Simple Algorithm Aggregation Improves Signal Strength Based Localization

Xiaoyan Li
xili@cs.lafayette.edu
Department of Computer Science
Lafayette College, Easton, PA 18042

Richard P. Martin
rmartin@cs.rutgers.edu
Department of Computer Science, Rutgers University
110 Frelinghuysen Rd, Piscataway, NJ 08854

Abstract—In this paper we propose using algorithm aggregation to improve signal strength based localization performance. Prior work comparing a spectrum of received signal strength (RSS) based localization algorithms concluded that their performance is strikingly similar and none of them manage to localize objects accurately all the time. We, however, show that despite such similarity in average localization accuracy, simple algorithm aggregation can improve performance. Specifically, comparing with point-based algorithms, the performance improvement ranges from 23% (when aggregating 2 algorithms) to 51% (when aggregating all 12 algorithms we consider); comparing with area-based algorithms, aggregation offers comparable accuracy with much higher and directly controllable precision. As a guideline for the practical use of aggregation, our experimentation shows that aggregating any 3 or 4 algorithms out of our 12 algorithms will be able to achieve a good performance gain.

I. INTRODUCTION

Recent years have seen intense research investigating using Wireless Local Area Networks (WLANs) as a localization infrastructure. Many received signal strength (RSS) based localization algorithms, ranging from simple matching to complicated machine learning, have been proposed. Elnahrawy et al [1] compared a broad spectrum of such algorithms. They showed that although vastly different, none of the algorithms manage to localize objects accurately all the time due to the fundamental uncertainty from environmental effects. That work showed that all the algorithms’ performance is strikingly similar when examining their error CDFs (Cumulative Distribution Function): they all have similar slopes, medians and long tails.

Our work, however, found that although similar in overall performance, when examining a specific localization attempt, different localization algorithms are affected differently by the fundamental uncertainty. Certain algorithms will localize really well at locations that others do not. To take advantage of such complementary situations, we thus propose using aggregation, i.e., returning the union of the location estimate from each algorithm as the final localization result, to improve the overall performance.

To demonstrate the benefits of using algorithm aggregation, we compared it with both point-based (where the localization result is a single location estimate) and area-based algorithms (the localization results are presented as an area). Because aggregation returns multiple locations, it exhibits similar properties as area-based algorithms and it offers similar advantages in directing user’s search for an object. To be able to compare both sets of algorithms, our evaluation considers both accuracy and precision, where accuracy is the likelihood the object is within the area and precision is inversely related to the size of the returned area or the number of returned locations.

We ran our comparisons using measured data from 2 building sites; one is the computer science building (CoRE) at Rutgers University while the other is an industrial lab. We found that algorithm aggregation significantly improves the accuracy. Specifically, aggregation with just 2 algorithms can improve the accuracy by 23% (industrial lab) and 30% (CoRE building) respectively, while aggregation with 12 algorithms (which are all the algorithms we consider) can increase the accuracy by 51% (industrial lab) and 64% (CoRE building) respectively. Our results also showed that aggregation offers comparable accuracy with area-based algorithms but with much higher precision. Aggregation also provides a more direct control of the accuracy/precision tradeoff since the aggregation size directly maps to precision while the area-based algorithms can only indicate the trend of the area growth instead of the real size.

To give a practical guideline in using aggregation, we also explored the effects of aggregating a different number of algorithms and different combinations of algorithms. Results showed that the performance gain of using algorithm aggregation more or less becomes saturated around 4 algorithms and the variance of performance gain among different algorithms combinations stabilized to be around one foot at aggregation size 3. This suggests that users can aggregate any 3 or 4 algorithms from our 12 algorithms set, which is sufficient to bring out the diversity and achieve a good performance gain.

II. RELATED WORK

Recent years have seen tremendous efforts at building small and medium scale localization systems using WLAN technologies, especially using 802.11 and signal strength [2]–[4]. The underlying principles vary from trilateration, triangulation, scene matching (e.g., fingerprinting), and combinations of these approaches. A full treatment of the myriad of techniques is beyond the scope of this work. Instead of proposing a new localization algorithm, we show that a simple aggregation of the existing algorithms can significantly improve the overall performance.

One set of the existing algorithms [5] proposed to use area rather than a single point to represent localization results. Such presentation offers better description of localization uncertainty and provides tradeoff between localization accuracy and pre-
Our proposed algorithms aggregation offers similar uncertainty description and accuracy/precision tradeoff. We show that aggregation also allows more straightforward control of the precision and it can achieve comparable accuracy with much better precision.

[1], [6], [7] compared a host of localization algorithms ranging from simple matching to complicated neural networks. Their data showed that all algorithms have similar performance, both in accuracy and in their robustness to attacks when an adversary attenuates or amplifies the signal strength from Access Points (AP). Instead of comparing the overall similarity, our method exploits the difference in localization results at each particular point from different algorithms to achieve better accuracy.

### III. ALGORITHM AGGREGATION

To explain our proposed algorithm aggregation, we start with a broad overview of our algorithm menagerie, summarized in Table I. Since our aim is to explore the benefits of algorithm aggregation, details of the algorithms are not of essential importance. We thus refer the details of each algorithm to [1] and focus our discussion on a brief overview of typical RSS based localization.

#### A. Point-based and Area-based Algorithms

Typical RSS based indoor localization systems use the average RSS from the $n$ access points present in the building, $AP_1, AP_2, \ldots, AP_n$, as a fingerprint to differentiate locations. For example, the fingerprint at location $i$ is $(s_{i1}, s_{i2}, \ldots, s_{im})$. Deployment of such fingerprints can be divided into 2 phases. First, in the offline phase, signal fingerprints are empirically measured at $m$ locations. All $m$ fingerprints along with their locations $[(x_i, y_i), (s_{i1}, s_{i2}, \ldots, s_{im})]$ constitute the fingerprints for the sampled building. Second, in the online phase, RSS values collected by the object to be localized can then be used to compare with the floor fingerprints collected offline to estimate the location. Different localization algorithms thus differ in how such comparison and estimation is done in the online phase. They also differ in how they present the localization results: **Point-based algorithms** return the localization result as a single location, for example the most probable or best matching point. **Area-based algorithms** instead return the localization result as an area (represented as a group of grid tiles). In addition, we designed point-based algorithms from the area-based algorithms. For example, they will return the center of the area or the most probable point within the area. In this work we cover 3 area-based algorithms (Simple Point Matching (SPM), Area Based Probability (ABP), and Bayesian Network (Bayes)) and 5 standard point-based algorithms (RADAR, Averaged RADAR, Gridded RADAR, Highest Probability, and Averaged Highest Probability). We also cover 7 point-based algorithms derived from the area-based algorithms (Top of SPM, Center of SPM, Top of ABP, Center of ABP, Bayesian Point, Averaged Bayesian, and Center of Bayes). Aggregation is thus aggregating any combination of algorithms from the set of 12 point-based algorithms.

#### B. Algorithm Aggregation

Algorithm aggregation simply combines a set of point-based algorithms and returns the union of their location estimates as the final result. For example, if $n$ point-based algorithms return their localization results as $l_i, i = 1..n$, respectively, the aggregation result is then the whole location set $\{l_i, i = 1..n\}$. Aggregation thus combines the strength of all the algorithms. At a single location, as long as one of the algorithms can correctly localize, aggregation will be able to locate correctly as well. The performance gain from such aggregation, however, depends on the participating algorithms performing differently and complementing each other at each of the locations. Considering their overall performance similarity, it is not straightforward whether the existing algorithms satisfy such a requirement. In Section IV, we show that aggregation indeed significantly improves localization performance.

<table>
<thead>
<tr>
<th>Type</th>
<th>Algorithm</th>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area-Based</td>
<td>Simple Point Matching</td>
<td>SPM</td>
<td>Matches the RSS to a tile set using thresholds.</td>
</tr>
<tr>
<td></td>
<td>Area Based Probability</td>
<td>ABP-α</td>
<td>Matches the RSS to a tile set probabilistically with confidence bound $\alpha%$.</td>
</tr>
<tr>
<td></td>
<td>Bayesian Network</td>
<td>Bayes</td>
<td>Returns the most likely tiles using a Bayesian network.</td>
</tr>
<tr>
<td>Point-Based</td>
<td>RADAR</td>
<td>R1</td>
<td>Returns the midpoint of the closest 2 training points in signal space.</td>
</tr>
<tr>
<td></td>
<td>Averaged RADAR</td>
<td>R2</td>
<td>Applies RADAR using an interpolated grid.</td>
</tr>
<tr>
<td></td>
<td>Gridded RADAR</td>
<td>GR</td>
<td>Applies likelihood estimation to the received signal.</td>
</tr>
<tr>
<td></td>
<td>Highest Probability</td>
<td>P1</td>
<td>Returns the center of the resulted area from using Area Based Probability.</td>
</tr>
<tr>
<td></td>
<td>Averaged Highest Probability</td>
<td>P2</td>
<td>Returns the midpoint of the top 2 likelihoods.</td>
</tr>
<tr>
<td></td>
<td>Top of Simple Point Matching</td>
<td>S1</td>
<td>Returns the closest matching point within the resulted area from using Simple Point Matching.</td>
</tr>
<tr>
<td></td>
<td>Center of Simple Point Matching</td>
<td>SC</td>
<td>Returns the center of the resulted area from using Simple Point Matching.</td>
</tr>
<tr>
<td></td>
<td>Top of Area Based Probability</td>
<td>AT</td>
<td>Returns the most likely point within the resulted area from using Area Based Probability (results will be the same as Gridded Probability (GP)).</td>
</tr>
<tr>
<td></td>
<td>Center of Area Based Probability</td>
<td>AC-α</td>
<td>Returns the center of the resulted area from using Area Based Probability with confidence $\alpha$.</td>
</tr>
<tr>
<td></td>
<td>Bayesian Point</td>
<td>B1</td>
<td>Returns the most likely point using a Bayesian network.</td>
</tr>
<tr>
<td></td>
<td>Averaged Bayesian</td>
<td>B2</td>
<td>Returns the mid-point of the top 2 most likely points.</td>
</tr>
<tr>
<td></td>
<td>Center of Bayesian Network</td>
<td>BC</td>
<td>Returns the center of the resulted area from BN.</td>
</tr>
</tbody>
</table>

**TABLE I. All algorithms and variants.**
Grey spaces are corridors, white spaces are offices or laboratories. Squares are the APs locations. Large dots and small dots show an example random training set and testing set respectively.

**Fig. 1.** Floor plan and set up

**Fig. 2.** Distance accuracy for point-based algorithms and algorithm aggregation. Legends indicate aggregation size, the number of algorithms used for aggregation.

**TABLE II.** Aggregation performance gain. Each row corresponds to an aggregation size; Each column corresponds to a training size.

<table>
<thead>
<tr>
<th>Training Size</th>
<th>20</th>
<th>35</th>
<th>55</th>
<th>85</th>
<th>115</th>
<th>145</th>
<th>185</th>
<th>215</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>25.6%</td>
<td>26.2%</td>
<td>26.8%</td>
<td>26.1%</td>
<td>27.8%</td>
<td>28.0%</td>
<td>28.5%</td>
<td>23.3%</td>
</tr>
<tr>
<td>12</td>
<td>63.4%</td>
<td>57.1%</td>
<td>57.9%</td>
<td>59.2%</td>
<td>60.7%</td>
<td>57.6%</td>
<td>59.9%</td>
<td>51.0%</td>
</tr>
</tbody>
</table>

C. Limitations of Aggregation

Our simple algorithm aggregation approach is subject to the following limitations: First, algorithm aggregation presents the result as a set of possible locations but itself does not identify the best estimate. The filtering requires either context information or additional human efforts. Such additional uncertainty is the trade-off for the gained accuracy. For example, a user searching for an object will be responsible to try out the given estimates. However, should users be willing to make the tradeoff, our aggregation will be able to offer better estimates.

Second, while combining the strength of all the algorithms, aggregation combines their weaknesses as well. This means that among the estimates given, some may be bad estimates. This again is a tradeoff. In this paper, we are mainly interested in how much closer can aggregation push the estimates towards the true location. In Section IV, we thus evaluate based on the best estimate (minimum error).

### IV. Evaluation

A. Experiments Setup

In order to show our results are not an artifact of a specific floor, we used measured RSS data from 2 sites: The first site is the Computer Science department CoRE building (CoRE) at Rutgers University, while the second site is an office building at an industrial laboratory (Industrial). Figures 1(a) and (b) show the layout of these 2 floors, respectively.

We collected fingerprints at 286 locations on the 3rd floor of the CoRE building over a period of 2 days, which contains just over 50 rooms in a 200x80ft (16000 ft\(^2\)) area. A total of 252 fingerprint vectors were collected from the industrial
site, where the floor includes about 115 rooms in a 225x144ft (32400 ft²) area and has many corridors in-between these rooms.

B. Metrics

We use 2 metrics to evaluate algorithm aggregation, which allow us to compare it with both point-based and area-based algorithms properly. **Distance Accuracy** is the distance between the true location and the estimated location or the tiles in the estimated area. As discussed earlier, here the evaluated distance accuracy refers to the minimum error for both aggregation and area-based algorithms. **Precision** is inversely related to the size of the returned area (the ft²) or the number of returned locations. Notice that the larger the returned area, the lower the precision.

C. Aggregation vs. Point Based Algorithms

Since point-based algorithms only return a single location while algorithm aggregation returns \( n \) locations where \( n \) represents aggregation size, the number of algorithms used for aggregation, point-based algorithms apparently offer higher precision. In Section IV-E we show that this \( n \) can go as low as 3 or 4. This choice between point-based algorithms and aggregation is a tradeoff for each particular user to decide. In this section we thus focus our comparison on accuracy.

Figure 2 shows the distance accuracy for point-based algorithms (aggregation size 1) and algorithm aggregation. The accuracy is averaged across testing points. For point-based algorithms, the accuracy is further averaged across all the algorithms. For aggregation, accuracy is further averaged across different algorithms combinations for the same aggregation size. We see that aggregation significantly improves localization performance using even just 2 algorithms. The performance gain gradually levels out as we increase the aggregation size. For both buildings, the final accuracy for aggregating 12 algorithms comes to around 5 feet. Table II lists the performance gain in percentage. With 2 algorithm aggregation, CoRE building localization improves by over 30% while the industrial lab improves by over 23%. If we push the aggregation to include all the point-based algorithms (12 algorithm aggregation), the CoRE building results can improve by over 64% while the industrial lab results can improve by over 51%.

D. Aggregation vs. Area Based Algorithms

Figure 3 shows the distance accuracy for the area-based algorithms while Figure 4 shows the precision. Accuracy for
aggregation was shown previously in Figure 2. Precision for aggregation is not shown since it just corresponds to the aggregation size. In comparison, area-based algorithms offer slightly better accuracy. They can render an average error of around 2 to 3 feet while aggregation with 12 algorithms can reach close to 5 feet. However, this slight accuracy win is largely due to its low precision. In CoRE, the uncertainty area ranges from 150 \( ft^2 \) to 700 \( ft^2 \). In the industrial building, it may even go up to a few thousand \( ft^2 \). In comparison, algorithm aggregation returns 12 locations at most. Aggregation thus has achieved a comparable accuracy while drastically improved the precision. Essentially, rather than returns a whole large area, it picks a few most probable locations.

Another advantage of aggregation over area-based algorithms is its straightforward mapping to precision. By simply adjusting aggregation size, users can directly specify how many possible locations they are willing to accept to achieve better accuracy. Area-based algorithms, on the other hand, cannot offer such direct control over area sizes.

E. Choosing aggregation size and algorithms combination

Aggregating \( n \) algorithms will return \( n \) locations and require each of the algorithms to be executed. The results uncertainty and its overhead are thus proportional to the aggregation size. Naturally we would like to aggregate fewer algorithms. In order to give a practical guideline for using aggregation, we thus explore whether we can shrink the aggregation size, and with a given size, how to choose the specific algorithms.

From Figure 2, we see that the performance gain of using aggregation gradually saturates or flattens as we scale up the aggregation size. Starting from aggregation size 4, the performance gain is minimal. Thus, unless the requirement for accuracy is very high, we should choose an aggregation size of no more than 4. We further plot the variance of accuracy for aggregating different algorithm combinations in Figure 5. It shows that as the aggregation size increases, the fluctuation in performance among the different combinations decreases. It seems to reach reasonable stableness around one foot at size 3.

This implies that with sufficient aggregation size, the choice of algorithm combination does not matter much. In fact, a more detailed look indicates that for a given aggregation size there is no single best combination for both sites or for all the training sizes. In summary, aggregating any 3 or 4 algorithms out of our 12 algorithms can achieve a good performance gain.

V. CONCLUSION

In this paper we proposed to use algorithm aggregation, i.e., returning the union of the location estimate from each algorithm as the final localization result, to improve localization performance. Our experimentation results showed that aggregation significantly improves localization accuracy over point-based algorithms and it offers comparable accuracy at much higher and directly controllable precision over area-based algorithms. Finally, to give a guideline for using aggregation, we showed that aggregation with any 3 or 4 algorithms can achieve a good performance gain.

REFERENCES


