Robust Indoor Location Estimation of Stationary and Mobile Users

Youngjune Gwon, Ravi Jain, and Toshiro Kawahara
DoCoMo Communications Laboratories USA, Inc.
181 Metro Drive, Suite 300
San Jose, CA 95110, USA
{gyj, jain, kawahara}@docomolabs-usa.com

***Abstract*—We present algorithms for estimating the location of stationary and mobile users based on heterogeneous indoor RF technologies. We propose two location algorithms, Selective Fusion Location Estimation (SELFLOC) and Region of Confidence (RoC), which can be used in conjunction with classical location algorithms such as triangulation, or with third-party commercial location estimation systems. The SELFLOC algorithm infers the user location by selectively fusing location information from multiple wireless technologies and/or multiple classical location algorithms in a theoretically optimal manner. The RoC algorithm attempts to overcome the problem of aliasing in the signal domain, where different physical locations have similar RF characteristics, which is particularly acute when users are mobile. We have empirically validated the proposed algorithms using wireless LAN and Bluetooth technology. Our experimental results show that applying SELFLOC for stationary users when using multiple wireless technologies and multiple classical location algorithms can improve location accuracy significantly, with mean distance errors as low as 1.6 m. For mobile users we find that using RoC can allow us to obtain mean errors as low as 3.7 m. Both algorithms can be used in conjunction with a commercial location estimation system and improve its accuracy further.**

Keywords-location estimation;static scene analysis; sensor fusion; region of confidence; wireless LAN; Bluetooth

I. INTRODUCTION

The recent growth of interest in pervasive computing and location-aware systems and services provides a strong motivation to develop techniques for estimating the location of devices – and hence users – in both outdoor and indoor environments. Indoor location estimation is particularly challenging due to the poor indoor coverage of Global Positioning System (GPS). One approach to indoor location estimation is to deploy a dedicated sensor network (e.g. using infra-red or ultra-sound technology), but this has obvious cost and maintenance ramifications. An alternative approach (such as proposed in [1] [2] [3] [4] [5] [6]) is to use existing wireless LAN infrastructures. This latter approach uses techniques that employ radio signal information obtained from wireless beacons to infer location estimates. Static scene analysis collectively refers to the basis of these techniques and typically consists of an offline stage where calibration of the indoor wireless environment is carried out and a run-time stage where the calibration is used for location estimation.

In parallel with the increased interest in location estimation, we observe that future mobile wireless systems are expected to consist of heterogeneous wireless access technologies. In fact, user terminals that combine cellular as well as wireless LAN technology, or multiple wireless LAN technologies, are available or expected soon. Our primary focus in this paper is to develop indoor location estimation algorithms that benefit from the heterogeneity of wireless access technologies.

We introduce two location algorithms, namely Selective Fusion Location Estimation (SELFLOC) and Region of Confidence (RoC), which can perform estimation and tracking of the location of stationary and mobile users. While the focus of this paper and our current implementation of the SELFLOC and RoC algorithms are for indoor location estimation, in principle they are applicable to appropriate outdoor scenarios also.

The SELFLOC algorithm combines multiple branches of information sources (such as heterogeneous RF sensors) and selectively weights them such that error contribution from each branch can be minimized. SELFLOC input branches can be used not only for combining multiple information sources but also for combining multiple classical location algorithms.

The RoC algorithm attempts to overcome the problem of *aliasing*, where two physically different locations possess similar RF characteristics that lead to misinterpretation of RF measurements. Conventional RF-based location algorithms typically facilitate filtering in the signal domain; RoC uses analysis in the space domain to filter outliers. Since aliasing is an especially acute problem where mobility is taken under consideration, we focus on the use of RoC algorithm for the mobile user case.

We have designed and implemented our algorithms in a heterogeneous wireless environment that consists of both IEEE 802.11b wireless LAN and Bluetooth access points. We have carried out extensive empirical measurements and performance evaluation under many different circumstances. All of the data and results presented in this paper are experimentally obtained from actual measurement with commercially available IEEE 802.11b and Bluetooth access points and client cards.

The main objective of this paper is not to develop new radio signal processing algorithms (e.g. time-delay-of-arrival) for location estimation or new geometric algorithms (e.g. classical algorithms such as triangulation). It is rather to fuse or aggregate the location estimates used from these underlying algorithms intelligently to improve accuracy. As new algorithms are developed, or as new wireless technologies are deployed, our algorithms can be applied and used to improve accuracy further.

Early work in this area included the RADAR [1] system that showed that accurate indoor location estimation could be achieved without deploying separate sensor network infrastructures. Their idea is to infer location of an IEEE 802.11b wireless LAN user by leveraging received signal strength information available from multiple wireless LAN beacons. In following work [12], RADAR was enhanced by a Viterbi-like algorithm that specifically addresses the issues such as continuous tracking and signal aliasing. The Nibble system [2] took a probabilistic approach in a similar wireless LAN environment.

In [3], [6] and [13], the MultiLoc system, which utilizes information from multiple wireless (or wired) technologies, was proposed. The MultiLoc system employs two simple 'sensor fusion' techniques to illustrate the benefit of combining heterogeneous information sources in location estimation. (The SELFLOC algorithm used in this paper is more sophisticated, and the RoC algorithm also considers the mobile user case).

In [4], the authors demonstrated that RF information available from local wireless access points is sufficient to allow a mobile device to reliably track its location by using probabilistic algorithms widely used in the field of robotics. In [5], an enhanced triangulation method named 'Triangulation Mapping Interpolation (TMI)' was explained and evaluated in comparison to other methods.

In general, it is not yet clear that any single location estimation algorithm is best suited for all environments or technologies. We thus consider the combination of multiple algorithms and show that combining them intelligently can indeed provide significant benefits.

Our work differs from the previous work in that (1) we describe how radio signal information from multiple technologies can be fused using a known optimal algorithm to improve accuracy; (2) we show how location estimates using multiple estimation algorithms can be combined, again using a known optimal algorithm, to improve accuracy; (3) we employ information filtering and processing in both the signal and space domains to better cope with aliasing of RF measurements; and (4) we consider tracking of the location of mobile users.

The rest of the paper is organized as follows. In Section II, we briefly present background on RF-based location estimation. In Section III, we explain SELFLOC, an algorithm for stationary location estimation. RoC, an algorithm for mobile location estimation, is explained in Section IV. Section V describes our experimental testbed, measurement, data collection, and software implementation. Section VI presents the performance evaluation of our location algorithms. We compare and analyze the accuracies of the SELFLOC and RoC algorithms using the mean distance error metric. Section VII concludes the paper.

II. Location Estimation Using RF Technologies

High precision indoor location systems typically require a separate sensor network infrastructure that is used for positioning purposes only. Such systems utilize ultra-sound or infrared sensors that are densely placed in an area. Given the reduction in installation and maintenance cost and effort, RF-based location estimation using local wireless infrastructures is becoming increasingly attractive especially as they are emerging as popular wireless access networks. We have adopted an RF-based static scene analysis experimental framework for indoor location estimation.

A. *Static Scene Analysis*

Static scene analysis involves the examination of certain features in an environment containing the location system. Static scene analysis consists of 'offline' and 'run-time' stages.

- Offline stage: during this stage, measurement of the RF features at known locations is carried out. Wireless access points deployed in the environment periodically transmit beacons. A signal metric measured from detected beacons, such as received signal strength, can be a useful RF feature. Signal metrics with respect to each access point are collected. The collected signal metrics are stored in a location database to relate the signal information and coordinates of the known locations.

- Run-time stage: measurement of the same signal metric as used in the offline stage is carried out. The location database is accessed to compare the signal metrics collected during the run-time (at an unknown location) with the stored entries. A location estimation algorithm is then applied to infer the location estimate for the unknown location.

In the remainder of this paper, we assume that the measurable RF feature is signal strength (or a proxy for signal strength). Received Signal Strength Indicator (RSSI) measured from wireless beacons is our choice of the signal metric although our algorithms are applicable to other metrics also.

B. *Classical Location Estimation Algorithms*

We first describe three known, classical, location estimation algorithms.

- Triangulation (TN): the algorithm forms circles centered at the RF access points, where the radius of each circle depends on the measured signal strength. The radius is approximated by comparing the run-time measurement with the information stored in the location database. A system of equations representing the circles is solved; typically, the vertices of a number of common areas shared by the circles are found using the solutions from the system of equations. Averaging the coordinates of the solutions that form the smallest area gives the final location estimate.

- K-nearest neighbor averaging (KNN): the algorithm searches for K-location entries from the location database having the smallest root mean square error in signal space with the given run-time measurement at the unknown location. Averaging the coordinates of the K-locations gives the final location estimate.

- Smallest M-vertex polygon (SMP): M candidate locations from *each* access point whose distance in the signal space with the given run-time measurement are searched from the location database. M-vertex polygons are formed by including at least one candidate location from each access point. The smallest polygon is the one having the shortest perimeter. Averaging the coordinates of vertices of the smallest polygon gives the final location estimate.

TN has been widely used by various known location systems including Global Positioning System (GPS). KNN has been used in RADAR [1] [12]. SMP has been used in MultiLoc [3] [6].

C. Sensor Fusion

Sensor fusion is the process of combining multiple and independent observations to obtain improved accuracy and robustness. Today's mobile devices typically incorporate two or more different wireless technologies (e.g., IEEE 802.11a, 11b, 11g, and Bluetooth). This enables sensor fusion location estimation by simultaneous use of information available from heterogeneous RF sensors. In the following sections, we will describe algorithms to perform the sensor fusion intelligently.

III. STATIONARY LOCATION ESTIMATION USING SELFLOC

The Selective Fusion Location Estimation (SELFLOC) algorithm infers a location estimate by combining (or fusing) multiple information sources. These information sources can be multiple location estimates from different algorithms.

Fig. 1 shows an overview of the SELFLOC algorithm to fuse three information sources (input branches). Each input branch, conveying its own information, is individually weighted. The weighted sum gives the SELFLOC estimate. The branch weights are calibrated during the offline stage using error feedback, as described below. The main benefit of the SELFLOC algorithm is the accuracy gain by combining *uncorrelated* information contributed from multiple branch inputs of heterogeneous sources and algorithms.

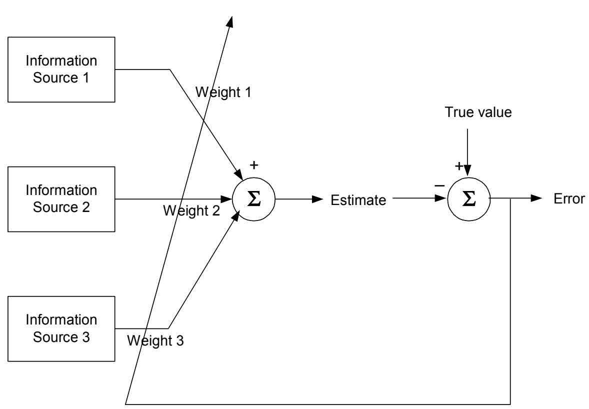

Figure 1. Overview of the SELFLOC algorithm

A. Minimum Mean Square Error (MMSE) Algorithm for SELFLOC Weight Calibration

We have adopted the MMSE algorithm [10] for SELFLOC weight training and calibration. This is performed during the offline stage. We denote the true location of the user, $d = (d_x, d_y)$; this is the *desired* output that SELFLOC attempts to estimate. Suppose we are interested in fusing three location estimates available independently and x_1, x_2, and x_3 represent x-coordinates from the estimates (we consider three inputs for ease of exposition; the algorithm generalizes to more inputs in a straightforward manner). Also, the y-coordinate estimate follows similarly. Let $\underline{X}$, a column vector, contain the SELFLOC input branches having values x_1, x_2, and x_3 such that $\underline{X} = [x_1 \quad x_2 \quad x_3]^T$. Then, the SELFLOC estimate, $\hat{x}$, is written as:

$$\hat{x} = w_1 \cdot x_1 + w_2 \cdot x_2 + w_3 \cdot x_3, \tag{1}$$

where w_1, w_2, and w_3 are branch weights.

Eq. (1) is rewritten as $\hat{x} = W_x^T \cdot \overline{X}$ where W_x is a column vector containing the branch weights w_1, w_2, and w_3. The SELFLOC estimation error is given by:

$$e_x = d_x - \hat{x} = d_x - W_x^T \cdot \overline{X} \tag{2}$$

The Mean Square Error (MSE) is:

$$Ee_x^2 = Ed^2 + W_x^T E[\underline{X} \cdot \underline{X}^T] W_x - 2E[d \cdot \underline{X}^T] W_x . \tag{3}$$

Denoting $R_{XX} = E[\underline{X} \cdot \underline{X}^T]$ and $R_{DX} = E[d \cdot \underline{X}^T]$,

$$MSE = Ed^2 + W_x^T R_{XX} W_x - 2 R_{DX} W_x . \tag{4}$$

We are interested in minimizing the MSE and we take the derivative of (4) with respect to W. Solving for W, yields the optimal MMSE weight vector, W_x^*:

$$W_x^* = R_{XX}^{-1} R_{DX} . \tag{5}$$

R_{XX}^{-1} is the inverse of correlation matrix of $\underline{X}$ and R_{DX} is cross correlation vector between $\underline{X}$ and d_x.

B. Applying SELFLOC to Classical Location Algorithms

The SELFLOC algorithm can be applied to a classical location algorithm to improve its accuracy. We have explained three location algorithms in Section II, triangulation (TN), K-nearest neighbor averaging (KNN), and smallest M-vertex polygon (SMP). We explain the SELFLOC enhancement of these algorithms.

Triangulation infers the location estimate by averaging the coordinates of the vertices of the smallest-area triangle formed by solving the system of equations representing circles. A simple modification to triangulation is to apply some other weighting criterion other than equal weighting (i.e., averaging) on the vertices of the smallest triangle. The input branches of SELFLOC-enhanced triangulation (eTN) consist of coordinates of the vertices. In eTN, x and y coordinates of the vertices are weighted and summed (separately) to yield the final estimate.

The SELFLOC-enhanced K-nearest neighbor (eKNN) and SELFLOC-enhanced smallest M-vertex polygon (eSMP) algorithms are structured in a similar fashion. In eKNN, location coordinates of the K-closest matched samples become the SELFLOC input branches. In eSMP, the vertices of the smallest perimeter polygon become the SELFLOC input branches. The eTN, eKNN, and eSMP algorithms require offline SELFLOC weight calibration before the run-time use. We use MMSE, as described earlier, to calibrate these weights.

C. Aggregation of Multiple Classical Location Algorithms Using SELFLOC

The SELFLOC algorithm enables the aggregation (or fusion) of different classical location algorithms. For example, we can attempt to aggregate the TN, KNN, and SMP algorithms using SELFLOC. The estimates from the three algorithms become the SELFLOC input branches. Again, we require two separate SELFLOC components for x and y coordinates. The final location estimate is written as:

$$\hat{x} = w_{TN\_x} \cdot x_{TN} + w_{KNN\_x} \cdot x_{KNN} + w_{SMP\_x} \cdot x_{SMP}, \tag{6}$$

$$\hat{y} = w_{TN\_y} \cdot y_{TN} + w_{KNN\_y} \cdot y_{KNN} + w_{SMP\_y} \cdot y_{SMP}. \tag{7}$$

x_{TN}, x_{KNN}, and x_{SMP} designate x-coordinate estimates from the TN, KNN, and SMP algorithms, respectively. The input branches are weighted by w_{TN_x}, w_{KNN_x}, and w_{SMP_x}, and summed to yield the final estimate, $\hat{x}$, as in (6). The same naming convention applies to y-coordinate and its final estimate, $\hat{y}$, is given in (7). The SELFLOC weights for x and y components (in column vector form), $W_x = [w_{TN\_x} \quad w_{KNN\_x} \quad w_{SMP\_x}]^T$ and $W_y = [w_{TN\_y} \quad w_{KNN\_y} \quad w_{SMP\_y}]^T$, must be calibrated during the offline stage.

D. SELFLOC Weight Localization and Iterative Location Refinement

Calibrated SELFLOC weights reflect adaptation to a specific area, where the samples used for calibration should be obtained from various locations contained in the area. Having a single set of the SELFLOC weights over a large area may have some adverse effects on the accuracy of a SELFLOC system. The accuracy of the SELFLOC estimation may degrade because some regions deviate heavily from the generalized characteristics of the entire area.

The SELFLOC weight localization approach aims to achieve more accuracy by dividing the SELFLOC system into regions and applying a different set of weights to each region. Fig. 2 depicts this approach, named SELFLOC weight localization with iterative location refinement. It consists of two stages: location approximation and refinement.

In the first stage, simple location approximation to identify the region is performed (the base TN, KNN, or SMP algorithms suffice for this task). By using the unique set of weights designated for the identified region, a refined estimate, $\hat{L}_F$, is obtained. In the second stage, $\hat{L}_F$ is fed back and used to re-identify the region. The refined estimate is different from the previous one if the region differs. If the current and the previous refined estimates match, the final estimate is found. If they differ, the location refinement stage will be repeated iteratively until the estimates match.

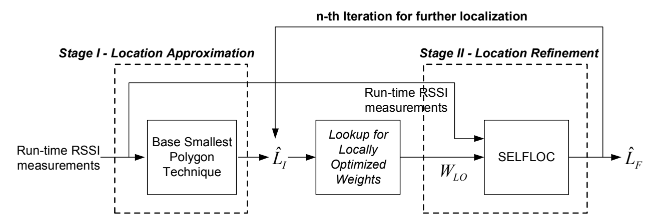

Figure 2. SELFLOC weight localization with iterative location refinement

IV. MOBILE LOCATION ESTIMATION USING ROC

Classical location algorithms in static scene analysis such as triangulation, K-nearest neighbor, and smallest polygon essentially perform a non-linear transformation based on mapping between the signal and space domains (i.e., the location algorithm's input, signal metrics, are collected, mapped, and computed to give the output which is a location estimate represented in space). The main drawback of such algorithms is the possibility of physically different locations being represented similarly in the signal domain, a phenomenon known as *aliasing*. The Region of Confidence algorithm (RoC) attempts to counter aliasing in the signal domain. The algorithm first forms a region of confidence (RoC) within which the true location of a user lies with some high probability. Then, a series of estimates obtained in close time intervals (from either the same or different classical location algorithms) are filtered using the formed RoC. Estimates bound within the RoC are accepted for further processing. The complete RoC algorithm is as follows.

A. RoC Formation Using Triangulation

We consider a method of forming a RoC by examining geometric properties of the triangulation algorithm. For ease of exposition, we consider triangulation using three access points. The shape of the RoC is either a circle or fraction of a circle. The center of the RoC is the final estimate from triangulation. The radius of the RoC should be set such that the RoC includes the true location of the mobile user with some high probability.

Triangulation comprises the following steps: (1) a system of equations representing circles is solved to obtain a set of points, which we call *triangulation points*; (2) each triangulation point is taken to be the vertex of a triangle; (3) the areas of all possible triangles formed using these vertices are computed and compared; and (4) the centroid of the triangle with the smallest area is taken as the location estimate (i.e., statistical average of coordinates of the vertices). We take the radius of the RoC to be the distance between the centroid of the smallest-area triangle and the furthest triangulation point from the centroid. We have empirically found out that using this value for the RoC radius includes the true location over 90 % of the time for our data.

1) Common Patterns of Circles in Triangulation

Triangulation forms various patterns of circles. These patterns are classified based on the number of real and imaginary solutions for the system of equations of circles in the triangulation algorithm. Fig. 3 depicts three different types of common patterns of circles in triangulation. For Type 1, there are 6 distinct real solutions (i.e., Points *a, b, c, d, e,* and *f*). Triangle *(b,d,e)* yields the smallest area. Thus, the final location estimate for Type 1 using triangulation is the centroid of triangle *(b,d,e)*. For Type 2, the smallest area triangle happens to be formed by using Points *h, i,* and *j*. For Type 3, there is no triangle, since there are only two triangulation points. The final estimate is obtained by computing the midpoint of the line joining Points *k* and *l*.

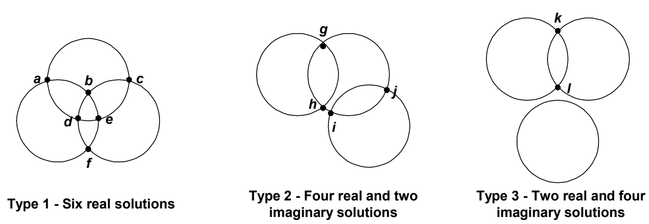

Figure 3. Common three-circle patterns of triangulation

2) Rare Patterns of Circles in Triangulation

Fig. 4 illustrates rare patterns of circles in triangulation. Type 4a has no real solutions. The centers of the circles are taken as the vertices of a triangle and the centroid of the triangle becomes the final location estimate. Type 4b represents the most desirable situation (yet most unlikely) where there is a triple root (Point *n*), or the final estimate. The average of Points *q* and *r* gives the final estimate for Type 4c. Point *s* is the final estimate for Type 4d and the average of Points *t, u*, and *v* gives the final estimate for Type 4e.

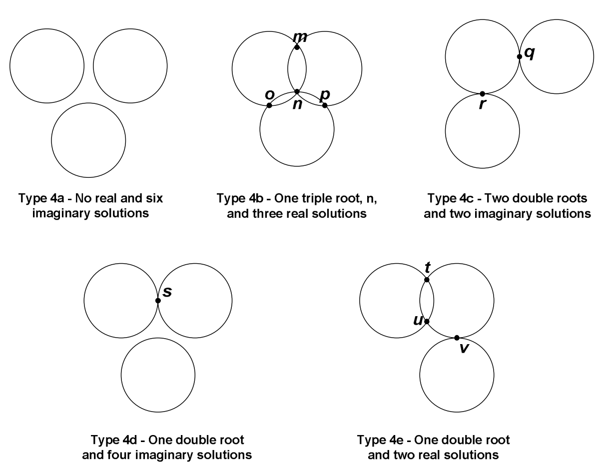

Figure 4. Rare three-circle patterns of triangulation

3) Formation of RoC for Each Type

Fig. 5 presents the RoC for Type 1. 'X' designates the final location estimate of triangulation. Since the distance between 'X' to Point *b* is the longest, it becomes the radius of the RoC (hence, Points *d* and *e* reside inside the RoC).

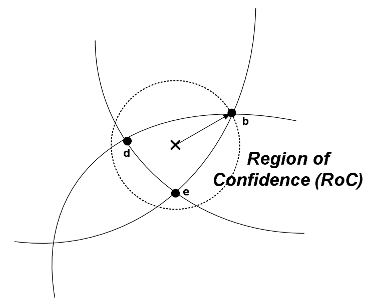

Figure 5. RoC for Type 1

Fig. 6 presents the RoC for Type 2. The final estimate from triangulation (designated as 'X' in the figure) is computed by averaging coordinates of Points *h, i*, and *j*. Since the distance between 'X' and Point *j* is the longest, it becomes the RoC radius which includes Points *h* and *i*.

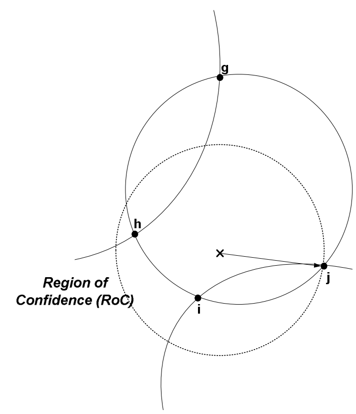

Figure 6. RoC for Type 2

Fig. 7 presents the RoC for Type 3. In this case, the RoC is half circle (or in an arc form) since there is no triangulation point relating the bottom circle. The RoC is biased to include the influence of the bottom circle.

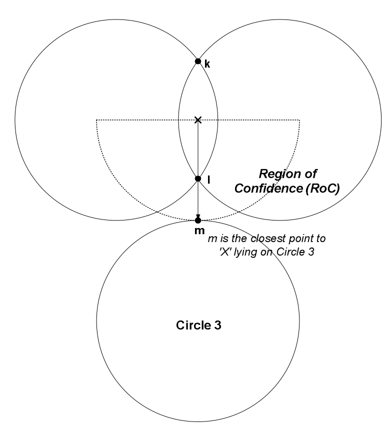

Figure 7. RoC for Type 3

Similar heuristics apply to form RoCs for rare patterns (Types 4a, 4b, 4c, 4d, and 4e). Fig. 8 presents a brief description for each of the rare types.

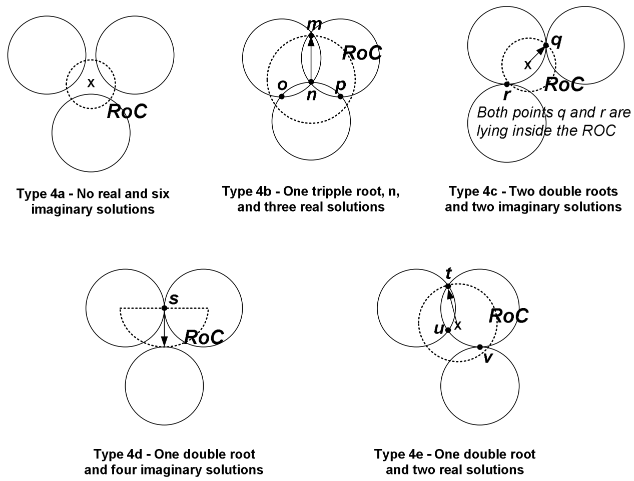

Figure 8. RoC for rare patterns (Types 4a, 4b, 4c, 4d, and 4e)

B. *Region of Confidence Filtering*

After a RoC is formed, filtering estimates based on the given RoC occurs. The RoC filtering is a straightforward processing in the space domain. The location estimates whose coordinates fall within the RoC are accepted and considered for the next step.

C. *Time Segmentation and Shortest Path Heuristic*

The RoC-filtered estimates are organized into time-indexed bins that contain location estimates obtained during time intervals of the same length. This 'time segmentation' processing relates the location estimates with their acquisition time. Then, the shortest path that connects at least one estimate from each bin is found. The final location of the RoC algorithm is obtained by averaging the estimates that constitute the shortest path.

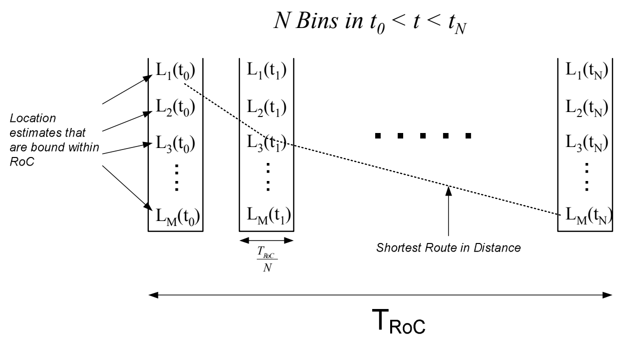

Figure 9. Time segmentation and shortest path

Fig. 9 depicts time segmentation and shortest path. Each RoC is assumed to be valid only for a time period T_{RoC} (a configurable parameter). When T_{RoC} expires, a new RoC must be generated. There are N time-indexed bins, each corresponding to a time interval of T_{RoC}/N. The distance between the location estimates from each pair of consecutive bins is found. Thus, if location estimates $L_1(t_0)$ and $L_2(t_1)$ are (x_1, y_1) and (x_2, y_2), respectively, then the Euclidean distance between them is denoted $d_{12} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$. For each pair of consecutive bins, the locations corresponding to the shortest distance are found; these are called *shortest-path* locations. The final location estimate is simply the average of these shortest-path locations.

D. *Regeneration of RoC*

The success of the mobile location estimation algorithm depends on whether the RoC contains the mobile user's true location or not. In order to fulfill this requirement, the RoC must be regenerated based on observing the user movement.

We denote T_{RoC} as the RoC timer. Whenever T_{RoC} expires, a new RoC must be formed. This is based on the following three methods:

- *The total distance of the shortest path exceeds radius of the RoC*: it is possible for a mobile user to move out of the current RoC when the total distance of the shortest path exceeds the RoC radius. The feedback loop in Fig. 10 is used to convey the total distance of the shortest path.
- *T_{RoC} is a configurable parameter and the RoC is regenerated periodically whenever T_{RoC} expires*: the system explicitly configures the value of T_{RoC}. Although there are many ways to determine the appropriate value for T_{RoC}, we can approximate T_{RoC} using the velocity of a user: $T_{RoC} = \frac{v}{r_{RoC}}$, where v is estimated velocity of the user and r_{RoC} is the radius of the previous RoC.
- *RoC generation based on prefetched run-time inputs*: this is accomplished by inserting a small time delay for candidate process as shown in Fig. 10. The impact of this time delay is to have the look-ahead measurement values for the RoC generation. The delayed run-time inputs (measurement values) are used for candidate process. This will smoothly form a new RoC before reaching the T_{RoC} timeout.

E. *Complete Mobile Location Estimation Algorithm*

Fig. 10 depicts the complete mobile location estimation algorithm, which consists of seven logical steps:

1. *Preprocessing of run-time input*: received signal strength samples (or acquired signal metrics) are band-pass filtered (i.e., with high and low thresholds) and averaged.[†]
2. *Region of confidence (RoC) formation*: the RoC is formed with the preprocessed run-time input.
3. *Candidate process*: a set of location estimates from the same or different location algorithms (e.g. TN, KNN, and SMP) are obtained and time-stamped.

[†] We set variable high and low thresholds such that they cut off each of the highest and lowest 15 % of the sample values acquired during the run-time.

4. *RoC filtering*: the candidate estimates are filtered through the RoC in the space domain. The coordinates of estimates that fall within the RoC are passed to the next step.

5. *Shortest path*: the RoC-filtered estimates are sorted in time-indexed bins and the shortest path connecting at least one estimate from each bin is found.

6. *Final processing*: the estimates that constitute the shortest path are averaged to yield the final estimate.

7. *Regeneration of RoC*: a new RoC is formed when T_{RoC} expires. The feedback loop in Fig. 10 (from shortest path to RoC formation) conveys the information to generate a new RoC.

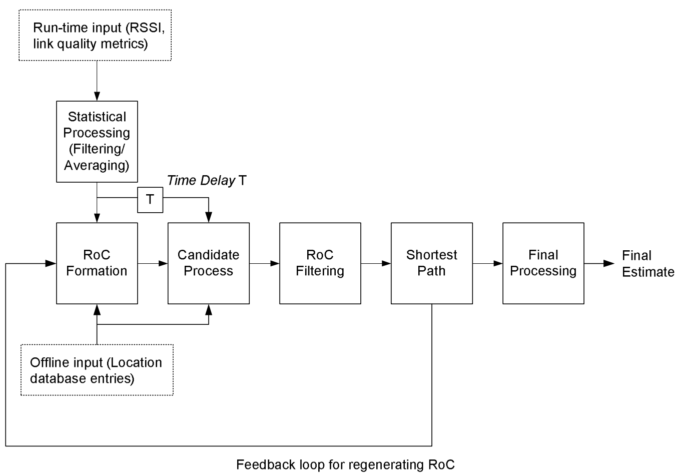

Figure 10. Mobile location estimation algorithm

V. EXPERIMENTS

A. Description of Testbed

Our experimental testbed is located on the third floor of a seven-story building. This is a typical office environment that includes cubicles, small offices, and conference rooms. The floor map and our *x*-*y* coordinate system are illustrated in Fig. 11. The unit grid for calibration measurements has a dimension of 1.422 m by 1.422 m (14/3 ft. by 14/3 ft.) and the testbed spans 39.83 m by 25.60 m, for a total area of 1020 m^2 (10975 sq. ft.).

B. Empirical Measurement and Data Collection

Our mobile node runs Redhat Linux 2.4.18 OS with both a LinkSys Instant Wireless™ client adaptor for IEEE 802.11b wireless LAN and a 3Com Bluetooth™ PC card. Signal strength measurement could not be performed at all grid locations due to the presence of desks, walls, etc. We ran three independent measurement sessions obtaining three independent sets of measurement data for both wireless LAN and Bluetooth, as detailed below:

- Wireless LAN: We placed four LinkSys WAP11 access points. They are designated as ★, labeled W_1, W_2, W_3, and W_4 in Fig. 3. Throughout the entire area, defined by $x \in (0, 28)$ and $y \in (0, 18)$ in our coordinate system, we were able to capture beacons from all four wireless LAN access points, and we collected the RSSI samples in 207 different grid locations. At each location, at least 40 samples were collected for each different direction (heading north, south, west, and east) yielding 160 samples total per each access point.

- Bluetooth: We placed three Axis 9010 Bluetooth access points. They are designated as ▲, labeled B_1, B_2, and B_3 in Fig. 3. Due to its short radio range, only the area defined by $x \in (0, 15)$ and $y \in (0, 11)$ was considered for Bluetooth measurements; this contained 71 different grid locations. We collected samples by reading the link quality metric in Bluetooth beacons. At each location, we obtained 25 samples for each direction, to collect 100 samples total.

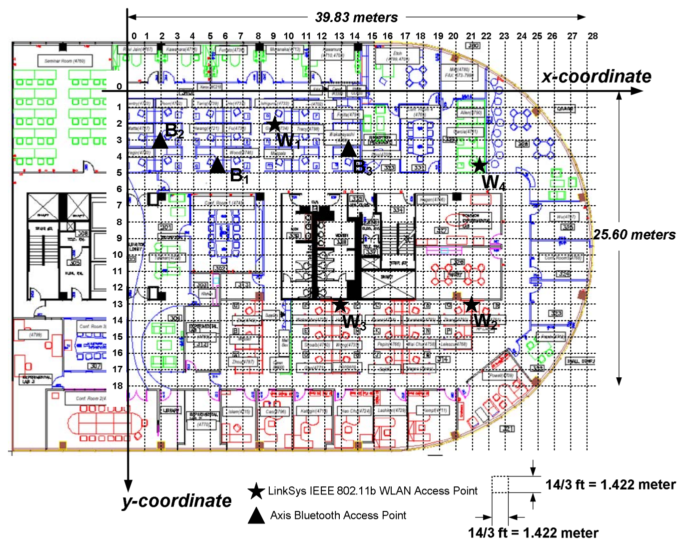

Figure 11. Map and coordinate system of experimental testbed

C. Software Implementation

The wireless LAN air-interface module was written in C based on a device driver developed by the Linux-WLAN Project [8]. The Bluetooth air-interface module was also written in C using the BlueZ Linux-Bluetooth device driver [9]. We built a location information database using samples collected during one of our three measurement sessions. The location information database allows access to the actual *x* and *y* coordinates of measurement location, mean and standard deviation of signal strength (in RSSI or link quality metric) with respect to each access point, and location of access points. We implemented the base location algorithms and SELFLOC in MATLAB.

VI. PERFORMANCE EVALUATION AND DISCUSSION

We use the mean distance error metric to evaluate the accuracies of our location algorithms. Mean distance error represents average *Euclidean* distance between the estimate $(\hat{x}, \hat{y})$ and the true location (x, y), i.e., $d = \sqrt{(x-\hat{x})^2 + (y-\hat{y})^2}$.

A. Stationary Location Estimation

We have conducted six different sets of experiments for stationary location estimation. As mentioned earlier, we have four wireless LAN and three Bluetooth access points. Depending on what combination of access points are utilized, we have three different location systems that use only single technology sensors, namely 3W, 3B, and 4W systems referring to the use of 3 wireless LAN, 3 Bluetooth, and 4 wireless LAN access points, respectively. We also have three different location systems that simultaneously use multiple technology sensors, namely 3W1B, 2W2B, and 1W3B systems referring to the sensor fusion of 3 wireless LAN and 1 Bluetooth, 2 wireless LAN and 2 Bluetooth, and 1 wireless LAN and 3 Bluetooth access points, respectively.

1) SELFLOC with a Single Technology

The mean distance errors for estimation using single technology sensors are depicted in Fig. 12. Mean distance errors of the base TN, KNN, and SMP algorithms are shown in the first three columns of Fig. 12. Mean distance error for fusion of all three algorithms (TN + KNN + SMP) using SELFLOC is shown in the last column. Accuracy of the Bluetooth system (3B) is generally better than that of the wireless LAN systems. KNN achieves better performance than TN and SMP for the same conditions. The SELFLOC fusion of all three algorithms is better than any single algorithm by 0.4 – 3.6 m in mean distance error. Overall, the 3B system fusing all three algorithms is the best performer, with mean distance error slightly higher than 4 m.

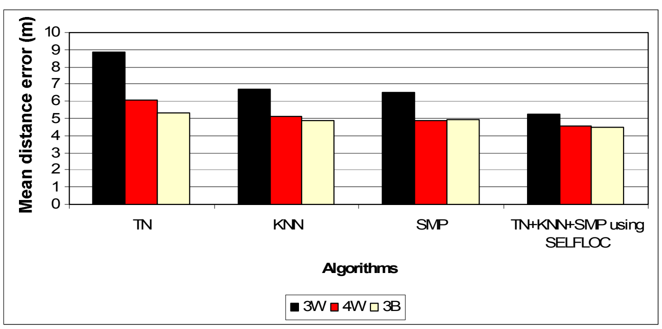

Figure 12. Mean distance errors of single technology sensor estimation with base algorithms

Fig. 13 depicts the mean distance errors for the SELFLOC-enhanced algorithms, eTN, eKNN, and eSMP, using single technology sensors. On the last column of Fig. 13, the mean distance error for fusion of all three enhanced algorithms (eTN + eKNN + eSMP) using SELFLOC is shown. The accuracy of the 3B Bluetooth system is better than the 3W wireless LAN system. However, the accuracy of the 4W system surpasses the 3B system. Mean distance error performance for fusion of all three algorithms is again better than any single algorithm by 0.3 – 2.4 m. The 4W system fusing all three algorithms is the best performer with mean distance error slightly higher than 3 m.

The SELFLOC enhancement to classical location algorithms (e.g, TN to eTN, KNN to eKNN, and SMP to eSMP) results in better accuracy. By comparing the mean distance errors displayed in Figs. 12 and 13, it is equivalent to an accuracy gain of 0.3 – 1.6 m. The results indicate more noticeable improvement in the wireless LAN systems.

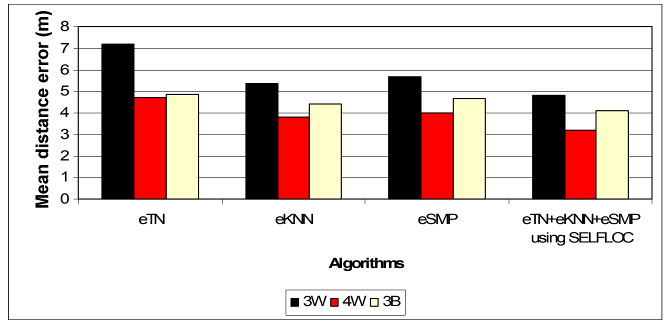

Figure 13. Mean distance errors of single technology sensor estimation with SELFLOC-enhanced algorithms

2) SELFLOC with Sensor Fusion

Fig. 14 depicts the mean distance errors for sensor fusion estimation using the base TN, KNN, and SMP algorithms. The last column designates the fusion of all three algorithms (TN + KNN + SMP) using SELFLOC. We first observe two similar trends as in the single technology case: a) KNN is better than TN and SMP; b) fusion of all three algorithms using SELFLOC is better than any single algorithm. Overall, the best performer is the 2W2B system whose best-case mean distance error is 3.0 m when fusing all three algorithms using SELFLOC.

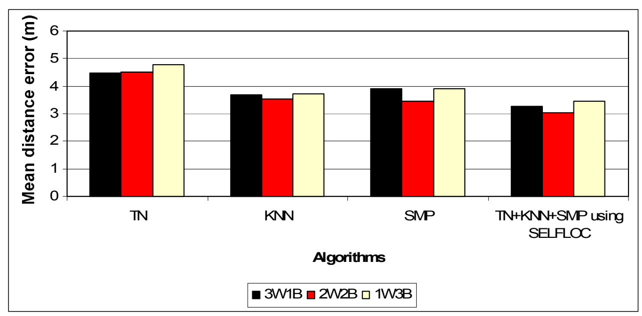

Figure 14. Mean distance errors of sensor fusion estimation with base algorithms

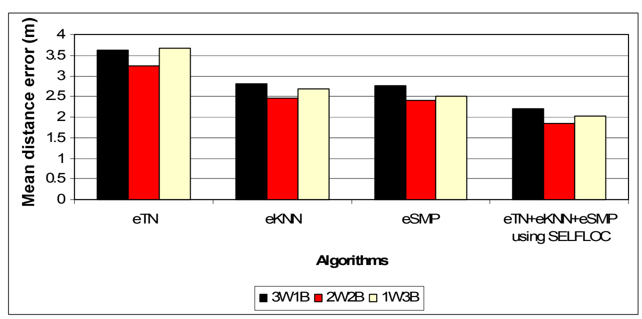

Figure 15. Mean distance errors of sensor fusion estimation with SELFLOC-enhanced algorithms

Fig. 15 depicts the mean distance errors for sensor fusion estimation using the SELFLOC-enhanced algorithms, eTN, eKNN, and eSMP. The last column provides the result for fusing all three algorithms. The 2W2B system again achieves the best performance. Its best-case mean distance error is 1.8 m when fusing the three enhanced algorithms using SELFLOC.

Sensor fusion estimation generally yields results superior to single technology estimation. We have obtained the best-case mean distance error of 3.1 m for the 4W system (see Fig. 13) and it is still worse than the mean distance errors of 2.7 m, 1.8 m, and 2.1 m for the 3W1B, 2W2B, and 1W3B systems, respectively (see Fig. 15).

3) SELFLOC Weight Localization

The SELFLOC weight localization requires dividing the entire floor into several regions for calibrating the SELFLOC weights separately for each region. We consider *three* regions for the systems, based on wireless LAN (i.e., 3W and 4W). Referring to the map and coordinate system of our testbed (see Fig. 11), we define Regions A, B, and C as areas contained by $x \in (0, 22)$ and $y \in (0, 6)$, $x \in (22, 28)$ and $y \in (0, 18)$, and $x \in (0, 22)$ and $y \in (7, 18)$, respectively.

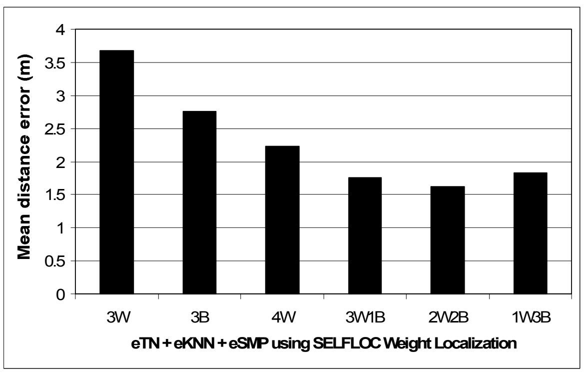

Figure 16. Mean distance errors of sensor fusion estimation with SELFLOC weight localization with iterative location refinement

On the other hand, there are *two* regions for any system using Blutooth (i.e., 3B, 3W1B, 2W2B, and 1W3B). Bluetooth regions are smaller areas than the wireless LAN's since the Bluetooth coverage is smaller. We define Regions 1 and 2 as areas contained by $x \in (0, 9)$ and $y \in (0, 11)$ and $x \in (9, 15)$ and $y \in (0,11)$.

Fig. 16 depicts the mean distance errors of the SELFLOC weight localization with iterative refinement. We observe further improvement in accuracy for all of the 3W, 3B, 4W, 3W1B, 2W2B, and 1W3B systems. Consistent with the previous results, the 2W2B achieves the best accuracy. Its mean distance error is 1.6 m and this is our best result for stationary location estimation. The maximum number of iterations was 3, indicating that ping-ponging across more than 2 regions can occur.

4) Fusion with a Commerically Available Location System Using SELFLOC Algorithm

Ekahau Positioning Engine™ [11] is a commercially available location system based on the IEEE 802.11 wireless LAN technology. We installed an Ekahau system with three wireless LAN access points. Then, we fused its estimates with the results of the 3W and 3B eKNN systems using the SELFLOC algorithm. Fig. 17 depicts the mean distance errors for Ekahau, 3B eKNN, 3W eKNN, and the fusion systems.[††]

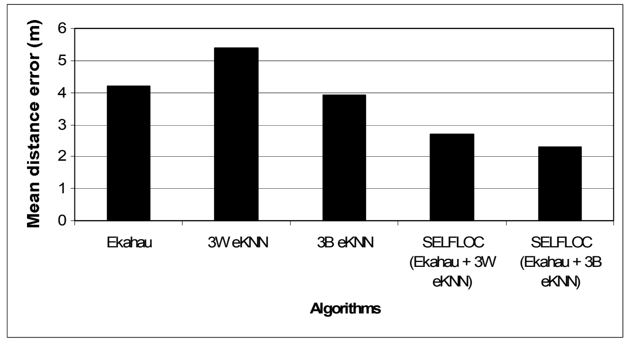

Figure 17. Mean distance errors of Ekahau, 3W and 3B eKNN, and fusion estimation using SELFLOC

Ekahau achieves better accuracy than the 3W eKNN, but slightly worse than the 3B eKNN. When Ekahau is fused with the inferior 3W eKNN, the fused system yields better performance than both Ekahau and the 3W eKNN. The same occurs for the case of Ekahau and the 3B eKNN. The latter (Ekahau and the 3B eKNN) provides better estimation by 0.4 m. The fusion with the 3W and 3B eKNN enables accuracy gains of 1.5 m and 1.9 m over the 3W Ekahau.

B. *Mobile Location Estimation*

For mobile user location estimation we have considered three different movement paths in our testbed. Paths 1, 2, and 3 are depicted in Figs. 19 and 20. Path 1 has a rectangular trajectory while Paths 2 and 3 are straight lines. Each path contains a series of grid points that are contiguous. When crossing each grid point in a path during the actual measurement process, the mobile measurement module requires a special key stroke to distinguish received signal strength samples acquired from different inter-grid point intervals.

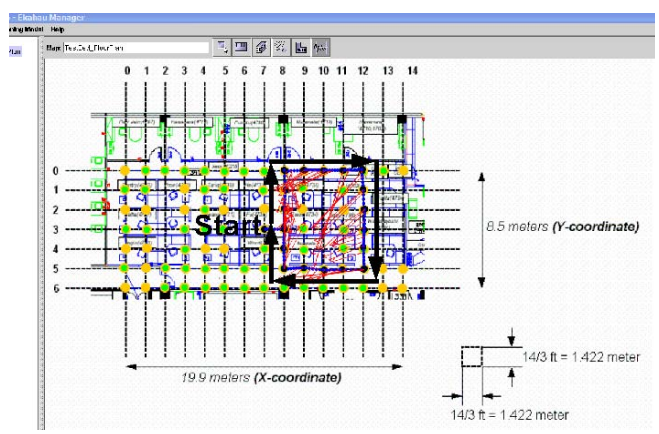

Figure 18. Mobile user Path 1

[††] We stress that our results are based on limited experimental evaluation of Ekahau Positioning Engine™ [11] in our environment and may not represent the whole effectiveness of Ekahau products in general.

We consider the RoC algorithm with eKNN for our mobile location estimation since the eKNN mean distance error performance in general was found to be the best. We also consider the RoC algorithm with Ekahau Positioning Engine.

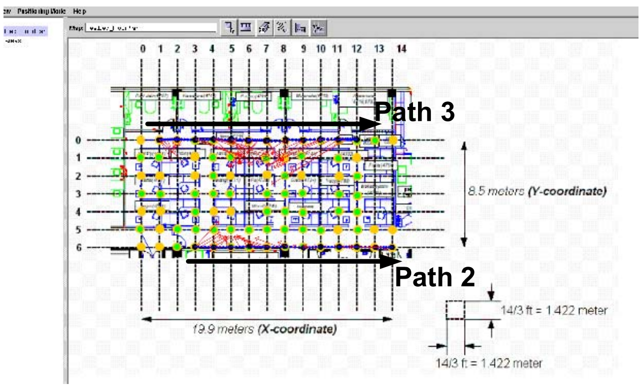

Figure 19. Mobile user Paths 2 and 3

In the following results, we consider a number of variations to form the region of confidence. In the RoC algorithm for single technology sensors, the region of confidence is drawn using the results of applying triangulation to either the wireless LAN or Bluetooth measurement only, as appropriate.

In the RoC algorithm for sensor fusion, the region of confidence is drawn using the results of applying triangulation to the mixed wireless LAN and Bluetooth measurements, i.e., 2W1B and 1W2B. In the RoC algorithm for Ekahau, the region of confidence is drawn using the results of applying triangulation to the wireless LAN measurement only since Ekahau is a wireless LAN-based system.

1) RoC Algorithm with eKNN for a Single Technology

Fig. 20 depicts the mean distance error performance of the RoC algorithm with eKNN for single technology sensors. The region of confidence is generated using either 3W or 3B. Overall, eKNN with the RoC algorithm achieved better accuracy than without the RoC algorithm. The RoC algorithm using wireless LAN achieves nearly 2 – 2.5 m accuracy gain whereas using Bluetooth is capable of 1.5 m accuracy gain. Even though the RoC algorithm using 3B eKNN yields better mean distance error than using 3W eKNN, the RoC improvement is more effective with the wireless LAN-based system.

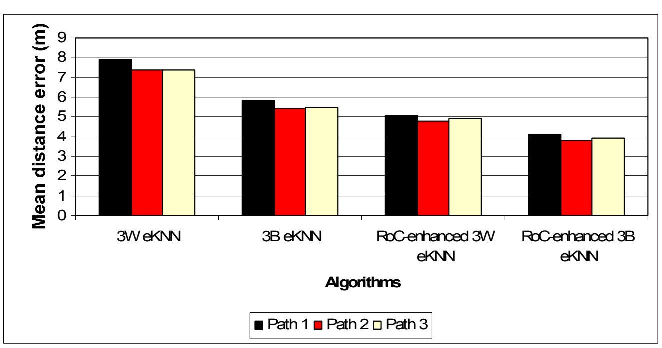

Figure 20. Mean distance errors of mobile location estimation using RoC algorithm with single technology

2) RoC Algorithm for Sensor Fusion

Fig. 21 depicts the mean distance error performance of the RoC algorithm with eKNN for sensor fusion. We consider 2W1B and 1W2B sensor mixtures. Again, both 2W1B and 1W2B systems with the RoC algorithm achieve better accuracy performance than without the RoC algorithm. The accuracy gain for 2W1B is almost 2 m which surpasses that of 1W2B by 0.5 m although the overall accuracy of the 1W2B system is better than the 2W1B system. However, accuracy of the 1W2B system is worse than the 3B system.

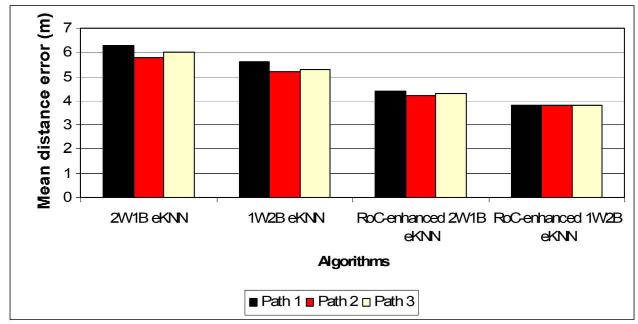

Figure 21. Mean distance errors of mobile location estimation using RoC algorithm for sensor fusion (multiple technologies)

3) Commercially Available Location System with RoC Algorithm

Fig. 22 depicts the mean distance errors of the base Ekahau system and Ekahau with the RoC algorithm. Ekahau uses three wireless LAN access points and we applied triangulation to form the region of confidence. We observe that the RoC algorithm improves the accuracy of the base Ekahau system for mobile location estimation. The accuracy gain is approximately 1.5 – 2 m, which is similar to the gain for the eKNN system using three wireless LAN access points.

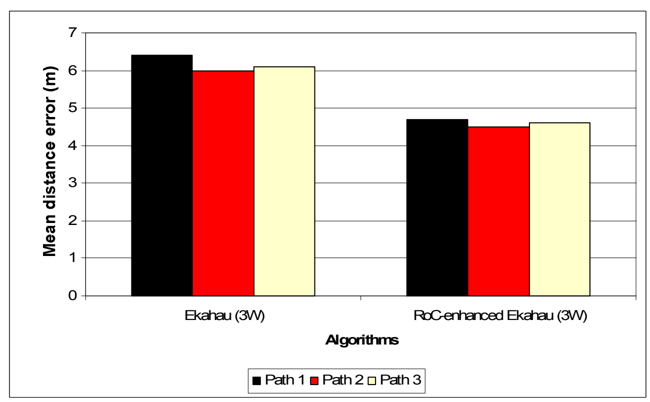

Figure 22. Mean distance errors of mobile location estimation using RoC algorithm for fusing Ekahau and eKNN

VII. Conclusions And Future Work

We have presented an experimental study of algorithms estimating the location of stationary and mobile users based on heterogeneous indoor RF technologies. We have proposed two algorithms, namely SELFLOC and RoC, which can fuse multiple information sources and location algorithms to improve accuracy and reliability of the estimation.

We have implemented our algorithms and empirically evaluated them using commercially available 802.11b wireless LAN and Bluetooth hardware. We also have implemented our algorithms in conjunction with a commercial location estimation product. In both cases, we have been able to achieve significant improvements in our metric, which is the mean distance error. We summarize our findings below.

1. *Benefits of using multiple technologies with SELFLOC.* We consider the effect of using Bluetooth access points instead of WLAN access points, while keeping the total number of access points fixed at 4. Even with the classical algorithms (TN, KNN or SMP), there is 11 – 28 % improvement in location error; this increases to 23 – 47 % with the same algorithms individually enhanced using weights generated using SELFLOC (i.e., eTN, eKNN, eSMP). Finally, when SELFLOC is applied to the enhanced algorithms, an improvement of 47 – 70 % is obtained.

2. *Benefits of using multiple algorithms with SELFLOC.* We observe in our study that different classical algorithms perform differently for different locations. We considered the effect of combining the outputs of multiple classical algorithms using weights generated by SELFLOC. We found that an improvement of 17 – 42 % could be obtained.

3. *Benefit of using SELFLOC with a commercial wireless LAN location system.* When the output of a commercial location system is combined using the eKNN algorithm, the same number of wireless LAN access points, and weights generated by SELFLOC, we were able to obtain a 36 % improvement in mean accuracy.

4. *RoC for mobile location estimation with a single technology.* We have found that for the movement paths studied using Bluetooth access points with Bluetooth RoC gave the best results, with a 21 – 29 % improvement compared to not using RoC.

5. *RoC for mobile location estimation using multiple technologies*. For a fixed number of access points, using Bluetooth access points instead of wireless LAN improves accuracy. The best results are obtained for 1W2B, with 24 – 38 % improvement compared to not using RoC.

6. *RoC for mobile location estimation using a commercial system.* Using RoC with the eKNN algorithm to enhance the performance of a commercial system resulted in a 57 – 62 % improvement in accuracy.

7. *Comparison of classical algorithms*. The mean distance error with KNN is consistently lower (by 17 – 36 %) than that for TN, while only slightly lower than SMP. The same relationship holds when each algorithm is enhanced using SELFLOC to weight the RF signal strength measurements: eKNN beats eTN by 11 – 23 %. This relationship generally holds in all our single-technology experiments, although generally the three wireless LAN case enjoys the most improvement. Finally, this relationship also holds when fusing data from multiple technologies.

8. *Comparison of technologies.* Bluetooth with 3 access points generally provided 23 – 40 % better error than 802.11b WLAN for the same number of access points, and comparable performance to that with 4 WLAN access points. This is consistent with the smaller radius of Bluetooth cells.

We have achieved the best mean distance error of 1.6 m when using the SELFLOC weight localization algorithm with iterative location refinement for multiple-technology location estimation. For this, the entire testbed is divided into smaller regions having separately calibrated sets of the SELFLOC weights. These weights are then applied for each individual algorithm as well as weighted combinations of multiple algorithms. We have also been successful in improving the accuracy of a wireless LAN-based commercially available location system by almost 2 – 3 m using the SELFLOC-enhanced eKNN algorithm.[†††]

We have used RoC to estimate the location of a mobile user and shown that similar results apply. In particular, we have achieved the best mean distance error of 3.8 m when using a single technology with eKNN. When using multiple technologies, the mixture of one wireless LAN and two Bluetooth access points (1W2B) improves the mean distance error compared to a pure wireless LAN system (3W) from 4.8 m to 3.8 m. Finally, using RoC and eKNN in conjunction with a commercial location estimation system can improve the error from 6 m to 4.5 m.

Our future work includes a number of open issues for RF-based location estimation. Reducing the manual effort related to offline calibration and measurement is the most urgent issue for practical deployment. Another technical challenge arises when an access point is transmitting beacons at variable power levels since the location database entries and calibration can be invalidated. We are currently investigating these issues.

ACKNOWLEDGMENT

We acknowledge Jonathan Wood and Raghu Dendukuri of DoCoMo Communications Laboratories USA, Inc. for their contribution to implementation and experiments.

REFERENCES

[1] Bahl, P., and Padmannabhan, V., N., “RADAR: An In-Building RF-based User Location and Tracking System, ” IEEE Infocom 2000, Tel-Aviv, Israel, March 2000

[2] Castro, P., et al., “A Probabilistic Location Service for Wireless Network Environments (Nibble),” Ubiquitous Computing 2001, Atlanta, Georgia, September 2001

[3] Pandya, D., Jain, R., and Lupu, E., “Indoor Location Using Multiple Wireless Technologies,” IEEE PIMRC, Beijing, China, September 2003

[†††] We stress that our results are based on limited experimental evaluation of Ekahau Positioning Engine™ [11] in our environment and may not represent the whole effectiveness of Ekahau products in general.

[4] Ladd, A., M., et al., "Robotics-Based Location Sensing Using Wireless Ethernet," Mobicom 2002, Atlanta, Georgia, September 2002

[5] Smailagic, A., and Kogan, D., "Location Sensing and Privacy in a Context-aware Computing Environment," IEEE Wireless Communications, Vol. 9 No. 5, October 2002

[6] Pandya, D., "A Hierarchical Location Tracking System Project Report" Master's degree thesis, Department of Computing, Imperial College, London, UK, June 2002

[7] Hightower, J., and Borriello, G., "Location Sensing Techniques," IEEE Computer Magazine, August 2001

[8] The linux-wlan^{TM} Project, http://www.linux-wlan.com/

[9] BlueZ, Official Linux Bluetooth Protocol Stack, http://bluez.sourceforge.net/

[10] Widrow, B., and Stearns, S., "Adaptive Signal Processing," Prentice Hall, Upper Saddle River, New Jersey, 1985

[11] Ekahau Positioning Engine^{TM} 2.0, http://www.ekahau.com

[12] Bahl, P., Padmanabhan, V., N., and Balachandran, A., "Enhancements to the RADAR User Location and Tracking System," Microsoft Research Technical Report: MSR-TR-00-12, February 2000

[13] Jain, R., "Gazillions of Gizmos: What Are the Challenges?," Panel Presentation, International Conference on Data Engineering (ICDE), April 2001, http://www.docomolabs-usa.com/researchers/RJain