Device-Agnostic Wi-Fi Fingerprint Positioning for Consumer Applications

Brett Fischler*, Daniel Griffin*, Tyler Lubeck*, Kenneth Hunter Wapman*, Mai Vu†
Department of Computer Science* & Department of Electrical and Computer Engineering†, Tufts University, MA, USA
Brett.Fischler@gmail.com, {Daniel.Griffin, Hunter.Wapman}@tufts.edu, Tyler@TylerLubeck.com, MaiVu@ece.tufts.edu

Abstract—**There is a growing need to position wireless devices in the real world for applications such as navigation, emergency location services, and contextual advertisements. Though GPS and the cellular network provide viable outdoor accuracy, these approaches are unsuited for indoor positioning. We propose a high-accuracy Wi-Fi Fingerprint-based indoor positioning system ideal for consumer applications. This system can be implemented in any Wi-Fi-enabled environment without modifying the site. We propose several optimized distance metrics as well as two novel environment-based methods, a density penalty factor and a floor preprocessing technique, to improve the accuracy of the Weighted K-Nearest Neighbor algorithm. Our implementation requires only thirteen minutes of site-survey time per 100 square meters and has an average positioning accuracy of 2.66 meters, which is sufficient for most practical applications.**

I. Introduction

As the number of wireless computing devices grows, there is a corresponding need to integrate these devices with the physical world to provide contextually relevant services. The device's position can be particularly valuable to a consumer or business. A consumer could locate a friend within a building; a business could provide location-dependent advertisements.

Global Navigation Satellite Systems (GNSS), such as GPS, achieve acceptable accuracy outdoors, but perform poorly indoors, under No-Line of Sight (No-LoS) conditions [1]. GNSS relies on LoS conditions to determine the position of a wireless device through triangulation or trilateration. The lack of LoS conditions indoors poses a serious challenge to these methods, requiring specialized hardware to achieve acceptable accuracy [2].

Most indoor positioning systems rely on Wi-Fi networks, as they are widely available and supported by most consumer hardware. This availability makes Wi-Fi networks the lowest cost option for positioning, since the required infrastructure already exists. When using Wi-Fi for positioning, there are two viable approaches: site-simulation using a signal-propagation model [3], and manual site-survey [4]. Site-simulation allows for rapid site integration but requires prior knowledge of the location of wireless access points (WAPs). Conversely, site-survey makes no assumptions about the environment but requires more time to integrate each site.

Several indoor positioning systems have been successfully implemented. The RADAR positioning system utilizes fingerprinting, achieving an average accuracy of 3m to 5m within a controlled environment [5]. Another method, based on site-simulation, achieves an average accuracy of 1.46m by incorporating additional building information [3]. While these systems are successful, they were implemented in highly-controlled environments using site information that is not commonly available in commercial settings.

We design and implement an indoor positioning system using Wi-Fi signals based on the site-survey method for use within a fixed commercial environment. Our system uses the nearest neighbor algorithm to estimate the position of smart phone devices. We further propose several novel distance metrics, as well as two environment-based algorithm augmentations to improve accuracy. Our system requires a low site-survey time and achieves a competitive level of accuracy, despite using consumer hardware.

This paper is organized as follows: we first state the aims of our work, followed by a discussion of how Fingerprints (FP) are recorded and how the FP database is created. We then discuss our positioning algorithm, describing the distance metrics and algorithm refinements, before presenting the performance of our implementation and comparing it against other systems. Finally, we discuss areas for future work.

II. Indoor Positioning System

The basic requirements for this indoor positioning system are to provide room-level precision, widespread coverage, and real-time feedback. An indoor positioning system for use in commercial settings must additionally be usable on a variety of devices, serve multiple users simultaneously, and scale with a large user base. Further, the system must function without extensive knowledge of or modifications to the site, and must be cheap to implement.

We design and implement an indoor positioning system which demonstrates the viability of an indoor positioning application in an environment facing constraints similar to the commercial setting described above. Our approach uses Received Signal Strength (RSS) measurements from a Wi-Fi network to position a Wireless Device (WD). RSS measurements can be effectively used for wireless indoor positioning by utilizing a Fingerprint (FP) map and a classifying algorithm. Each FP has a unique position on a floor plan, as well as a set of paired MAC addresses and RSS values.

Our application follows the client-server model, which uses a single central server to serve a large number of clients [6]. The server manages the map database and uses HTTP to respond with our position estimation upon receiving a request containing an FP set. This approach is hardware-agnostic,

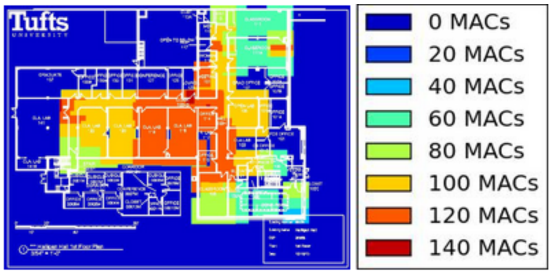

Fig. 1. Visible unique MAC addresses in accessible areas on the first floor of Halligan Hall

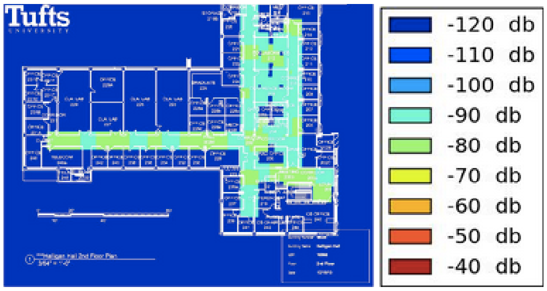

Fig. 2. Average RSS contour map in accessible areas on the second floor of Halligan Hall

easily scales to multiple users, and also allows simultaneous FP recording on multiple WDs, significantly speeding up the site-survey process.

We provide quantitative analyses of how different decisions, such as the number of scans and floor-detection preprocessing, affect the runtime and accuracy of our positioning application. Our application could be modified for higher accuracy at the cost of increasing mapping time, and vice versa.

III. Floor Mapping

We now describe the building environment and discuss the methods used to generate the FP database.

A. Building Environment

We mapped two floors of Halligan Hall, the Computer Science and Electrical Engineering building at Tufts University. The average accessible area of each floor is 600m^2. The floor plans were available, but the locations of wireless access points were not disclosed. Furthermore, there are several inaccessible areas in the building, such as restrooms and offices. As such, this building models the commercial constraints described above. Since the locations of the access points were unknown, contour maps were generated in order to examine the number of unique MAC addresses visible at each accessible location, as well as the average RSS values. Unsurveyed areas were assumed to have 0 visible MAC addresses and have an average RSS of -120 dB.

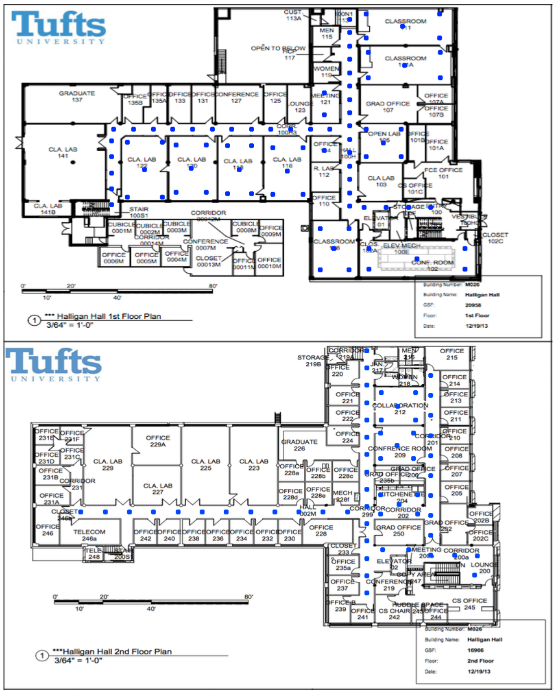
Tufts UNIVERSITY
*** Halligan Hall 1st Floor Plan
3/64" = 1'-0"
Tufts UNIVERSITY
***Halligan Hall 2nd Floor Plan
3/64" = 1'-0"

Fig. 3. Mapped locations in Halligan Hall on both floors

TABLE I
Floor Recording Times (average 1.5 sec/scan)

Scans per Position	2	10	**30**	60	100
Scan Time per Floor (Hours)	0.24	1.2	**3.5**	7.0	12

We performed the mapping of both visible MAC addresses and average RSS values on all floors. Figure 1 shows the MAC addresses visible on the first floor of Halligan Hall, while Figure 2 shows the average RSS values on the second floor of Halligan Hall.

B. Fingerprint Logging Methodology

To build the FP database, the average of thirty RSS scans is logged for each mapped location, indicated in Figure 3. These locations were chosen manually through an examination of the floor plans and knowledge of the facility's accessibility. Each recording is stored as four values: a MAC address, an average RSS value, a standard deviation, and an association with a location on a map. The recordings were taken at locations estimated with reference to the points in Figure 3.

In deciding the number of scans for each chosen location, we consider three factors: time, RSS precision, and MAC address coverage. We test a sample of scan values between one and one hundred using a single WD. The average Euclidean distance between each of the scans is under 1 dB when using σ_E (Equation 2). Given the similarity of RSS values independent of the number of scans per location, we focus on the remaining two factors: time and MAC coverage.

Increasing the number of scans improves MAC coverage, but involves a time tradeoff, as scanning is the most time-consuming part of site survey. Different WDs have dramatically different scan times, ranging from 0.20 to 2.0 seconds per scan. Table I lists the approximate time to record a floor

TABLE II
MAC COVERAGE BY SCAN NUMBERS

MAC Coverage (%)	2 Scans	10 Scans	**30 Scans**	60 Scans	100 Scans
Trial 1	85.3	91.2	**96.1**	100	100
Trial 2	88.7	90.3	**96.8**	100	100
Trial 3	74.4	79.5	**89.2**	97.6	100
Trial 4	59.6	84.3	**95.5**	97.8	100
Average	77.0	86.3	**94.4**	98.9	100

for each number of scans, assuming the worst-case scan time. The number of locations on each floor is based on the average number of locations facing only one direction.

We choose a number of scans for site surveying which balances MAC coverage against worst-case device runtime. Given the similarity of MAC coverage, as shown in Table II, for 30 scans and 100 scans and the runtimes of each, we opt for 30 scans.

When recording the FP database, four devices were utilized: the HTC One m7 (Wi-Fi: IEEE 802.11 a/ac/b/g/n), LG Nexus 5 (Wi-Fi: IEEE 802.11 a/ac/b/g/n), Samsung Galaxy S3 (Wi-Fi: Wi-Fi 802.11 a/b/g/n), and Samsung Galaxy Nexus (Wi-Fi: IEEE 802.11 a/b/g/n). Because of different phones' hardware and processing capabilities, the average difference between each phone's recording of the same physical location is 6.3 dB, using σ_E (Equation 3). This value is notably high in comparison to the average difference of 1.7 dB between four surveys made using the same phone. Despite this inconsistency among devices, our final application provides acceptable levels of accuracy. This decision to use multiple devices reflects our desire to build a device-agnostic system. Further, supporting a variety of device types significantly speeds up the mapping process, as parallel mapping is possible.

A final consideration is the direction of the recording device, as direction significantly affects measurements [7]. We found an average difference of 4.0 dB between four surveys made with the same WD in the same location, recorded at 90 degree angles from each other. We decided to map facing a single direction, as this reduced mapping time by a factor of four. Our implementation does, however, support mapping with multiple directions for applications requiring higher accuracy.

The set of FPs collected during mapping is used as the database for our positioning algorithm. Upon receiving a positioning request containing FP A, the algorithm compares A with each FP entry in the database, using the FP entries most similar to A to estimate A's position on the map.

IV. POSITIONING ALGORITHMS

A number of classifiers are suitable for determining the location of an unknown point given an RSS Fingerprint map [8][4]. In our implementation, we use the deterministic Weighted K-Nearest Neighbor (WKNN) method to perform position estimation, as it experimentally improves our algorithm's performance over the standard KNN method. In this section, we discuss the WKNN method and propose several distance metrics to improve its performance.

A. Weighted K-Nearest Neighbor Algorithm

Consider a database S of mapped FPs, in which $N = [n_1, n_2, ..., n_K]$ is the set of K mapped FPs nearest to some recorded point ρ measured by a distance metric σ, which quantifies distance as a function of RSS. The estimated location $\hat{\rho}'$ of ρ is calculated as:

$$\hat{\rho}' = \sum_{i=1}^{K} \frac{\omega_i}{\omega_{total}} n_i', \quad w_i = \sigma(n_i, \rho)^{-1} \tag{1}$$

where ω_i is the weight assigned to n_i, n_i' is the recorded position corresponding to n_i, ω_{total} is the sum of the weights of all FPs in N, and $\sigma(n_i, \rho)$ is the signal distance between n_i and ρ using distance metric σ. The K-Nearest Neighbor (KNN) method is a special case of WKNN which uses equal weights for FPs in N. We discuss the specific distance metrics we implement in the next section. We use a K value of 3 because it has been shown to have a higher accuracy than the Nearest Neighbor (NN) method, where K is 1 [3].

B. Distance Metrics

WKNN is a very flexible classifier which can incorporate a variety of distance metrics. In this section, we discuss the four distance metrics we use to quantify the difference between two RSS measurements: Euclidean distance (σ_E), penalty-weighted Euclidean distance ($\sigma_{E'}$), inverse Jaccard coefficient ($\sigma_{J^{-1}}$), and a linear combination of $\sigma_{E'}$ and $\sigma_{J^{-1}}$ ($\sigma_{E'+J^{-1}}$). For consistency in our definitions, we here define two FPs A and B with RSS vectors $a = [a_1, a_2, ..., a_m]$ and $b = [b_1, b_2, ..., b_n]$, respectively.

1) Euclidean Distance (σ_E): In the ideal case where the sets of MAC addresses seen by A and B are identical, the Euclidean distance between A and B is:

$$\sigma_{E_{\text{ideal}}}(A, B) = (\sum_i |a_i - b_i|^2)^{\frac{1}{2}} \tag{2}$$

where the MAC address of a_i corresponds to that of b_i. However, in a realistic environment, there are often several MAC addresses seen by A and not by B, and vice-versa. One proposal to handle these mismatches assigns a threshold value equal to the lowest recorded RSS measurement in the FP database to missing RSS measurements [9]. By this approach, the Euclidean distance between A and B is:

$$\begin{aligned}\sigma_E(A, B) = (&\sum_i |a_i - b_i|^2 \\ &+ \sum_{j, b_j \notin b} |a_j - I_{min}|^2 \\ &+ \sum_{k, a_j \notin a} |I_{min} - b_k|^2)^{\frac{1}{2}}\end{aligned} \tag{3}$$

where I_{min} is the lowest recorded RSS measurement in the FP database. In most cases, this approach has the desired effect of increasing the distance between two locations with non-matching MAC addresses.

2) Penalty-Weighted Euclidean Distance ($\sigma_{E'}$): The σ_E distance metric can produce false matches in the case where A sees a MAC address with a low RSS value and B does not see that MAC address at all. To address this issue, we propose a penalty-weighted Euclidean distance metric:

$$\begin{aligned}\sigma_E(A,B) = (&\sum_i |a_i - b_i|^2 \\ &+ \alpha \times \sum_{j,b_j \notin b} |a_j - I_{min}|^2 \\ &+ \alpha \times \sum_{k,a_j \notin a} |I_{min} - b_k|^2)^{\frac{1}{2}}\end{aligned} \tag{4}$$

where α is a constant value. By introducing this new coefficient, our algorithm assigns a different weight to mismatching MAC addresses. Through iterative testing, we determine the best value of α to be 0.25. This increases the distance for locations with non-matching MAC addresses but limits the effect of false RSS matches.

3) Inverse Jaccard Coefficient ($\sigma_{J^{-1}}$): The Jaccard coefficient is equal to the number of elements in the intersection divided by the number of elements in the union [10]. The larger the Jaccard coefficient, the greater the similarity between the two sets. We use the inverse Jaccard coefficient with our implementation of WKNN, as smaller values indicate higher similarity within the WKNN algorithm. The inverse Jaccard coefficient is:

$$\sigma_{J^{-1}}(A,B) = \left(\frac{|a \cap b|}{|a \cup b|}\right)^{-1} \tag{5}$$

where a and b here represent the set of seen MAC addresses instead of average RSS values. The inverse Jaccard coefficient is small when the MAC addresses have high overlap, which indicates high similarity within the WKNN algorithm. As this metric only examines MAC addresses, it is a quick yet coarse method to determine distances between locations.

4) Linear Combination of $\sigma_{E'}$ and $\sigma_{J^{-1}}$ ($\sigma_{E'+J^{-1}}$): Since the penalty-weighted Euclidean distance metric and the inverse Jaccard coefficient both reveal valuable information about the difference between RSS vectors, we propose a new distance metric that is the linear combination of these two measures:

$$\sigma_{E'+J^{-1}}(A,B) = \beta_1 \times \sigma_{E'}(A,B) + \beta_2 \times \sigma_{J^{-1}}(A,B) \tag{6}$$

where β_1 and β_2 are constants. Through iterative testing, we determine the best values of β_1 and β_2 to be 1.0 and 0.45, respectively. This result utilizes the effectiveness of the penalty-weighted Euclidean distance metric at detecting differences in RSS values for common MAC addresses, as well as the ability of the inverse Jaccard metric to penalize unique MAC addresses.

V. Map-Aware Algorithm Refinement Techniques

In addition to the distance metrics we use in the WKNN algorithm, we make several refinements to our algorithm that improve the performance of our position finder.

A. Floor Detection Preprocessing

Before running WKNN with a more complex distance metric, we first run it using only the inverse Jaccard coefficient in order to determine the floor of our test point. Calculating the inverse Jaccard coefficient between two sets of MAC addresses does not require RSS values, allowing for faster performance compared to calculating Euclidean distance. Once we determine the floor of the query point, we find its coordinates by running WKNN with composite distance σ_C (defined in section V-C), only iterating over FPs in the database on the identified floor. While this change does not yield significant time improvements in our two-floor test environment, its benefits become more evident in an environment with a large number of floors and buildings.

B. Density-Penalizing Neighbor Weighting Factor

We propose a novel neighbor-weighting factor to improve accuracy when locating positions in sparsely mapped areas. This factor improves accuracy when locating points in corners and near the edges of the mapped areas. We define the neighbor factor W_d for some point d as:

$$\begin{gathered} W_d = \phi \times \sum_i F(d, di) \\ F(d,di) = \{ \begin{array}{cc} 1: & \sigma_E(d,d_i) < \Gamma \\ 0: & \text{otherwise} \end{array} \end{gathered} \tag{7}$$

where ϕ is a constant, d_i is the i^{th} fingerprint in the database, and Γ is the neighbor distance threshold. Through iterative testing, we determine the optimal value of ϕ to be 0.08 and Γ to be 20 meters. The minimum neighbor count is always one, as a point is always its own neighbor; this ensures a non-zero value in all cases. This factor increases the importance of points with low neighbor counts, helping improve accuracy in areas that are sparsely mapped.

C. Composite Distance Metric σ_C

Our WKNN ultimately measures distance as a linear combination of the penalty-weighted Euclidean distance, $\sigma_{E'}$, the inverse Jaccard coefficient $\sigma_{J^{-1}}$, and the neighbor factor, W, as follows:

$$\sigma_C(A,B) = \beta_1 \times \sigma_{E'}(A,B) + \beta_2 \times \sigma_{J^{-1}}(A,B) + W_B \tag{8}$$

where β_1 and β_2 are constants, chosen as in Equation 6. This composite distance metric combines the RSS comparisons of the penalty-weighting Euclidean distance metric, the MAC comparisons of the inverse Jaccard coefficient, and the density weightings of the neighbor factor. We evaluate this metric in the next section and compare it to each of the component distant metrics.

VI. Performance Evaluation

Through empirical evaluation, we determine our positioning application has an average error of 2.66m, meeting the requirements of most commercial applications. We break our results and analysis into three sections. First, we discuss our testing methodology. Second, we compare the four discussed

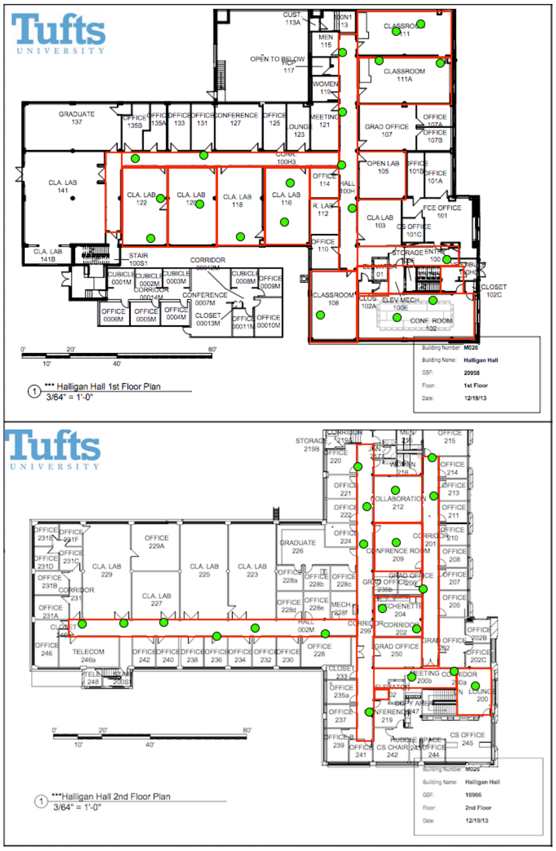

Fig. 4. Test points on Halligan Floor 1 (Top) and Floor 2 (Bottom); red boxes indicate accessible regions, and green dots indicate test points

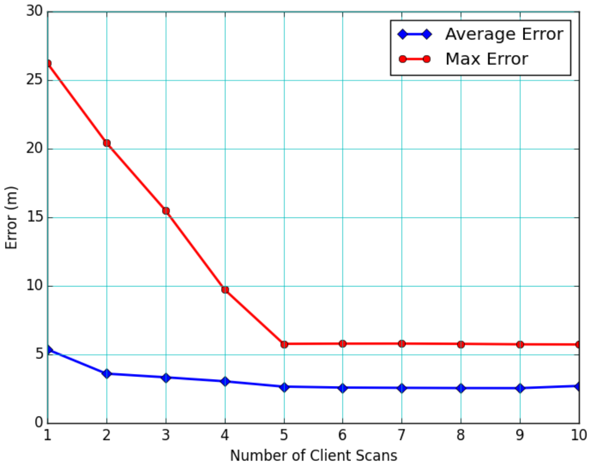

Fig. 5. Accuracy of the composite distance metric for various numbers of positioning scans

TABLE III
DISTANCE METRIC ACCURACY

Distance Metric	Average Error (m)	Standard Deviation (m)
Inverse Jaccard Coefficient	4.75	3.49
Euclidean distance	3.56	2.31
Penalty-Weighted Euclidean distance	2.87	2.10
Linear Combination of $\sigma_{E'}$ and $\sigma_{J^{-1}}$ $(\sigma_{E'+J^{-1}})$	2.78	1.76
Composite Distance Metric	2.66	1.44

distance metrics: Euclidean distance (σ), penalty-weighted Euclidean distance ($\sigma_{E'}$), inverse Jaccard coefficient ($\sigma_{J^{-1}}$), and a linear combination of $\sigma_{E'}$ and $\sigma_{J^{-1}}$ ($\sigma_{E'+J^{-1}}$). Finally, we analyze our map-aware algorithm refinements: Floor Detection Preprocessing and Density-Penalizing Neighbor Weighting.

A. Testing Methodology

We recorded a set of points for the purpose of fine-tuning our coefficients, the coefficient training set, as well as a second set of points for performance testing, the testing set. Both sets contain points randomly generated by the Mitchell's Best Candidate II algorithm [11], which efficiently creates a Poisson Distribution of points within the mapped regions, as displayed in Figure 4. All FPs in the testing set were recorded using the HTC One m7.

The accuracy of our positioning algorithm is dependent on the number of scans the WD makes for each test point, as shown in Figure 5. Increasing the number of scans beyond five does not significantly improve the performance of our algorithm; therefore, we choose to test our positioning application against FPs gathered with five scans.

B. Distance Metric Comparisons

Figure 6 shows the accuracy of our WKNN algorithm using the four distance metrics discussed in Section IV-B. As the graph shows, the standard Euclidean distance metric performs significantly better than the inverse Jaccard coefficient, yielding an average error of 3.56m compared to 4.75m. Using the penalty-weighted Euclidean distance metric reduces the error to 2.87m. A linear combination of the two metrics only improves the performance slightly, bringing the average error to 2.78m. However, this linear combination reduces the maximum error to 9.70m, which is 21.1% smaller than that of the penalty-weighted Euclidean distance metric alone. The composite distance metric is discussed in the next section.

C. Map-Aware Algorithm Refinement

We add our Density-Penalizing Neighbor Weighting technique as a parameter to our composite distance metric (σ_C) in Equation 8. The accuracy of this metric can be seen in Figure 6. Overall, neighbor weighting decreases the average error from 2.78m to 2.66m. For points with less than 5m error, neighbor weighting increases the average error slightly, by 0.15m. However, the technique reduces the maximum error by 40.5%, from 9.70m to 5.77m, and the average error of points with more than a 5m error by 50.1%, from 7.12m to 3.55m. We consider this slight increase in average error for points with error less than 5m well worth the dramatic reduction in maximum error. The standard deviation of the error shown in Table III clearly reflects this improvement.

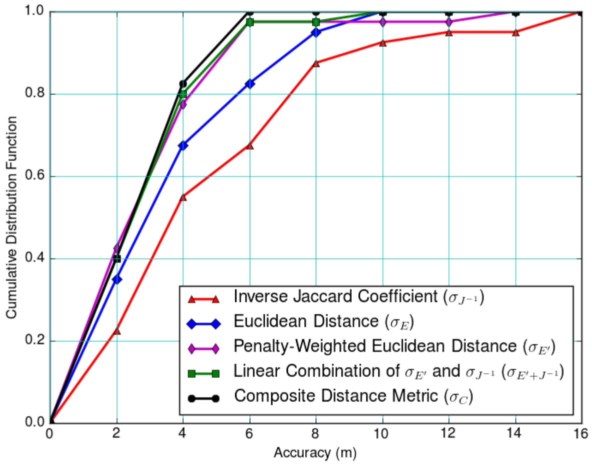

Fig. 6. Cumulative distribution of positioning accuracy for various WKNN distance metrics (K=3)

Our novel Floor Detection Preprocessing technique does not improve runtime results in our testing environment, but our simulation of a 100 floor database indicates this technique would reduce positioning time by 39.1%, from 6.89 seconds to 4.20 seconds. This time reduction results from the reduced number of penalty-weighted Euclidean distance calculations required once a floor has been identified. This performance improvement would be extremely significant in a large-scale commercial application.

VII. Discussion

Our positioning application has runtime and accuracy comparable to existing FP-based systems [2]. Although there are fingerprinting positioning systems with higher accuracy, these often utilize application-specific hardware, whereas our application operates with consumer hardware and assumes no prior site information.

Microsoft RADAR is an example of a system using KNN that achieved an average error of 3m to 5m using RF signals [5]. This system was implemented using one mapping device on a single floor that had three prearranged base stations. Our system achieves better accuracy with consumer hardware in a commercial-like setting. Subsequent iterations of RADAR incorporated continual user-tracking and multiple RSS maps for different environments, such as time of day [12]. The same improvements could be applied to our system.

Another very recent system uses improved double-peak Gaussian distributions and a probabilistic approach to determine position [13]. This approach used a single device for mapping, and reported an average error of 1.5m. While this is a notably higher accuracy, the statistical analysis techniques used could be incorporated into our positioning application to improve accuracy.

Beyond increasing scan numbers, several other methods and additional data could be incorporated to further improve our system accuracy, including the use of RSS standard deviation, multiple directions in site-survey, WD inertia, and prior knowledge of the site. These options are left as future work.

VIII. Conclusion

We implement a positioning system addressing many of the obstacles to widespread indoor positioning. We propose two novel metrics for use with the WKNN algorithm: Density-Penalizing Neighbor Weighting and penalty-weighted Euclidean distance, as well an algorithm refinement for improved scalability: Floor Detection Preprocessing. Further improvement could be obtained by integrating other algorithm refinement techniques, such as Double Gaussian Distribution processing and the inclusion of user-feedback in the mapping process. Our positioning system is device-agnostic, able to serve multiple users, places no requirements on the site, and has an average accuracy of 2.66m, satisfying the requirements of most applications.

Acknowledgment

We would like to thank Professor Chris Gregg of Tufts University for his advice and encouragement.

References

[1] Lakmali, B.D.S. and D. Dias, 'Database Correlation for GSM Location in Outdoor and Indoor Environments', *4th Int'l Conf. on Information and Automation for Sustainability (ICIAFS).* Dec. 2008.

[2] H, Liu, H. Darabi, P. Banerjee, and J. Liu, 'Survey of Wireless Indoor Positioning Techniques and Systems', *IEEE. Trans. Systems, Man, and Cybernetics, Part C: Applications and Reviews, Vol. 37, No. 6, pp. 1067 - 1080,* Nov. 2007.

[3] A. Kokkinis, M. Raspopoulos, L. Kanaris, A. Liotta and S. Stavrou, 'Map-aided fingerprint-based indoor positioning', *IEEE 24th Int'l Symp. Personal Indoor and Mobile Radio Communications (PIMRC),* Sept. 2013.

[4] S. Saha, K. Chaudhuri, D. Sanghi, and P. Bhagwat, 'Location Determination of a Mobile Device Using IEEE 802.11b Access Point Signals', *IIEEE Wireless Communications and Networking Conf. (WCNC),*. Mar. 2003.

[5] P. Bahl and V. N. Padmanabhan, 'RADAR: An in-building RF-based user location and tracking system', *in Proc. IEEE INFOCOM.* Mar. 2000.

[6] R. Fielding, 'Architectural Styles and the Design of Network-based Software Architectures', *(Doctoral dissertation). University of California, Irvine.* 2000.

[7] A. H. Sayed, A. Tarighat, and N. Khajehnouri, 'Network-based Wireless Location: Challenges Faced in Developing Techniques for Accurate Wireless Location Information', *IEEE Signal Processing Magazine. Vol. 22, No. 4, pp. 24 - 40,*. July 2005.

[8] Tsung-Nan Lin and Po-Chiang Lin, 'Performance Comparison of Indoor Positioning Techniques Based on Location Fingerprinting in Wireless Networks', *Int'l Conf. on Wireless Networks, Communications and Mobile Computing,* June 2005.

[9] P. Kemppi and S. Nousiainen, 'Database Correlation Method for Multi-System Positioning', *IEEE 63rd Vehicular Technology Conference (VTC).* May 2006.

[10] D. Dunbar and G. Humphreys, 'A Spatial Data Structure for Fast Poisson-Disk Sample Generation', *Proc. ACM SIGGRAPH,*. July 2006

[11] J. Machaj, P. Brida, and R. Piche, 'Rank Based Fingerprinting Algorithm for Indoor Positioning', *Int'l Conf. on Indoor Positioning and Indoor Navigation (IPIN), Guimaraes, Portugal.* September 2011.

[12] P. Bahl, V. N. Padmanabhan and A. Balachandran, 'Enhancements to the RADAR user location and tracking system', *TechReport MSR-TR-2000-12, Microsoft Corporation.* Feb. 2000.

[13] L. Chen, B. Li, K. Zhao, C. Rizos, and Z. Zheng, 'An Improved Algorithm to Generate a Wi-Fi Fingerprint Database for Indoor Positioning', *Sensors Vol. 13, No. 8, pp. 11085-11096, DOI: 10.3390/s130811085,*. Aug. 2013.