowing from real measurement data. In this situation, the mapping algorithm may wrongfully think that there are parallel paths between the start and end, and it is very likely to cause mapping failure.

1.2.3.2 Fingerprints Extraction

Previous approaches usually adopted cluster-based algorithm to extract fingerprints by merging the the nearby RSS samples within a certain threshold. This is a coarse-grained algorithm, and it cannot identify the device offset and directional shadowing. Our solution comes from a intuitive observation: although the RSS samples are highly deviated for the same physical position if under antenna variation and directional shadowing, the temporal RSS samples transition trends are very similar. If we could correctly identify the matching between these RSS trajectories, the fingerprints could be extracted without two errors mentioned above.

This intuitive idea could be transformed to a a n-partite graph matching problem [14, 15] if we see these trajectories as curve-shaped graphs. This can further be done by $n-1$ times iteratively graph matching between n-th trajectory and previous resulting RSS fingerprints graph. Relaxation-based approaches approximate the graph matching elegantly, and the only job is to build the affinity matrix $\mathcal{M}$.

Given two RSS trajectories, G^P and G^Q, each of them contains n^P and n^Q RSS

measurements respectively. A possible assignment a is defined as $a = (i, i')$, where $i \in \mathcal{V}^P$ and $i' \in \mathcal{V}^Q$. Given another possible assignment $b = (j, j')$, the compatibility $m_{a,b} \in \mathcal{M}$ is assigned as follows in current system:

$$\mathcal{M}_{(a,b)} = \begin{cases} e^{-||d_{ij} - d_{i'j'}||}, & \text{if } ||d_{ij} - d_{i'j'}|| > \epsilon \\ 0, & \text{otherwise} \end{cases} \tag{1.1}$$

where d_{ij} and $d_{i'j'}$ are the distance between i and j, and their assignment pairs i' and j', respectively. Since RSS attenuation along distance is non-linear, in our system the Minkowski distance [16] with value $p = 1.7$ is used to define the distance between RSS samples a and b by the following equation:

$$d_{ab} = \left(\sum_{i=1}^{n} |rss_{ai} - rss_{bi}|^p \right)^{\frac{1}{p}} \tag{1.2}$$

where n represents the numbers of all heard APs, and rss_{ai} for the i-th AP's RSS value of RSS sample a.

In each iteration the fingerprints extraction is a partial graph matching problem, and it is not tight integer constrained. We use spectral matching (SM) algorithm [14] to calculate the optimal column-wise vector $\mathcal{X}$ with length $n^P \times n^Q$. $\mathcal{X}$ is further reshaped to an *association matrix* $\boldsymbol{A}^{P \times Q}$, where each element $\boldsymbol{A}_{ij}$ denotes the *matching rate* between $i \in \mathcal{V}^P$ and $j \in \mathcal{V}^Q$.

When obtaining the association matrix $\boldsymbol{A}^{P \times Q}$, the fingerprints extraction is easy. Due to the representativeness of fingerprints, the task of extracting fingerprints can be easily done by identifying the sampling points which have no correspondence in historical data. Let $\boldsymbol{U}_A$ be the upper triangular matrix of $\boldsymbol{A}$ and $\boldsymbol{D}_A$ for its diagonal matrix, a RSS point i is considered to be a RSS fingerprint **iff**

$$\sum_i (\boldsymbol{U}_A - \boldsymbol{D}_A) = 0$$

After identifying the fingerprint points, we need to assign the RSS measurement values for each fingerprint. Due to the device variation, the RSS sample points for a single location may present in multiple clusters. We assign the RSS value for the specific fingerprint as the RSS samples which has the overall shortest distance to other RSS samples.

1.2.3.3 Fingerprints Transition Graph

The Fingerprints Transition Graph $\mathcal{G}^F = (\mathcal{V}^F, \mathcal{E}^F)$ records the spatial connectivity of all fingerprints. Since the vertice set $\mathcal{V}^F$ is the fingerprints set $\mathcal{C}^F$, we only need to determine the edge set $\mathcal{E}^F$ and the weight of edges $W(\mathcal{E}^F)$.

Basically, any two nodes $i, j \in \mathcal{V}^F$ have an edge $e_{ij} \in E^F$ if they satisfy the following two conditions:

1) i and j are subsequent RSS fingerprints within the same RSS fingerprints trajectory;

2) or when i and j belong to different trajectory, the distance d_{ij} is smaller than ϵ and at least one of both is the $start$ or end of a trajectory.

The weight of edges will be set to the absolute distance between fingerprints in traditional approaches [5, 6, 7], however, in SENIL, there is no absolute distance information due to the absence of physical space measurement. Fortunately, the walking duration becomes a distance indicator, and we can assign *virtual distance* to the weight of edges. The virtual distance is based on a reasonable assumption: although people may vary their walking speed according to mood or ongoing tasks, they usually maintain a constant speed during a single walking trip. Based on this assumption, we can infer the distance ratio between nearby fingerprints sequences along a trajectory, and eventually obtain a global distance matrix. Our virtual distance assignment algorithm is briefly described as blow.

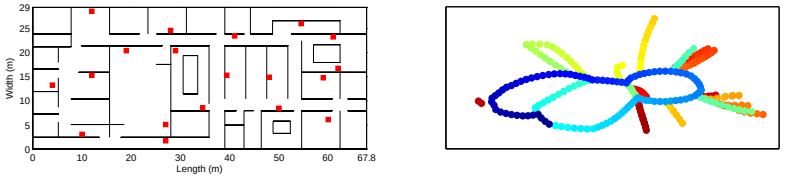
1) Find the longest common fingerprints sequence S_c among all RSS trajectories of all users.
2) Find the most stable user u_s who walks through sequence S_c with minimum time difference, and set the virtual distance $L_v(S_c)$ to the average walking time of sequence S_c by user u_s.
3) For every trajectory T_i containing sequence S_{lc}, calculate the virtual speed $V_v(T_i)$ according to the virtual distance $L_v(S_c)$, and assign virtual distance for other common fingerprints sequence within T_i.
4) Calculate the virtual speed for trajectories which pass through the assigned sequences, and calculate virtual distance for those un-assigned sequence.
5) Repeat Step 4 if there is un-assigned fingerprints sequence.
6) Assign virtual distance for every fingerprints within the same sequence using interpolation.

We design and develop a prototype system of SENIL. The prototype system is deployed in a large $2000m^2$ office environment with circular corridor network as shown in Fig. 1.3(a). We have collect more than 17000 RSS trajectories across more than 20 different mobile devices. Fig. 1.3(b) shows the extracted RSS fingerprint transition graph.

1.2.4 Establishing the Mapping between Floor Plan and RSS Readings

Ordinary floor plan is not friendly for crowdsensing based approaches, especially when client-side doesn't provide direction information. By studying the shortest walking distance (SWD) in indoor space, we realize that there is a highly curly 2D manifold embedded in a 2D floor plan polygon [7]. The unfolded version of this manifold will remove the direction information but preserving the SWD information, which is friendly for crowdsensing-based approaches.

The technical underpinning of automatic establishing the mapping is based on the topological similarity between the unfolded version of floor plan and the fingerprint transition graph, which is a typical graph matching problem. However, challenges



(a) Floor Plan (b) RSS Transition Graph

Figure 1.3 (a) The 2D floor plan of the deployed area. (b) The RSS transition graph extracted from very large amount of RSS readings.

exist in many aspects, *e.g.* unacceptable computation cost and low accuracy of large-scale graph matching [15, 14], scaling effect, and the unsupervised nature of auto mapping. Here we start from transforming the floor plan, then go through the auto mapping algorithm.

1.2.4.1 Floor Plan in Manifold's Eyes

n-dimensional manifold is a topological space that near each point resembles n-dimensional euclidean space, while globally not euclidean. The indoor floor plan shares the same property. Due to the obstruction of walls, the shortest walking distance (SWD) between two points i and j in the floor plan $\mathbf{P}$ equals to the euclidean distance $d_{eu}(i,j)$ **iff** the points i and j are within the same local isolation with direct Line-of-sight distance. When they are not in the same isolation, the SWD would be the geodesic distance $d_{geo}(i,j)$ which detour through various of obstructions. In this way, indoor space could be essentially viewed as a 2D-manifold $\mathfrak{S}$ embedded in a 2D-polygon $\mathbf{P}$.

In a manifold space, the euclidean distance is misleading. The geodesic distance actually reveals the true structure of the manifold. Therefore, we resample the 2D floor plan $\mathbf{P}$ using n points uniformly scatted. A n points graph G^M is then constructed. For every pair of nodes $i,j \in V^M$, there is an edge e_{ij} **iff** the correspondence points i^P, j^P in the floor plan $\mathbf{P}$ are in their mutual neighborhood with *direct line-of-sight* distance, and the weight for edge is the direct distance that $W(e_{ij}) = d_{eu}(i^P, j^P)$.

1.2.4.2 Unsupervised Mapping

Since the RSS samples are measured along users' walking trajectories, the RSS fingerprints transition graph G^F also share the same floor plan manifold structure. An intuitive idea of establishing the fingerprint map is to apply graph matching algorithm directly upon G^F and G^M. However, the accuracy and performance of large-scale graph matching ($>$50 points) is very poor for unsupervised situation. A lightweight relaxation to the problem is to apply graphing matching only on corridor points. Once the corridor points graph are mapped correctly, it is easy to match the rooms points. Unfortunately, the accuracy and performance of graph matching between such corridor points graphs is still not satisfied.

Fortunately, the highly sparse chain structure of corridor points give us a hint, and we devise a method called "Skeleton-Based Matching" to achieve unsupervised accurate mapping between G^F and G^M even for very complex indoor environment. The basic idea is that: due to the high sparsity and chained structure of corridor points graph, we can extract a coarse-grained skeleton graph from it while preserving the identical topological structure. Adopting GM algorithm on skeleton graphs will result high accuracy and performance. Once the main structures of G^F and G^M are matched, the remaining parts will be matched easily.

The algorithm is detailed in following 4 steps, including *skeleton graphs extraction*, *skeleton graphs normalization*, *skeleton graphs matching*, and *find-grained points matching*.

Skeleton Graph Extraction

Two sub-steps are required to extract the skeleton graphs. 1). Identifying the corridor points graph $G^{CF} \in G^F$ and $G^{CM} \in G^M$; 2). extracting skeleton graphs G^{SF} and G^{SM} based on corridor points graph $G^{CF} \in G^F$.

In the first sub-step, a customized centrality measure $C(V)$ is designed to identify the core corridor network. For a given point $v \in V$, its centrality $C(v)$ is measured as follows:

$$C(v) = \sum_{s \neq v \neq t \in V} \sigma_{st}(v) \tag{1.3}$$

where $\sigma_{st}(v)$ is the numbers of shortest path from s to t via v.

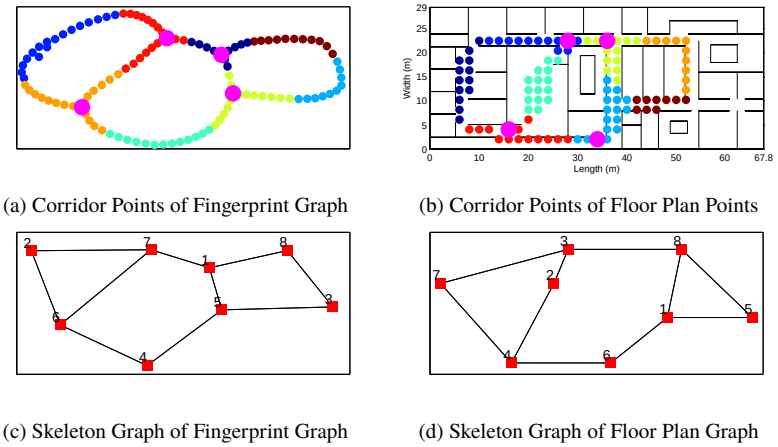
Based on this definition, we design a iterative algorithm to remove the non-central points effectively. In each round of iteration, the centrality $C(v)$ is measured for every points. If $C(v)$ is smaller than a low-bound τ, then remove the points from the graph. This procedure repeats until no points is removed.

In the second sub-step, the skeleton graphs V^S is generated by clustering the corridor points graph G^{CF} and G^{CM}. We use spectral clustering (SC) [17] as the clustering algorithm. SC is computationally faster than K-means and it only requires the adjacency matrix which is exactly suitable in our case that both G^{CF} and G^{CM} are represented only in adjacency matrix. Fig. 1.4(a) and (b) show the corridor points of fingerprint graph and corridor points.

By clustering on G^{CF} and G^{CM}, we obtain the vertices set of skeleton graph G^{SF} and G^{SM}. The edge set E^{SF} and E^{SM} follow the underlying points, that if two points i, j, belonging to different clustersc_a and c_b respectively, have an edge, then there is an edge between c_a and c_b. The weight of edge e_{ab} is defined as the shortest distance between the *central points* of cluster a and b, and the *central point* of a cluster is the point i which has the shortest distances to other points within the cluster. Fig. 1.4(c) and (d) show the extracted skeleton graph of fingerprint graph and corridor points respectively.

Graphs Normalization

Since the graph matching algorithm assigns the matching score according to pairwise distance, G and G^M must be normalized to same scale so as to guarantee an accurate



(a) Corridor Points of Fingerprint Graph

(b) Corridor Points of Floor Plan Points

(c) Skeleton Graph of Fingerprint Graph

(d) Skeleton Graph of Floor Plan Graph

Figure 1.4 Corridor Points and Skeleton Points extraction from fingerprint graph and floorplan graph, respectively.

matching result.

The scaling effect can be undone by some global normalization of the graph shape. Possible ways includes normalization of bounding box, or normalized Laplace-Beltrami eigenvalues (LBE). The bounding box approaches works only for rigid graph, while LBE is sensitive to deformations. We use the commonly accepted *longest geodesic distance* L_{lg} as scale indicator.

Skeletons Matching

After the extraction and normalization of skeleton graphs, we now find the best correspondence between G^{SF} and G^{SM}. Let positive symmetrical square matrix M^{SR} and M^{SM} represent their adjacency matrices. We build the affinity matrix $M^{SR\times SM}$ for graph matching as follows:

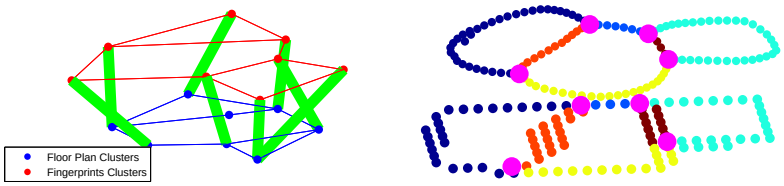
$$M^{SR\times SM} = e^{(\mathbf{1}^{SM} \otimes M^{SR} - \mathbf{1}^{SR} \otimes M^{SM})^2} \tag{1.4}$$

where $\otimes$ denotes the Kronecker product [18] and $\mathbf{1}^{SR}$ denotes the full-$\mathbf{1}$ matrix with the same size of G^{SR}. The idea behind Eq.(1.4) is to enumerate all possible matching candidates and store them in a large adjacency matrix $M^{SR\times SM}$. We use RRWM algorithm to perform the graph matching, and Hungarian algorithm is further applied to discretize the $\mathcal{X}$ in order to meet the final integer constraints $\mathcal{X} \in \{0,1\}^n$. Fig. 1.5(a) shows the skeleton graphs matching.

Corridor Points Matching

Although G^{SF} and G^{SM} are matching in previous step, however, the corresponding clustering groups in G^{CF} and G^{CM} are not necessarily matched due to the inconsistency of clustering operation.

In the corridor points graphs, we notice that only a few points connect multiple



(a) Skeleton Graph Matching (b) Corridor Points Matching

Figure 1.5 The Skeleton and corridor points matching

chain structures, and they may serve as the *bridge points*. Since the topological structures of G^{CF} and G^{CM} are identical, if we could identify the correct correspondences of these *bridge points*, the points within the chain structure will be matched easily. In order to identify the bridge node, we introduce a new metric called "bridge centrality", which is equal to the number of shortest paths from all vertices to all others within nearby clusters that pass through that node, defined as follows:

$$C_{bg}(v) = \sum_{\{(s,t)|v\in c^i, s,t\in NN(c^i), s,t\notin c^i\}} \sigma_{st}(v) \tag{1.5}$$

where $NN(c^i)$ denotes the nearby clusters around c^i. The bridge point will be the point with the highest bridge centrality.

After identifying the of correspondence of bridge points in G^{CF} and G^{CM}, the chain structures are easily matched according to the start and end bridge points. Fig. 1.5(a) shows the corridor points matching.

Rooms Points Matching

If we remove the corridor points graph G^{CM} from G^M, the rooms points will naturally forms several clusters. Each room points cluster C_R connects to the corridor by a door point, and the rooms matching is also easy by matching the door points in both G^F and G^M.

However, if there is more than one rooms connect to a single door point in the corridor, *e.g.* two rooms in opposite side along the corridor, there may be mis-matching. This kind of mismatching can be canceled using coarse-grained propagation model. Along with the corridor points matching, the coarse-grained position estimation for APs can be done. With the rough location of APs, the rooms mis-matching can be easily eliminated by checking the RSS values. The smaller RSS difference means higher probability of being in the candidate rooms. Figure 1.6 shows the accuracy of corridor points matching and room points matching.

1.2.5 User Localization

SENIL provides a unified localization and tracking service by treating the direct localization request as a tracking request without historical data. Here we mainly focus

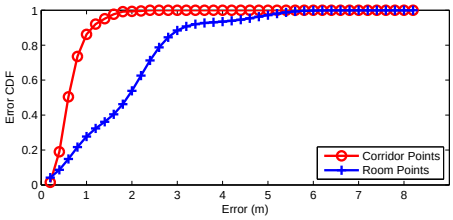


Figure 1.6 The matching error of corridor points and room points.

on tracking technique in SENIL. In our solution, the users' trajectories are globally determined from the very beginning by transforming the tracking problem to a graph matching problem between the measured RSS samples transition graph G^S and the fingerprint transition graph G^F.

1.2.6
Graph Matching Based Tracking

Graph matching based tracking is to find the best correspondence between the sequence of RSS samples of tracking request and extracted fingerprints. This is exactly the same matching process undertaken in Section 1.2.3, except for the differences that the tracking is to find the matched points, while in Section 1.2.3 focus on collecting RSS fingerprints.

Let $\mathcal{X}^{n^S \times n^F}$ represent the association matrix obtained through spectral matching [14] where n^S and n^F are the numbers of RSS samples and candidate fingerprints set respectively. Due to the error in RSS measurement and fingerprints map construction, a single RSS samples p_i may correspond to multiple fingerprints in $\mathcal{X}$,*e.g.*, a RSS sample may correspond to two fingerprints, one is in corridor, and another is in a room. Fortunately the temporal correlation can help eliminate those false correspondence by checking the spatial continuity between current and subsequent candidate fingerprints. After eliminating the false correspondence, the globally estimated coordinates sequence will be given by $T_{G^S} = f_{G^T}(G^S)$, where $f : fingerprint \rightarrow G^M$ represents the mapping from fingerprints to the floor plan manifold $G^M y$, and T_{G^S} are the resulting coordinates sequence in G^M.

1.2.7
Evaluation

We site-survey the testing environment with approximately 3×3m grid and implement RADAR system as comparison to SENIL. 5 students walk along a pre-defined testing trace for 5 rounds. Fig. 1.7 shows the error CDF of RADAR and SENIL with different length of historical data. Based on the commonly accepted 3×3m grid site-survey, RADAR has less than 3.5m error in 80% situations, while SENIL is only 1m error larger than RADAR. When in tracking scenarios, every 3m of increased track-

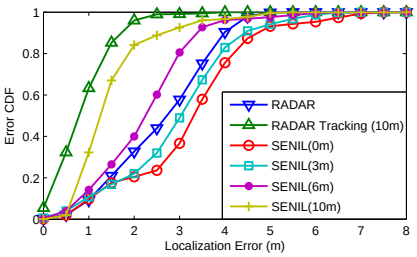


Figure 1.7 The localization & tracking error CDF with different amount of RSS fingerprints data. The RADAR scheme is as comparison.

ing range brings approximately 0.7m tracking accuracy improvement. When tracking range is 10m, SENIL tracking accuracy is very close to RADAR-based tracking. In practical tracking scenarios, 10m of tracking range is a very low requirement.

1.3 Device-free Sensing using Big Data

1.3.1 Customer Behavior Identification

With the explosive development of e-business, making a buck becomes more challenging for Brick-and-Mortar (B&M) retailers. The availableness of online payment, the pervasiveness of smartphones, and the conveniences of few-day delivery make the future of physical retailing ever more tenuous. According to a report issued last year, Amazon has had tremendous success by using collected online data to discover what products his users are likely to buy. Moreover, as the study noted, Amazon reported that 30 percent of sales were due to its recommendation engine [19].

Physical stores and retailers seem destined to be left out in the cold. Fortunately, with the recent advanced technologies, retailers have great chances to thrive in the future. In fact, it is possible that traditional retailers could have a better understanding of their customers than ecommerce firms do. Through a combination of Big Data analytics and leading-edge technologies, brick and mortar retailers are promising to stay relevant. Retailers can also benefit from Big Data to unlock the mysteries of customer behavior and understand why, what, how people buy. Accumulating the data of customer behaviors over a period could help retailers to determine which Ads should be introduced, where and how items should be displayed, etc.

As an intuitive solution, retail stores can use video cameras to distinguish whether a customer is male or female, know what he/she buy by identifying the customer's gestures, and analyze why he/she did or did not buy a certain product by interpreting the customer's facial expressions. This solution is attractive. However, privacy concerns raise a barrier for its implementation. To achieve above goals without compromising the user's privacy and disturbing the customer purchasing experience, we

(a) Discovering popular items

(b) Revealing explicit correlations

(c) Disclosing implicit correlations

Figure 1.8 Three objectives of CBID.

propose a RFID-based solution (called CBID) to identify on-site customer behaviors and further deduce products correlations, which can provide rich information for B&M stores.

CBID is implemented by commodity RFID readers and off-the-shelf tags. Supposing each products in the store is attached by a passive tag, CBID system can identify customer behaviors by collecting and analyzing wireless signal changes of RFID reader and tag communication. Specifically, CBID has three main objectives, namely discovering popular items, revealing explicit correlations, and disclosing implicit correlations.

1.3.1.1 Doppler-based Popular Item Discovery

Popular items are the ones with great interests of customers. Customer behavior analysis of sales history can only consider the purchase actions to indicate interests of items. The unique advantage of CBID is that it can identify other actions such as picking up an item, as shown in Figure 1.8(a). Intuitively, the more times an item is picked, the more attention the item gains. B&M stores can then adjust the display accordingly. In the CBID system, we count the detected pick-up actions of an item as a metric to reflect the amount of customer attention gained by an item, namely *popularity*.

Popular item discovery in CBID can be achieved by detecting and counting the pick-up events by customers. In this part, we propose an accurate and robust discovery algorithm of popular items discovery by detecting variations of tag's Doppler frequency shift. Doppler reflects the relative velocity between a reader and tag. When a customer picks up a tagged item, the speed of item movement, $i.e.$, the velocity of the attached tag perpendicular to the reader antenna, can result in frequency shifts.

Adaptive Doppler Peaks Detection

After accurate estimation of the Doppler frequency shift, to detect a pick-up action, our system should find the peaks of Doppler frequency values quickly and precisely. Many existing algorithms have been proposed for frequency change detection or peak detection. However, they require to record measurement data and then apply analysis tools. In real deployment of the CBID system, keeping the values of Doppler frequency would be a overloaded task considering a huge amount of data will be generated by monitoring the movements of thousands of tags in very short

time intervals.

Therefore, we design an accurate magnitude threshold-based detection which drops the data not related to the Doppler change as much as possible. A challenge is that such a deterministic threshold does not exist because of two limitations in real deployment. 1) Tag diversity induces diverse noise levels across different tags, and 2) item locations diversity also raises different Doppler patterns. For the above reasons, we adopt an adaptive CUSUM algorithm [20] to track possible changes in Doppler values.

Let $d(t)$ denote the Doppler frequency value at time t and its probability density is denoted by p_θ, where θ is the distribution parameter. Assume that a change happens at time T, the probability density of Doppler frequency changes from p_x to p_y. Then we can use a logarithmic ratio to compare these two distributions:

$$l_y = \log \frac{p_y(d(t))}{p_x(d(t))} \tag{1.6}$$

For the parameter x, it is easy to be obtained from the historical change record. But the parameter y, which is also called post-change distribution parameter, is unknown. It also varies during the detection process, leading more difficulty to predict.

To solve this problem, we define the change time T as:

$$T = \inf \left\{ t | \max \sum_{i=k}^{t} l_y(i) \geq h \right\}, 1 \leq k \leq t, \tag{1.7}$$

where h is a predefined value [21]. Given that the post-change distribution is unknown, the time T cannot be directly known from Equation 1.7. CBID then recursively applies the CUSUM method proposed in [20] to determine the range of parameter y. Based on the range of y, we can detect changes of the Doppler frequency more accurately.

We verify the feasibility of above solution via extensive experiments. We conduct experiments in three different scenarios. Here we only show a small part of results about change point detection in Figure 1.9. Each of the subfigures 1.9(a)-(d) corresponds to a pick-up experiment. In such an experiment, a volunteer picks up a book from a shelve and puts it back. We record the signal phase and time stamp of every tag packet during this period. The blue curve represents the estimates of Doppler frequency. By examining these estimates using the adaptive CUSUM algorithm, CBID detects a sudden change marked by a red circle in each of the subfigures. Experimental results of four actions indicate that the algorithm works well in different scenarios. As a result, the CUSUM algorithm can successfully skip perturbations in signal caused by environmental noise and discover the real change point in Doppler frequency values.

1.3.1.2 Location-based Explicit Correlation Discovery

We say items have *explicit correlations* if they are rival or complementary. For example, a customer may compare a bottle of CocaCola and a bottle of Pepsi. As another

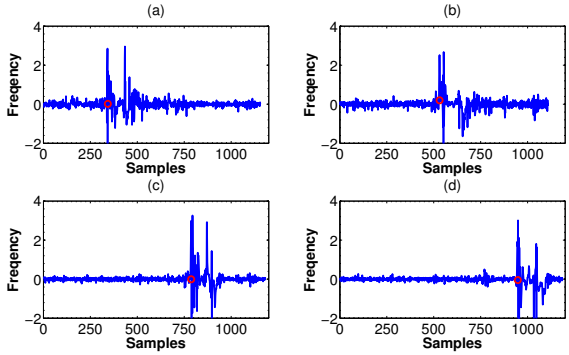


Figure 1.9 Change point detection by adaptive CUSUM: The subfigures show the Doppler frequency changes of four continuous pick-up actions. The blue curve shows the estimated Doppler values and the red circles are the change points detected by adaptive CUSUM.

example, a customer may pick up a cap and a jacket to decide if they match each other. Sales history analysis may reveal some of complementary items but can hardly discover rival items. The pick-up behaviors of explicit-correlated items are spatially and temporally close, $e.g.$, a customer may pick up these items simultaneously (Figure 1.8(b)). The CBID system applies novel passive tag localization methods to reveal such spatial and temporal correlations of moving items.

The main challenge is to connect items that are picked by a same person. However collected data are unable to report which items belong to whom. For the example of the scenario shown in Figure 1.8(b), two customers stand in front of a goods shelf. When three events of item pick-up are detected, our system should further figure out that two pick-up actions belongs to a same customer. It is generally the truth that when two items are holding by a same person, their positions should be close. Inspired by this, we form the problem of explicit correlation discovery as locating tagged items and deciding which tags are geographically close.

To locate close tags, an intuitive approach is to use the RSS of backscattered RF signal as the position fingerprint. Nevertheless, this method has some well-known drawbacks. First, it utilizes the Friis Equation that clarifies an RSS is inversely proportional to the biquadrate of the distance. The equation can only become valid with a strong assumption: the communication parties should be in a unobstructed free-space. In indoor environments, the multipath effect will yield severe influence to RSS measurement. Second, it is a huge workload to build an RSS map in advance. In addition, recent studies further show that such an RSS value strongly depends on the multipath profile [22]. Multipath profiling for localization, though being accurate, requires modifications on the reader hardware, $e.g.$, adopting the technique of synthetic aperture radar. These modifications result in non-trivial customization overhead.

Thus, we propose to allow the reader to intentionally move its antenna and collect

multiple RSS samples from a same tag at different antenna locations. CBID then uses multiple RSS from a same tag as a new metric for location. This method does not need any modification on existing RFID hardware and protocols.

Antenna Movement Model

Different from traditional RSS models where one tag location is assumed to be corresponded to a single RSS value, the proposed solution in CBID moves the reader antenna and collects multiple RSS samples for a tag. We first develop a simple antenna movement model to characterize the direction-distance relationship at different antenna locations. Based on that, CBID can retrieve the RSS information of tags and determine whether they are close or not. In this new model, there are two requirements.

- In order to improve the accuracy, it is better to avoid changing the dominance of direct path when antenna moves. So, the antenna should not move out of a range. We define an upper bound of this range as $\frac{Y}{5}$, where Y is the vertical distance between the antenna and tag (as shown in Figure 1.10).
- Inspired by Jakes Model [23], the signal within half wavelength are indistinguishable. Therefore the distance of antenna movement (r) has a lower bound $\frac{\lambda}{2}$.

In summary, we have $\frac{Y}{5} \leq r \leq \frac{\lambda}{2}$. To meet these conditions, we prefer to simply move an antenna back and forth in our implementation, termed as shuttle antenna. Figure 1.10 illustrates an example of the antenna movement model, where O is the origin location of antenna, T is a static tag at position (x, Y) with distance d from O, and the direction from O to T is in the angle α. A is the shuttle antenna. The position vector of tag T is :

$$\overrightarrow{OT} = (d \cdot \cos\alpha, d \cdot \sin\alpha) = (x, Y) \tag{1.8}$$

Suppose that A moves along the vertical axis within a maximum range R. The displacement of A periodically changes with t within the range $(-R, R)$. We take a sine function to simulate this displacement for simplicity. Therefore, the position vector of A can be represented as:

$$\overrightarrow{OA} = (0, R \cdot \sin t) \tag{1.9}$$

The distance vector from the antenna to tag is $\overrightarrow{AT}$:

$$\begin{aligned} \overrightarrow{AT} = \overrightarrow{OT} - \overrightarrow{OA} &= (x, Y) - (0, R \cdot \sin t) \\ &= (x, Y - R \cdot \sin t) \end{aligned} \tag{1.10}$$

Integration of Multi-RSS

The Received Signal Strength (RSS) is a measurement on the power of a received radio signal. In RFID systems, RSS is one of the reader outputs, reflecting the power of received backscattered signal P_{Rx}. RSS can be measured in the unit of dbm and

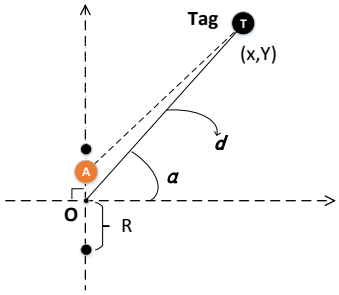


Figure 1.10 Illustration of Antenna Movement Model

calculated as: $RSS = 10\lg(\frac{P_{Rx}}{1mW})$. According to the Friis Function [24], we can model RSS as:

$$RSS = 10\log\left(\frac{P_{Tx}}{1mW}T_b G_r^2 G_t^2\left(\frac{\lambda}{4\pi d}\right)^4\right) \tag{1.11}$$

In this part, we propose to use a new metric, called Integration of Multi-RSS (IMR), as the fingerprint of a tag's location. By moving the reader antenna, CBID obtains a set of RSS values for a fixed tag ($i.e.$, a fixed location) when the antenna moves. The RSS value varies because $\overline{AT}$ changes. Let $M = \frac{P_{Tx}}{1mW} \cdot {G_r}^2 {G_t}^2 T_b (\frac{\lambda}{4\pi})^4$. According to Equation 1.11, at time t, the RSS received from the tag can be represented as:

$$\begin{aligned} RSS(t) &= 10\lg\left(M \cdot \frac{1}{|AT|^4}\right) \\ &= 10\lg\left(M \cdot \left(\frac{1}{\sqrt{x^2 + (Y - R\sin t)}}\right)^4\right) \\ &= 10\lg\left(M \cdot \left(\frac{1}{x^2 + (Y - R\sin t)}\right)^2\right) \end{aligned} \tag{1.12}$$

Let P_{Tx} be transmit power of reader, which is set to $32dbm$. G_r is $8dbi$ for our reader antenna Laird A9028R30NF. G_t is $2dbi$, which is the typical gain for real dipole-like tag antenna. T_b is $\frac{1}{3}$ [24] here for backscatter transmission loss. Y can be measured through real deployment and in our experiment, we put the shuttle antenna $1m$ away from the tags. Then, the only unknown parameters in Equation 1.12 are the tag's horizontal axis value x and the time t. The IMR metric is used to reflect t and x, and thereby fingerprint the tag's location.

The CBID reader collects the RSS values of a tag in a complete antenna movement

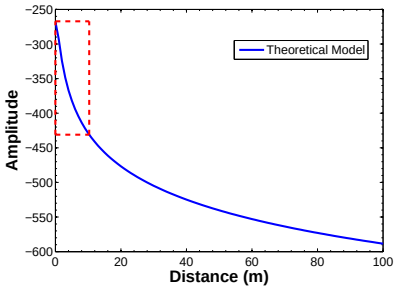


Figure 1.11 The IMR values of different tag positions from our theoretical model, given the same vertical distance. The red dotted box marked the practical reader range in $0 \sim 10$ meters.

cycle. The multiple RSS values together provide the IMR fingerprint of the tag:

$$\begin{aligned}
&\int_0^{2\pi} RSS(t)dt \\
&\quad = \int_0^{2\pi} \left(10\lg\left(M \cdot \left(\frac{1}{x^2 + (Y - R\sin t)}\right)^2\right)\right) dt \\
&\quad = \int_0^{2\pi} 10\lg M dt - \int_0^{2\pi} (20\lg(x^2 + (Y - R\sin t)))dt \\
&\quad = 10\int_0^{2\pi} \lg M dt - 20\int_0^{2\pi} \lg(x^2 + (Y - R\sin t))dt
\end{aligned} \tag{1.13}$$

Replacing $\sin t$ with θ in Equation 1.13, we have

$$\begin{aligned}
&\int RSS(\theta)d\theta \\
&\quad = 10\int_0^{2\pi} \lg M dt - 20\int_0^{2\pi} \frac{\lg(x^2 + (Y - R\sin t)) \cdot \cos t dt}{\cos t} \\
&\quad = 10\int_{-1}^{1} \lg M d\theta - (20\int_0^{1} \frac{\lg(x^2 + (Y - R\theta))d\theta}{\sqrt{1-\theta^2}} \\
&\quad + 20\int_{-1}^{1} \frac{\lg(x^2 + (Y - R\theta))d\theta}{\sqrt{1-\theta^2}} \\
&\quad + 20\int_{-1}^{0} \frac{\lg(x^2 + (Y - R\theta))d\theta}{\sqrt{1-\theta^2}}
\end{aligned} \tag{1.14}$$

Given different x values, we can achieve the relationship between the tag locations and the proposed fingerprint IMR, which is demonstrated in Figure 1.11. We can see that when a tag's vertical distance Y to the antenna is constant, its IMR value is inversely proportional to the tag's horizontal distance x.

The IMR metric is better than those used by traditional RSS methods for tag localization with the following reasons. Let S be the signal vector received from a

given tag. S can be considered as the sum of a multi-path signal *MPS* and a noise signal N. Thus, we have the expression $S = MPS + N$. Normally, we can assume that the noise signal follows a typical gaussian distribution with a mathematical expectation of 0. Hence the integral value of N should tend to 0. On the other hand, the multi-path effect may cause RSS changes. In some positions and angles the RSS may be strengthened, while in other cases, the RSS may be weakened. In our implementation, we adopt the shuttle antenna pattern and collect a set of RSS values for each tag's position. Multiple sampling of a tag can help the system to neutralize the multi-path effect. Therefore, IMR-based localization has a better performance than single-RSS based approaches.

1.3.1.3 Movement Pattern Based Implicit Correlation Discovery

We say items have *implicit correlations* if they share similar pick-up, moving, or purchase correlations. The most famous example is the tale of beer and diapers, which says that beer and diaper sales are explicitly correlated because young fathers who are sent to buy diapers would also buy beer for themselves. CBID uses tag trajectory monitoring and clustering to disclose these correlations by detecting and analyzing movement patterns of tags, as shown in Figure 1.8(c).

Based on Section 1.3.1.1, the CBID system knows which items are picked up by customers. Furthermore, we expect to find out the reason behind. For example, a customer who has put onions and tomatoes into the shopping cart may also buy burgers. This correlation can be expressed as $Pre\{Onion, Tomato\} \Rightarrow Pn\{Burger\}$. $Pre\{\}$ represents that a customer picked the item previously, while $Pn\{\}$ represents that the customer is likely to pick the item in future. Previously picked items may be in hand or in a shopping cart, within a short range in either way. Identifying $Pre\{\}$ and $Pn\{\}$ items helps the system to connect items with implicit correlations.

Problem Formulation

When a customer chooses $Pn\{\}$ items and leaves the shopping area together with his/her $Pre\{\}$ items, the reader can easily detect the movement of these items. Hence if only one customer stands in the area, the correlated items are relatively easy to detect. However, real situations are much more complicated. Considering two or more customers $(C_1, C_2, ..., C_n)$ who are shopping in front of a shelf and they all have their wished items. Some of them may have picked some items and leaved this area almost at the same time. We denote all tags belonging to these customers are *near-context* tags for this area. The near-context tags are actually identified in one area of interests and their time stamps of residence differ slightly within certain time tolerance. In our experiments, we set the time tolerance as 0.5 second.

Then the problem is formulated as, giving all the data of near-context tags, for each $Pn_i\{\}$ item how to discover the previous items set $Pre_i\{\}$ which belongs to the same customer C_i.

One intuitive solution to this problem is to record each tag's Doppler frequency value and monitor its movement trajectory. For the items in a same shopping cart, they will move together and share a similar movement pattern. The traces of all near-context tags can be clustered to different customers. However, such a solution

encounters two challenges in a real RFID system.

- Irregular sampling period: In an UHF RFID system, identifying tags will experience a collision-arbitration process, by which different tags are identified in different time slots. This means the sampling time for different tags will not be the same. In addition, several other factors, such as signal attenuation due to tag movement, will also lead different sampling times to different tags. This problem makes the sample density varies over time for each tag and then ruins the accuracy of clustering.
- Unknown number of customers: As the number of customers is unknown for the system, exhaustive search for the best clustering result is an NP hard problem. Therefore, an effective clustering algorithm with rapid convergence is required.

Segment-based Interpolation

To overcome the first challenge, we implement a segment-based interpolation approach which makes all raw data have the same number of samples of different tags. We set the time length of one segment as $T_{seg} = 1s$ and find the minimal time stamp (T_{min}) and maximal time stamp (T_{max}) of all data reported from near-context tags.

Then the whole time period is partitioned into N segments, where

$$N = \lceil (T_{max} - T_{min})/T_{seg} \rceil$$

In each segment, CBID applies the Linear Interpolation method to fit the Doppler curves. The advantage of this method is that we can reduce the interpolated error. In addition, considering each segment of data is indeed one observation, this method will increase the amount of data used for the clustering algorithm presented in the next subsection.

Iterative Clustering Algorithm with Cosine Similarity

In this subsection, we present an iterative clustering approach to organize tags in different groups, which utilizes cosine similarity measurements.

To clearly explain the goal of this algorithm, we use one group of experiment data to serve as an example. Figure 1.12 shows tags' Doppler frequency data processed by the segment-based interpolation. In this set of experiments, two volunteers emulate a shopping procedure and each of them has a shopping cart. Their trajectories are random. Volunteer A has only one item which is associated with Tag1. The other three items (with Tag2, Tag3, and Tag4, respectively) are all in volunteer B's shopping cart. In order to extract implicit correlations of the items, we have to cluster the four tags.

As described in Section 1.3.1.3, we divide the Doppler data into N segments. Each segment of data is considered as a vector. We implement the cosine calculation to measure the similarity between two vector observations within the same time period. Supposing there are M near-context tags. The similarity metric $S^i_{j,k}$ for two

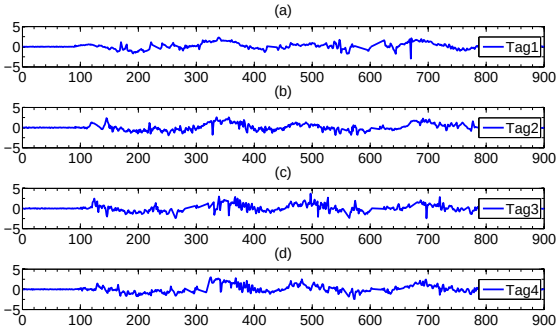


Figure 1.12 Doppler data of implicit-correlation experiment: Subfigure (a)-(d) demonstrate 4 tags' doppler data after segment-based interpolation. Tag1 belongs to volunteer A and the other 3 belong to volunteer B.

segments F_j^i, F_k^i can be expressed as :

$$S_{j,k}^i = \frac{F_j^i \cdot F_k^i}{\left\|F_j^i\right\| \left\|F_k^i\right\|}, 1 \leq i \leq N, 1 \leq j,k \leq M \tag{1.15}$$

where i denotes the ith segment of Doppler data and j,k are tags' identifiers. The similarity ranges from -1 to 1, where -1 means completely different and 1 means exactly the same.

For each time period i, $\binom{M}{2}$ times of comparisons should be conducted. Considering all the time periods, the size of the result matrix is $\binom{M}{2} \times N$:

$$\begin{bmatrix} S_{2,1}^1 & S_{2,1}^2 & \cdots & S_{2,1}^N \\ S_{3,1}^1 & S_{3,1}^2 & \cdots & S_{3,1}^N \\ \vdots & & \cdots & \vdots \\ S_{M,M-1}^1 & S_{M,M-1}^2 & \cdots & S_{M,M-1}^N \end{bmatrix} \tag{1.16}$$

The rows of the above matrix show all segments' similarity values between two segments of Doppler data. Therefore, the mean value of each row (*MS*) can reflect the average similarity of the Doppler data under comparison. Then we get the similarity vector:

$$R^t = [MS_{2,1}^t, MS_{3,1}^t, \cdots, MS_{M,M-1}^t] \tag{1.17}$$

where t indicates the tth iteration in our clustering algorithm. Giving the vector R^t, we can calculate a similarity threshold $Thre^t$ as:

$$Thre^t = mean(R^t) + \alpha \times std(R^t) \tag{1.18}$$

where α is a constant value.

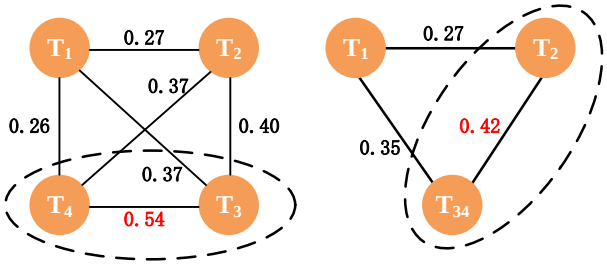


Figure 1.13 An example of iterative clustering algorithm.

We continue to use the example of Figure 1.12 to explain the effectiveness of our algorithm. At first, each tag is considered as one individual cluster. According to Equations 1.15, 1.16, 1.18, we can calculate the first similarity vector R^1 based on the measured data of the four tags. The results are shown in Figure 1.13. Each number in this figure means the average similarity between two tags. The threshold of the first iteration is 0.47, if $\alpha = 1$. We pick similarity values which are greater than the threshold: $MS^1_{3,4}$ in our example and put the two corresponding tags into one cluster. Hence Tag3 and Tag4 are clustered in the first iteration. The mean similarity R^2 and the threshold $Thre^2$ should be re-calculated in next iteration. When there is no similarity value is greater than the threshold, or only one cluster left in the end, the iterative algorithm is terminated.

1.3.2 Human Object Estimation

Human objects counting at a certain location is an crucial problem that has been researched within diverse research areas and application domains. Estimation of number of people has a extensive use in customers favor detection, stream of people location and quantification. By collecting these features of quite plenty of customers, we can reach a conclusion or get some experiences about customers purchase, shopping regional design or even people shopping habits. Currently, technique solutions based on three main points: mechanical barrier, sensing and imaging. The mechanical barrier based technology employs specific facilities to construct a one-way gate so that once a person access the gate, the facilities will move and count the number. Similarly, sensing based method utilize a break-beam sensor at the entrance of a one-way gate that if a person pass through it, the beam will be blocked. Teixeira[25] provide an extensive survey of methods for detecting human presence, count, location, etc. In contrast, the imager based technology uses pattern recognition techniques to identify the number of human beings in the videos.

We propose a system called R#, which employs existing passive Radio Frequen-

cy Identification(RFID) tags. According to the reflection, blocking and absorption of human beings, we can estimate the number of people by analyzing Radio Signal Strength(RSS) of the tag population. R# system has two major advantages. First, R# utilize off-the-shelf commercial RFID systems, which have been widely deployed in daily life. Secondly, R# is a device-free based solution that do not need to attach any device to human objects, which is suitable for the purpose of collecting information from quiet a lot of people.

However, we also face three key challenges. Firstly, we can extract limited types of information from commodity RFID devices. In addition, tag location and diversity lead to significant differences to the RSS value. Last by not the least, it is difficult to model a RF signal pattern.

To solve these challenges, R# employs a pre-deployed passive RFID system in supermarket, in which passive tags are attached to items. For a given region, a reader identifies a number (say 20) of passive tags using protocols in EPC C1G2 [26]. In particular, the tags are deployed in a line, as illustrated in Figure 1.14. The reader periodically collects the tag IDs and RSS values. For simplicity, we call an RSS sequence collected from such a period as an *observation* and each collection from a tag as a *sample* in the following sections.

R# works in three phases, *data preprocessing*, *feature extraction*, and *estimation*. In the first phase, R# preprocesses the raw RF signal data collected from the reader for later operations. In the second phase, R# extracts three features from the RSS values to establish a correlation between the number of human objects and the backscattered RF signal. Finally, R# employs machine learning techniques for human object estimation.

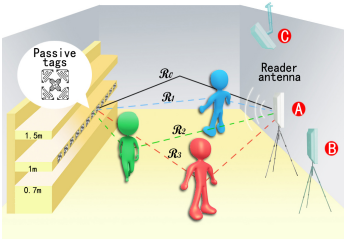


Figure 1.14 The multipath propagation of RF signal in R#.

1.3.2.1 Data Preprocessing

There are two operations in the preprocess phase, regrouping and interpolation.

Following the specification of EPC C1G2, tags are identified using a slotted ALOHA mechanism. That is, each tag randomly selects a time slot for reporting to the reader query and returning its ID. Hence the RSS values in one observation are 'timeline-based' but not grouped according to the tag ID, and hence cannot be used by the feature extraction scheme of R#. Thus, we regroup the RSS values by tag IDs in each observation.

Again due to the slotted Aloha mechanism, the slot in which a tag replies is ran-

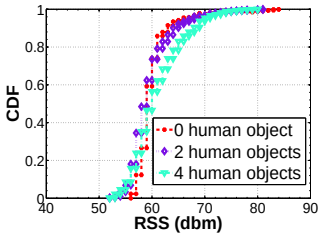


Figure 1.15 CDF of RSS

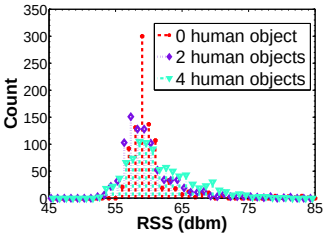


Figure 1.16 Distribution of RSS

domly and uncontrollably distributed, leading various numbers of samples from different tags in one observation. In addition, ambient factors, such as the shadowing, interference, and tag location, also influence the backscattered signal, resulting in different sampling periods to these tags. Suppose K is the maximum number of samples collected from a tag within one observation. To avoid the unfairness in the sampling and later processes, for each tag we use a Linear Interpolation method to virtually increase the number of its samples to K.

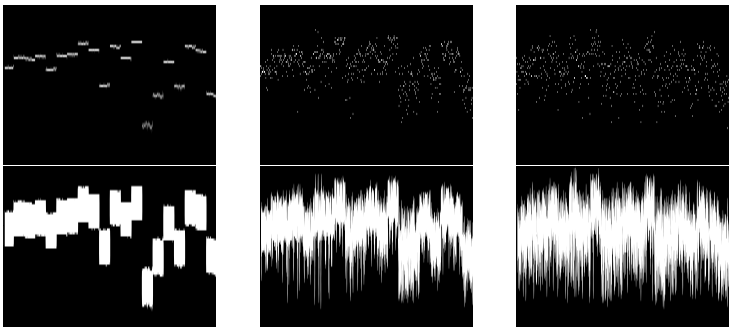
1.3.2.2 Feature Extraction

In R#, our expectation on a feature of the RF signal is that it monotonously increases if enlarging the number of human objects, or vice versa. To this end, we select three features, *entropy*, *size of dilated area* (*SDA*), and *mean squared error* (*MSE*). We present their extraction schemes as follows.

• ***Entropy.*** Given a constant transmission power of the reader (32dBm in our implementation) and fixed location for each tag, the collected RSS values vary within a range, denoted as $[R_{mi}, R_{ma}]$. the human interference may either strengthen or offset the overlapped signal at the reader antenna. Correspondingly, the R_{mi} and R_{ma} may change as well.

We setup a mathematical abstraction on the RSS values in an observation. If we use a random variable to represent them, those values are actually a distribution of this variable. As shown in Figure 1.15 and Figure 1.16, the distributions with different numbers of moving human objects are distinguishable. Inspired by this, we attempt to apply an information entropy based scheme to reflect the distribution of this variable, and hence yield the first *entropy* feature for R#.

According to the information theory, the entropy is a measurement of the uncertainty for a random variable. By monitoring the entropy of different observations, we can measure the degree of their disparity or concentration [27]. To calculate the entropy of an observation, we first establish the empirical distribution for this observation. The RSS range is divided into N bins with an equalized length (BL), where $N = \lceil (R_{ma} - R_{mi})/BL \rceil$. Let i denote the bin ID, $i < N$, x_i denote the number of RSS values falling into the ith bin, and $p_i = x_i / \sum_{i=1}^{N} x_i$ denote the probability that x_i RSS values fall into the ith bin. According to entropy theory, the discrete



(a) 0 human object (b) 2 human objects (c) 4 human objects

Figure 1.17 Binary visualization and dilation results of the observation of 0, 2, 4 human objects. When the number of human objects increases, the size of dilated area is enlarged.

entropy of an observation can be calculated as:

$$E(X) = -\sum_{i=1}^{N} p_i \cdot \log(p_i) \tag{1.19}$$

We extract the second feature from the collected RSS values by using a Morphological Image Processing based Scheme (MIPS). This feature is firstly discovered from the RSS data visualization. Intuitively, the higher the RSS variance, the larger the area the points could covered. To formalize this observation,

we first binary-visualize an observation with the following operations.

To simplify the subsequent processes, we extend the range of $[R_{mi}, R_{ma}]$ to $[R_{mi}^{'}, R_{ma}^{'}]$. Since the RSS resolution of our RFID readers is 0.5 dBm, we then normalize each RSS value x with the operation $(x - R_{mi}^{'}) \times 2$. Every x is within $[0, (R_{ma}^{'} - R_{mi}^{'}) \times 2]$.

We introduce a two-dimensional array $R_{l \times c}$ and initialize its elements as zero. Here l is the length of the observation and c is $(R_{ma}^{'} - R_{mi}^{'}) \times 2$. We then set all elements $R(k, x_i)$ to 1, where $1 \leq k \leq l, 1 \leq i \leq c$. Note that there is only one '1' in every column of $R_{l \times c}$. Each element in this array represents a pixel. Let '1' denote a white pixel and '0' denote a black pixel. Then we obtain a binary image. For example, the upper half parts of Figure. 1.17(a), (b), and (c) show the results of performing the binary visualization operation on the observations with 0, 2, and 4 human objects, respectively.

The core operation of MIPS is the morphological dilation, which makes a target in an image 'grown' or 'fatten'. This operation is based on two fundamental morphological operations, *reflection* and *translation*. The reflection of a set A is defined as: $\hat{A} = \{w | w = -a, a \in A\}$. The translation from a set A to a point $z = (z_1, z_2)$ is defined as: $(A)_z = \{c | c = a + z, a \in A\}$. To formalize, dilating a set A by using a set B is expressed as: $A \bigoplus B = \{z | ((\hat{B})_z \bigcap A) \neq \}$, where B is the *structuring*

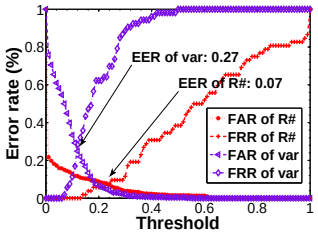
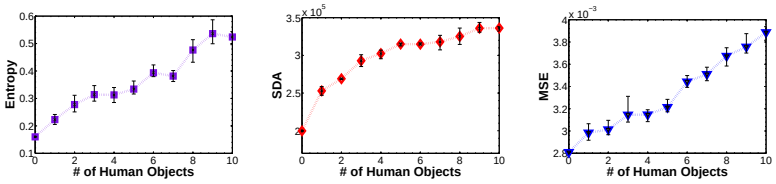


Figure 1.18 The error rate of R# vs. the error rate of variance-based method.



(a) Entropy (b) SDA (c) MSE

Figure 1.19 Three features used by R#.

element, which decides the degree of dilation. Generally, B is a set of '1's with a specific shape, such as a 'line', 'diamond', and the like. When conducting a dilation operation, B will translate on the entire image region of A, and exam which position is overlapped with its '1's. The dilation result is a set with all these overlapped positions. The lower parts of Figure 1.17 (a), (b), and (c) demonstrate the dilation results of upper original binary images. We can see that after dilation, white pixels get connected and the area of them is extended. We find that the area of white pixels can well represent the feature of observations. That is, when there are more human objects in the surveillance region, the area of white pixels becomes larger in the dilated image.

After the dilation, we find that the image has many burrs, which are probably derived from some outliers. To eliminate these thin protrusions, we conduct an *open operation*. The open operation is a combination of dilation and erosion operations. Different from the dilation, erosion can shrink or diminish a target in an image. In the open operation of R#, an erosion is followed by a dilation. An open operation often makes the edge of an object smooth, fills the gap, or eliminates the burr. Hence, the purpose of performing open operations for R# is to remove the noise and outliers from an observation. We define an erosion on a set A by using a set B as: $A \ominus B = \{z|(\hat{B})_z \subseteq A\}$. Hence, the open operation is symbolically expressed as: $A \circ B = (A \ominus B) \oplus B$, where B is a structuring element. After the open operation, we calculate the area of white pixels as the second feature.

• ***MSE.*** Based on the dilated image $f(x, y)$, we introduce another independent feature, the Mean Squared Error (MSE). By compressing $f(x, y)$ using Discrete Cosine Transform (DCT) followed by an de-compression, we obtain an recovered

image $\hat{f}(x, y)$. We find that the *fidelity* between f and $\hat{f}$ can be used for characterizing the observations. Specifically, the Mean Squared Error (MSE) represents the fidelity feature. MSE is normally used to measure the information loss. We notice that when there are more human objects moving in the area, the image of the observation will be more complicated. Correspondingly, when doing DCT compression, more information will lose.

Considering a $M \times N$ image $f(x, y)$, the forward DCT $T(u, v)$ can be expressed as:

$$T(u,v) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} f(x,y) g(x,y,u,v)$$

$$g(x,y,u,v) = a(u)a(v)k(u,v)$$

$$k(u,v) = \cos\left[\left(\frac{(2x+1)u\pi}{2M} \cos\left(\frac{(2y+1)v\pi}{2N}\right)\right)\right]$$

$where,$

$$a(u) = \begin{cases} \sqrt{\frac{1}{M}} & \mu = 0 \\ \sqrt{\frac{2}{M}} & \mu = 1, 2, ..., M\text{-}1 \end{cases}$$

$$a(v) = \begin{cases} \sqrt{\frac{1}{N}} & \upsilon = 0 \\ \sqrt{\frac{2}{N}} & \upsilon = 1, 2, ..., N\text{-}1 \end{cases}$$

DCT compression is an invertible transform and can be easily recovered. A brief procedure of DCT compression is as follows: we first divide the original image into some 8×8 subimages. We then use the DCT to represent these subfigures, and discard a part of the coefficients achieved (85% in our implementation). When recovering the image, we run the reverse DCT on the intercepted coefficient matrix. Although the discarded coefficients have little visual influence on the recovered image, they still incur MSE. The incurred MSE between $f(x, y)$ and $\hat{f}(x, y)$ is:

$$MSE = \left(\frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} \left(\hat{f}(x,y) - f(x,y)\right)^2\right)^{1/2}$$

1.3.2.3 **Machine Learning based Estimation**

Before the design of estimation mechanism, we exam the effectiveness of three features.

We arrange 0∼10 volunteers to move arbitrarily in the region and check the trend of features with varying the number of volunteers. The result is reported in Figure 1.19 (a), (b), (c). It reveals an interesting insight. Each of the three features shows a good ability of distinguishing different numbers of human objects in some ranges, $e.g.$, 0∼4 persons for SDA and 6∼10 persons for MSE. While the entropy feature shows

an instable variance upon certain numbers of persons. Fortunately, the three features exhibit the complementarity with each other, which inspires us to jointly use them in the classifier of R#.

We adopt Naive Bayes method from WEKA as our machine learning classifier. In real implementation, it is necessary to tune the classifier parameters to optimize the estimation accuracy. To achieve this goal, we organize above experiment results into two datasets, one for training and another for test. We then perform a 10-fold cross validation on the datasets to determine the key parameter of classification.

1.4 Conclusion

Indoor wireless sensing has spawned numerous applications in a wide range of living, production, commerce, and public services. The increase of mobile and pervasive computing has sharpened the need for accurate, robust, and off-the-shelf indoor action recognition schemes. We explore the properties of wireless signals and proposed a robust indoor daily sensing framework using the big data technology. We can achieve relatively high recognition accuracy for a set of similar daily actions insensitive to location, orientation, speed, and shopping habits.

Considering the advantages and limitations of wireless signals, we will put our effort into exploring how to gain more robust and subtle indoor sensing in future work, and we think vision-based methods and Wi-Fi based methods may be a good complement to each other using crowdsourcing techniques in the age of big data.

References

1 Hassanieh, H., Adib, F., Katabi, D., and Indyk, P. Faster gps via the sparse fourier transform, in *ACM Mobicom'2012*.

2 Bahl, P. and Padmanabhan, V. Radar: An in-building rf-based user location and tracking system, in *IEEE INFOCOM'2000*.

3 Youssef, M. and Agrawala, A. (2008) The horus location determination system. *Wireless Networks*.

4 Niculescu, D. and Nath, B. Ad hoc positioning system (aps) using aoa, in *IEEE INFOCOM'2003*.

5 Chintalapudi, K.K., Iyer, A.P., and Padmanabhan, V.N. Indoor localization without the pain, in *ACM Mobicom'2010*.

6 Rai, A., Chintalapudi, K.K., Padmanabhan, V.N., and Sen, R. Zee: zero-effort crowdsourcing for indoor localization, in *ACM Mobicom'2012*.

7 Yang, Z., Wu, C., and Liu, Y. Locating in fingerprint space: wireless indoor localization with little human intervention, in *ACM Mobicom'2012*.

8 Sen, S., Choudhury, R.R., and Nelakuditi, S. Spinloc: spin once to know your location, in *ACM HotMobile'2012*.

9 Sen, S., Radunovic, B., Choudhury, R.R., and Minka, T. You are facing the mona lisa: spot localization using phy layer information, in *ACM Mobisys'2012*.

10 Wu, K., Xiao, J., Yi, Y., Gao, M., and Ni, L.M. FILA: Fine-grained indoor localization, in *IEEE INFOCOM'2012*.

11 Nandakumar, R., Chintalapudi, K.K., and Padmanabhan, V.N. Centaur: locating devices in an office environment, in *ACM Mobicom'2012*.

12 Liu, H., Gan, Y., Yang, J., Sidhom, S., Wang, Y., Chen, Y., and Ye, F. Push the limit of wifi based localization for smartphones, in *ACM Mobicom'2012*.

13 Zhang, Z., Zhou, X., Zhang, W., Zhang, Y., Wang, G., Zhao, B., and Zheng, H. I am the antenna: Accurate outdoor ap location using smartphones, in *ACM Mobicom'2011*.

14 Leordeanu, M. and Hebert, M. A spectral technique for correspondence problems using pairwise constraints, in *IEEE ICCV'2005*.

15 Cho, M., Lee, J., and Lee, K. (2010) Reweighted random walks for graph matching. *Computer Vision–ECCV*.

16 Ichino, M. and Yaguchi, H. (1994) Generalized Minkowski metrics for mixed feature-type data analysis. *IEEE Transactions on Systems, Man, and Cybernetics*.

17 Von Luxburg, U. (2007) A tutorial on spectral clustering. *Statistics and computing, Springer*.

18 Garsia, A.M. and Remmel, J. (1985) Shuffles of permutations and the Kronecker product. *Graphs and Combinatorics*.

19 Matthews, C. (2012), Future of Retail: How Companies Can Employ Big Data to Create a Better Shopping Experience, http://business.time.com/2012/08/31/future-of-retail-how-companies-can-employ-big-data-to-create-a-better-shopping-experience/.

20 Li, C., Dai, H., and Li, H. (2009) Adaptive Quickest Change Detection with Unknown Parameter, in *Proceedings of IEEE ICASSP*.

21 Page, E. (1954) Continuous Inspection Schemes. *Biometrika*, pp. 100–115.

22 Wang, J. and Katabi, D. (2013) Dude,

Where's my Card?: RFID Positioning that Works with Multipath and Non-line of Sight, in *Proceedings of ACM SIGCOMM*.

23 Tse, D. (2005) *Fundamentals of Wireless Communication*, Cambridge university press.

24 Dobkin, D.M. (2008) *The RF in RFID - Passive UHF RFID in Practice*, Elsevier.

25 Teixeira, T., Dublon, G., and Savvides, A. (2010) A Survey of Human-sensing: Methods for Detecting Presence, Count, Location, Track, and Identity. *ACM Computing Surveys*, **5**.

26 EPCglobal (2008) *Specification for RFID Air Interface EPC: Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz-960 MHz*.

27 Zhang, J., Qin, Z., Ou, L., Jiang, P., Liu, J., and Liu, A. (2010) An Advanced Entropy-based DDOS Detection Scheme, in *Proceedings of IEEE ICINA*.