Indoor Position Estimation System for Wide Area Augmented Reality Applications

Chaoran Yu* Plamen Levchev† Michael Krishnan‡ Avideh Zakhor§
Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA, United States

ABSTRACT

An important problem in wide area augmented reality applications is to localize the user accurately so as to superimpose metadata in the user's field of view. This task is easier in outdoor environments since GPS is available. Also, for small area AR applications consisting of a scene with a great deal of texture, computer vision techniques can be used to determine a user's pose. In this paper, we propose an end-to-end system which can be used to (a) construct 3D models and associated 2D floor plan of large scale indoor environments in a fast, automated, scalable way; (b) construct multiple sensor e.g. WiFi and imagery signature databases for the same environment; (c) use the reference databases in part (b) to recover position of users wearing an AR glass or carrying a mobile device equipped with a camera and WiFi sensor. The system consists of a man portable backpack of sensors carried by an operator inside buildings walking at normal speeds. The sensor suite consists of laser scanners, cameras and an IMU. Particle filtering algorithms are used to recover 2D and 3D path of the operator, a 3D point cloud, the 2D floor plan, and 3D models of the environment. The same walkthrough that results in 2D maps also results in multi-modal sensor databases, in our case WiFi and imagery. The density of the resulting WiFi database in our system is considerably higher than existing systems due to continuous, rather than stop-and-go, WiFi signature acquisition. We use particle filtering algorithms in an Android application to combine inertial sensors on the mobile device or glass, with 2D maps and WiFi and image sensor databases to localize the user. Experimental results for the second floor of the electrical engineering building at UC Berkeley campus will be shown.

Keywords: Augmented reality, image-based localization, WiFi indoor localization, inertial sensors, particle filtering

Index Terms: I.4.9 [Image processing and computer vision]: Applications—Image-based localization; C.2.m [Computer-communication networks]: Miscellaneous—WiFi indoor localization

1 INTRODUCTION AND RELATED WORK

In recent years, indoor localization has received a great deal of attention among researchers. On one hand, it has a number of important applications such as location-aware intelligent shopping assistant and indoor real-time navigation. On the other hand, it is a technically challenging problem due to the fact that most buildings virtually block GPS signals. Hence alternate localization approaches are needed for the indoor environment.

The prevalence of WiFi infrastructure inside most buildings provides a natural starting point for this problem. A well-known approach is to construct a database of WiFi Received Signal Strength Indicator (RSSI) fingerprints for the building. The RSSI fingerprint for each location is a vector of decibel values where each row corresponds to the WiFi signal strength of a particular access point detected at that location. As the client-side application queries the database with an RSSI measurement, algorithms such as Redpin [1] or variants of $k-$NN [20, 23] are used to retrieve the location whose fingerprint is closest to the querying fingerprint. A major advantage of this method is its cost-effectiveness since hardware infrastructure is already in place in most indoor commercial and residential buildings. Furthermore, practically all mobile phones and consumer electronic devices have WiFi scanning capability. One disadvantage of this approach is that the location dependency of RSSI is not reliable and is subject to interference. Even though room-level accuracy e.g. approximately 5 to 10m has been demonstrated, the method does not achieve meter or sub-meter level accuracy [22].

Another approach that could be used for localization is inertial dead reckoning using wearable glass or smartphone's on-board accelerometer, gyroscope and magnetometer. Nowadays consumer electronic devices are quipped with increasingly more accurate sensors that are capable of sampling at fast data rates. Utilizing these sensors, users' speed and orientation of movement can be estimated and their path can therefore be tracked in real time. Magnetometer can be used to get users' orientation, but it suffers from interference from various indoor magnetic anomalies such as steel cabinets. Integrating raw accelerometer and gyroscope readings provides displacement information but integration introduces significant drift error [2]. To estimate users' movement, a commonly adopted method is to first detect their steps, and then estimate the corresponding step lengths [25, 26, 31]. The inertial sensor-based approach is different from the WiFi-based one in that it only estimates relative change of position, whereas WiFi measurements provide position estimates in the global coordinate frame. In an attempt to combine strengths of above two approaches and combat their respective weaknesses, it is possible to use a probabilistic technique, e.g. particle filtering, to derive an integrated location estimate. Specifically, work by Li et al. [3] has demonstrated that the combined system can achieve a mean tracking accuracy of 1.5m when the smartphone is held in hand.

Recently an image-based indoor localization scheme [4, 5, 16, 18] has been proposed for mobile devices with cameras. Firstly, a database of images is constructed via a man portable ambulatory backpack of sensors, and then images taken by the client mobile device are used to retrieve those images with most number of matching features in the database. The Shift Invariant Feature Transform [6] finds distinct features contained in images that are invariant to uniform scaling and partially to affine distortion. Even though this method generally achieves higher accuracy than WiFi RSSI matching, its performance is degraded when the query image has few distinguishing features, or when the pictures are of low quality due to out-of-focus and/or motion blur. In the first case, lack of distinct features would adversely affect the ranking of retrieved database images. In the second case, it is difficult to detect and extract features from blurry images. Liang et al. [4, 5] have shown that in the static case, where blur-free images are captured with a stationary camera, image-based localization could achieve an accuracy of 2

*e-mail: chaoran_yu@eecs.berkeley.edu
†e-mail: levchev@eecs.berkeley.edu
‡e-mail: mkrishna@eecs.berkeley.edu
§e-mail: avz@eecs.berkeley.edu

meters over 80% of times, and 4 meters over 90% of times.

To achieve multimodal indoor localization, a mechanism is needed to combine various sources of location estimates. Particle filtering is one of the most widely used techniques for sensor fusion due to its flexibility [21, 24, 29]. Non-probabilistic techniques such as that proposed in [19] are not suitable for dynamic tracking of a user's indoor position since they can not produce an integrated location estimate until both WiFi and image results are available. Furthermore, a reliable confidence metric for an indoor localization modality can be quite challenging to obtain.

In this paper, we propose an end-to-end system which can be used to (a) construct 3D models of large-scale indoor environments in a fast, automated, scalable way; (b) construct multiple sensor e.g. image and WiFi signature databases for the same environment; (c) use the reference databases in part (b) to recover position of users wearing an AR glass or carrying a mobile device equipped with a camera and WiFi sensor. The system consists of a man portable backpack of sensors carried by an operator inside buildings walking at normal speeds. The sensor suite consists of laser scanners, cameras and an IMU [8, 11, 12, 13, 14]. Particle filtering algorithms are used to recover 2D and 3D path of the operator, the 3D point cloud, the 2D floor plan, and 3D models of the environment. The same walkthrough that results in 2D maps, is used to generate multi-modal sensor databases, in our case WiFi and imagery. The spatial density of the resulting WiFi database in our system is considerably higher than existing methods due to continuous, rather than stop-and-go, movement of the operator during data acquisition. We use particle filtering algorithms in an Android application to combine inertial sensors on the mobile device or AR glass, WiFi and image sensor databases, and a 2D map of the environment to localize the user. Experimental results for the second floor of the electrical engineering building, Cory Hall, at UC Berkeley campus will be shown in Section 5.

2 DATA COLLECTION SYSTEM OVERVIEW

In previous work, we have demonstrated that an ambulatory backpack, shown in Figure 1, equipped with two cameras, one orientation sensor, and five 2D laser range sensors can be used to generate a 2D floor plan and 3D model of an indoor environment with a single walkthrough of the building with 10cm average 3D position accuracy [8, 11, 12, 13, 14]. Our system builds on top of that work by adding WiFi scanning capability to the backpack. Specifically, we use three USB AirPcap NX cards from Riverbed Technology to capture 802.11 beacons. As the human operator walks with the backpack, the Multi-Channel Aggregator captures WiFi beacons on all supported channels in the 2.4GHz and 5GHz bands, recording the timestamp and RSSI of each beacon. The operator walks at normal speeds, i.e. at about 0.7 m/s, to generate a dense WiFi signature database. Thus, with one walkthrough it is possible to generate both a 2D map of the environment and the associated WiFi and image databases for indoor localization.

There are a number of ways to generate the WiFi database: since the beacon period for each channel is 102 milliseconds (ms), it is possible to use 3 AirPcap cards to scan each channel for 102 ms before switching to a new channel. Assuming there are a total of 30 channels to scan in the 2.4 and 5 GHz bands, this results in 1 second end-to-end scanning of all 30 channels; hence at walking speed of 0.7m/s, the spacing between fingerprint data points in our WiFi database is approximately 0.7m. To minimize the overhead associated with switching between channels, it is also possible to scan each channel for a period of 204 or even 306 ms so as to obtain 2 or 3 beacons for each channel before switching to the next one. In doing so, we can average the two beacons for the 204 ms case, or take the median of the 3 beacons in the 306-ms case for each entry in our WiFi signature database to reduce the noise. However, this comes at the expense of lower spatial density of the fingerprints in our database. Specifically, assuming the same walking distance, the spatial density of 204 (306) ms case is half (a third) of that of the 102 ms database. An example of 306 ms database superimposed on the 2D floor plan of the 2nd floor of Cory Hall is shown in Fig 2. As shown, for a donut shaped square corridor of dimension 35 meters, there are 76 fingerprints in our database. We use this database for the experiments in this paper.

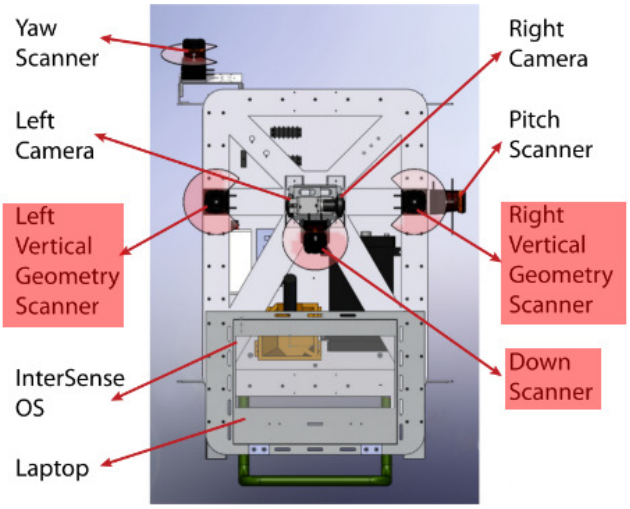


Figure 1: Ambulatory backpack equipped with various sensors.

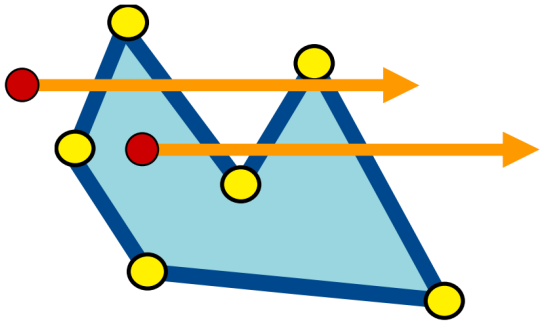


Figure 2: Ray tracing used to determine validity of a given coordinate.

We have experimentally verified that a normal walking speed of 0.7 m/sec does not result in significant blur to adversely affect the performance of our image-based localization method [4, 5, 16, 18]. Images taken with two fisheye cameras are post-processed using algorithm in [4, 5, 18] to construct a database containing processed images, their corresponding SIFT features, pose information and depth maps.

To summarize, with one walkthrough at normal walking speed with the backpack shown in Figure 1 we generate 2D maps and multiple sensor modality e.g. WiFi and image databases for an indoor environment in order to enable indoor positioning.

3 POSITIONING WITH WIFI

The WiFi database consists of MAC addresses detected at different locations and their corresponding RSSI values. The fingerprint associated with each point is a variable-sized set of of tuples of the

form (access point MAC address, RSSI). Since the operator of the backpack moves continuously, and since switching between channels introduces delays, the set of MAC addresses and their fingerprint, is not strictly associated with a single location. We compute the mean timestamp of all beacons in the fingerprint and associate the location of backpack at that mean timestamp with the fingerprint. Consequently the smearing associated with 306 ms scans results in greater position uncertainty than that of 102 ms scans.

When the user queries the WiFi localization database with an RSSI measurement, the WiFi localization server computes the Redpin [1] scores for all database points, and sorts them in descending order. We then build clusters starting with the highest score, adding a point to a cluster if it is within a certain fixed distance from any point in that cluster [15]. If multiple clusters are within this distance, it is assigned to the one containing the element with the highest score. Once a cluster of size 3 is created, we return the location of the database point closest to the centroid of that cluster. The motivation behind this approach is that although the closest point to the ground truth might not be the top result, there is likely a set of points close to ground truth and consequently very close to each other among the top results.

4 POSITIONING WITH INTEGRATION OF WIFI, INERTIAL SENSORS AND IMAGES

4.1 Particle Filtering and Motion Model

We use particle filtering to fuse inertial sensors, WiFi readings and images to localize the mobile device or the AR glass of the user. One strength of particle filtering with respect to other filtering algorithms is that it allows us to improve estimation accuracy by increasing the number of particles at the expense of increased computational complexity. The other aspect of our problem that makes particle filtering particularly attractive is that it can incorporate observations from WiFi and image localization estimates arriving at different rates. In the following subsections, components of the particle filtering are discussed.

Our motion model consists of user step length and orientation. We process accelerometer readings using algorithms in [2, 27] to detect steps and estimate their lengths. Then we combine step information with orientation to propagate particles [10, 25, 26, 28, 30, 31].

4.2 Observation Update: WiFi and Images

Initially, all particles are given unit weight. Upon receiving a WiFi or image coordinate estimate, each currently living particle is assigned an importance factor that incorporates the observation. Thus, the importance represents the probability of observing a measurement given the particle's current position. Each particle's weight is multiplied with the importance factor to obtain an updated weight.

The importance factor is obtained using a simple Gaussian parametric form:

$$f_i^{t+1} = \frac{1}{\sigma\sqrt{2\pi}} \exp \frac{-((x_z - x_i^t)^2 + (x_z - y_i^t)^2)}{2\sigma^2} \quad (1)$$

where (x_z, y_z) is the coordinate returned by WiFi or image localization server, f_i^{t+1} is the importance factor of particle i at new timestamp $t+1$, and σ is the variance of the Gaussian distribution. A smaller σ increases the influence of WiFi or image observations on particle weights. As σ approaches ∞, observations are not used to update particles at all; when σ is 0, the newly updated particle is assigned a large importance factor. Therefore, σ should be inversely proportional to the importance of the results returned by the WiFi or image localization server. In our implementation, σ for WiFi updates is a constant 2.0 in normal cases and 5.0 after most e.g. >90% particles have collided with walls and disappeared. The reason for choosing a larger σ in the wall-collision scenario is that when this happens, current location estimate is more likely to be erroneous, and therefore more of existing particles need to be kept in order to maintain this uncertainty. For image updates, we take advantage of retrieval confidence value c [5] returned by image localization server and assign $\sigma = 1 - c$. Finally, the particle's weight is updated as

$$w_i^{t+1} = f_i^{t+1} \times w_i^t \quad (2)$$

Although both WiFi and image results modify particle weights using Equation (1), there are differences between them. In our implementation, retrieving a coordinate from the WiFi database takes 1 to 2 seconds, whereas obtaining a coordinate from the image server takes 4 to 6 seconds. This temporal difference necessitates different treatment of two estimates. Since pictures are captured as the user is moving, by the time the image localization server returns a location estimate, the user is no longer in the original location the image was captured. In contrast, we find that 1 to 2 seconds of delay for WiFi could be safely ignored. The inherent coarser spatial granularity of WiFi-based localization also makes the delay less important.

Since the image localization returns an estimate of past location, we can use that estimate to correct particle weights in the present timestamp in the following manner. Assuming a picture is taken and sent to the image server at timestamp t_s, we record the current position (x_s, y_s) as approximated by the centroid of all particles. Starting from t_s, as the user walks, the steps and their respective headings are maintained. This *step history*, $(l_1, \theta_1), (l_2, \theta_2), \ldots, (l_n, \theta_n)$, is complete when the user receives the response (x_i, y_i) from the image server at time t_e. Then the received coordinate (x_i, y_i) is propagated backward in time to a new value according to the motion model discussed in IV-B and is replaced by a new location denoted by (x_i', y_i'). Finally (x_i', y_i') is used in Equation (1) to update the current particle cloud at timestamp t_e.

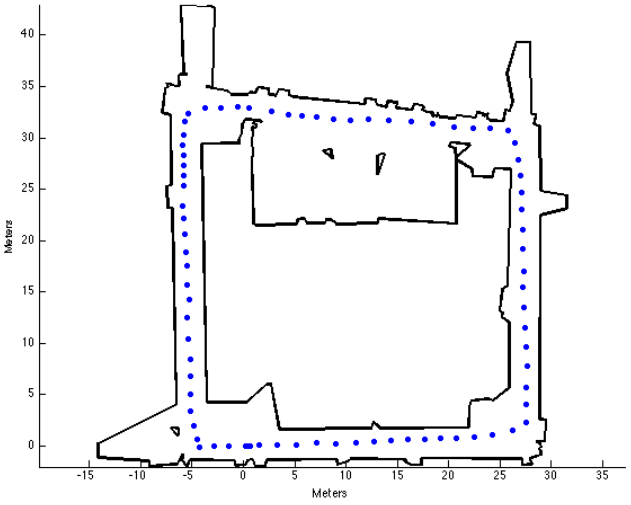
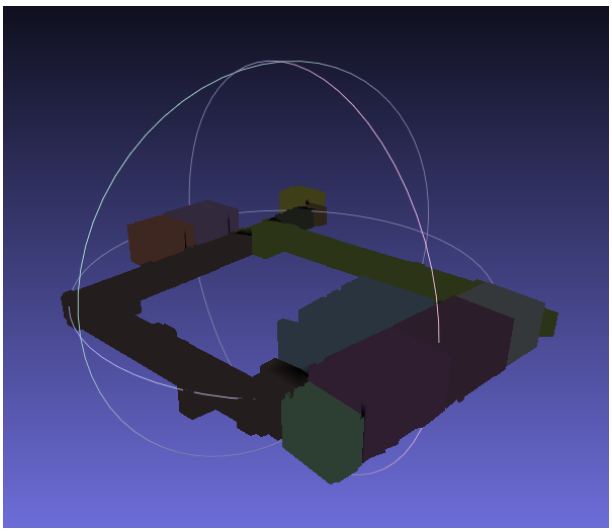


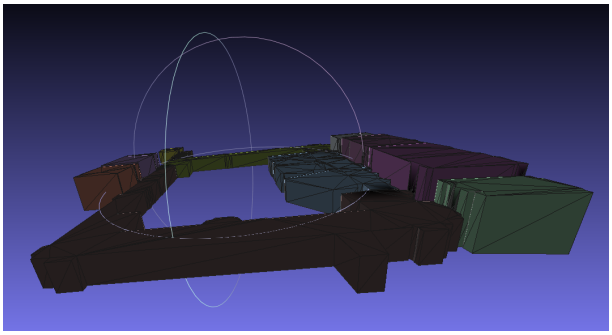
Figure 3: Spatial distribution of data points in our WiFi database, where each circle represents a WiFi signature at the corresponding location.

4.3 Resampling

We adopt a simple resampling criterion: Whenever the number of living particles is less than 10% of the initial particle count, resampling is triggered. The resampling procedure is described in Algorithm 1. The probability mass function is a mapping $\mathscr{L} \rightarrow [0,1]$,



(a)



(b)

(c)

Figure 4: (a, b) 3D model (c) 2D floor plan and room labels for second floor of Cory Hall.

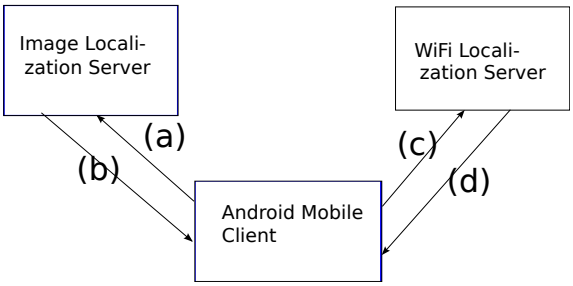


Figure 5: Client-side architecture and communication paths. The Android mobile client is either a smartphone or an AR glass. Arrows indicate direction of message transfer. Message content for each arrow is as follows. (a) Picture taken by smartphone camera. (b) Image-based location estimate (x_i, y_i). (c) WiFi measurements. (d) WiFi-based location estimate (x_w, y_w).

where $\mathscr{L}$ is the set of distinct coordinates represented by all current living particles. The values on the right hand side of the mapping are probabilities that are proportional to weights of particles at each location. All particles in the new cloud have unit weight, but there will be more particles at locations where the original cloud has greater weights. The purpose of resampling is to reduce diversity of particles in terms of both location distribution and weights by aggregating particles at locations that are more likely to be the true location of the user. After resampling step, the total number of particles is restored to the initial total particle count, N.

Data: Set M consisting of all current living particles and initial total particle count N
Result: Resampled set of living particles
$result = \emptyset$;
Compute probability mass function (PMF) from which resampling is done;
while $iteration < N$ **do**
  Generate a new particle from computed PMF;
  Assign the newly created particle unit weight;
  Add it to $result$ set;
  $iteration = iteration + 1$;
end
$M = result$;

Algorithm 1: Resampling Procedure

4.4 Location Report

After fusing multiple sources of information, the particle filter returns a single location estimate, which is the centroid of all the living particles. In computing the average position of the particle cloud, it is possible for the centroid to lie outside of the valid region e.g. outside of the walls of floor plan. In that case, we use binary search along the past trajectory to find the closest valid point as the answer.

To determine whether a point is in valid regions we use ray tracing as shown in Figure 2; specifically we draw a line from an arbitrary point outside of the polygon to the point in question, check whether the number of edges crossed is odd, and make sure that if the crossing is a vertex, it is counted only once. Odd number of crossed edges/vertices would indicate a valid point. This result is known in topology as Jordan curve theorem [7].

5 EXPERIMENTAL RESULTS

In this section, we evaluate the performance of our system consisting of step detection/step length estimation algorithm, and the

tracking/localization system for the second floor of the U.C. Berkeley electrical engineering building, Cory Hall. Our WiFi database used for these experiments is constructed with 306 ms channel dwell time and there are a total of 76 data points in the database as shown in Figure 3. The entire path is about 130m, which on average results in about one WiFi fingerprint per 1.71m in the database. The data collected by the backpack is post-processed using algorithms presented in [8, 9] to generate a 3D model and the 2D floor plan with room labeling of the second floor of Cory Hall, as shown in Figure 5.

5.1 Client-side System Architecture

The tracking system on the client side is implemented and tested on Android platform with two devices: Samsung Galaxy S4 smartphone and Vuzix M100 glass. Vuzix M100 runs Android 4.0 and Galaxy S4 runs Android 4.3; both system versions are recent enough to support required sensor APIs. Vuzix M100 has only 2.4GHz WiFi whereas Galaxy S4 has both 2.4GHz and 5GHz capability. The system architecture is shown in Figure 4.

The glass or smartphone acts as the client that collects WiFi measurements. The user takes a picture using the device every once in a while. Whenever a full WiFi scan is completed for both 2.4GHz and 5GHz channels, the client device sends the recorded RSSI to the WiFi localization server and then initiates another WiFi scan. Similarly, After a picture is taken, the image data is transmitted to the image localization server. After computation at localization servers is completed, coordinates resulting from WiFi and image are sent back to the client, though most likely not simultaneously. No sooner does the client receives a response i.e. coordinate from one of the localization servers, than it feeds that information to particle filtering algorithm. The client also detects steps and estimates step length in real time and incorporates that information in the particle filter.

5.2 Tracking with Inertial Sensors and WiFi

In Figures. 6 through 9, WiFi measurements are marked using red diamonds, image updates with correct retrieval are marked as green asterisks, and image updates with incorrect retrieval are marked as yellow circles. The floor plan has the same scale as the one shown in Figure 3, i.e. second floor of Cory Hall, but axes are not shown. Blue lines connect dots resulting from our location estimation process for a user holding a smartphone or wearing an AR glass. The ground-truth paths start at lower left corner, i.e. (0, 0) in Figure 3, go along the hallway to the lower right corner at (27, 0), turn left to arrive at (27, 31), turn left again to the upper left corner at (-5, 34), finally turn left once more to return to the start location.

Figure 6 shows two different scenarios of our experimental results. The floor plan is generated by post-processing backpack data as discussed earlier and is aligned with respect to global coordinate frame. The orientation data obtained from client device sensors are in global coordinate frame. Figure 6(a) shows a path generated by real-time tracking the user's position using inertial sensors and WiFi measurements on Galaxy S4. The communication method, shown as arrows in Figure 4, used for this case is WiFi. There is a noticeable jump at the lower left corner of the map due to poor connectivity of WiFi at that area. The entire walk has a total of 21 WiFi measurements and 162 detected steps.

Figure 6(b) is obtained in a similar way to Figure 6(a) except that 4G communication is used when WiFi connectivity is lost. This way the mobile client rarely loses connection with the server and the obtained path looks markedly smoother. In Figure 6(b), there are a total of 18 WiFi measurements and 170 detected steps. Figure 7 shows the path obtained using Vuzix M100 AR glass. There are a total of 22 WiFi measurements and 180 detected steps.

Paths in Figures. 6(a) and 6(b) are more accurate and look smoother than Figure 7 because Galaxy S4 has much better WiFi reception i.e. is capable of detecting more access points at a given location, than Vuzix M100. Furthermore, Vuzix M100 suffers from lower WiFi localization accuracy since it does not support 802.11n, i.e. 5GHz band. Finally, Vuzix M100 does not have 4G connectivity, so WiFi is the only option for data transfer between the AR glass and servers.

5.3 Tracking with Inertial Sensors and Images

Figure 8 shows the resulting path of integrating sensors and images. The mobile client transfers images to the server and receives coordinates using 4G connection. There are a total of 10 image updates and 123 detected steps. Among 10 image localization attempts, 1 failed to retrieve correct matching image in the database. The corresponding coordinate is (-6.1, 21.4) and is marked as a yellow circle in Figure 8. Since steps only estimate incremental changes of position, the error caused by this retrieval failure does not get fixed until next image update arrives, which has a correct retrieval and therefore results in an accurate location estimate at (17.0, 31.6). This path has more jumps and discontinuities than those obtained with WiFi only in Figures. 6 and 7. Even though image localization generally achieves higher accuracy than WiFi, its server processing time is longer. A user walking at normal speeds would have moved quite a distance between successive image localization results. Although we have dealt with this issue, the slow data rate still has an adverse effect on the smoothness of the path.

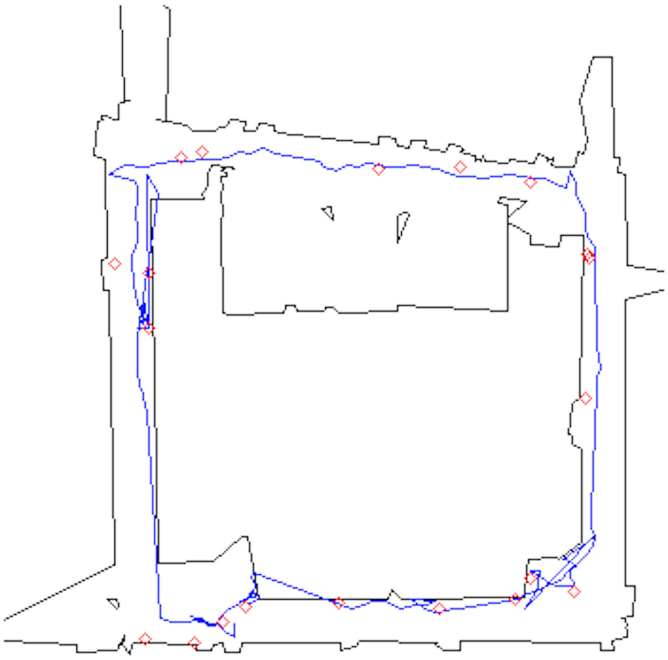
5.4 Integration of WiFi and Images

Figure 9 shows the resulting trajectory for the Galaxy S4 after the integration of all available information, i.e. inertial sensors, WiFi and images. Communication between the client device and localization servers is established with 4G. There are a total of 27 WiFi updates, 10 image updates and 126 detected steps. Steps are detected quite frequently, matching the walking speed of the user; WiFi retrieval arrives every 3 to 5 meters; image updates arrive least frequently, matching slow processing time of images on the server. Among 10 image localization attempts, there is one failed retrieval at coordinate (2.8, 0.0) marked as a yellow circle in Figure 9. Before this image update, the estimated user location was at (25.4, 3.8), while after feeding this incorrect image update into the particle filter, centroid of all the particles is dragged back to the start of the corridor i.e. lower left corner near the start location. Unlike in Figure 8, this adverse effect does not last long as the WiFi measurement shown as the red diamond at (20.6, 1.6) in Figure 9 arriving immediately after the erroneous image update, rapidly fixes this error.

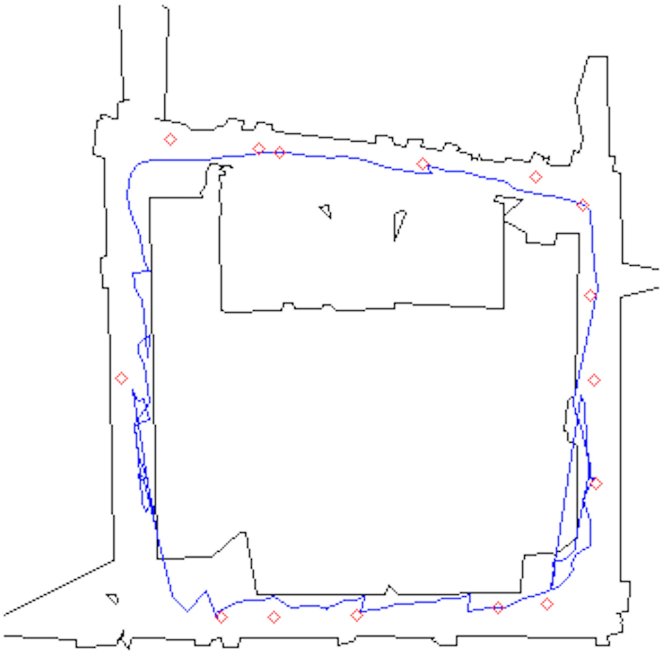
6 DISCUSSION AND FUTURE WORK

While Figures 6 through 9 demonstrate the performance of various localization methods qualitatively, it is important to also characterize their performance quantitatively. Specifically, Table 1 shows a number of metrics for comparing the paths in these figures. The first metric has to do with the number of jumps where a jump is defined as any movement of more than 3 meters long; the motivation behind this metric is that the user cannot possibly move more than 3 meters in any one step. The second metric has to do with total distance associated with all jumps. The third metric has to do with the number of backward jumps indicating the situations in which the step estimates are "ahead" of the actual location of the user by at least 3 meters. These jumps result in visually inaccurate and counter-intuitive paths. The forth metric has to do with the overall length of each path: intuitively, if there are too many backward jumps the total length of the path increases. As mentioned earlier, the ground-truth path length is approximately 130 meters.

As seen, the highest performance method in terms of all metrics is WiFi only with 4G communication. Interestingly, combination of image and WiFi results in worst performance, even though there



(a)



(b)

Figure 6: Trajectory obtained in real time through integration of sensors and WiFi measurements using (a) WiFi (b) 4G connections.

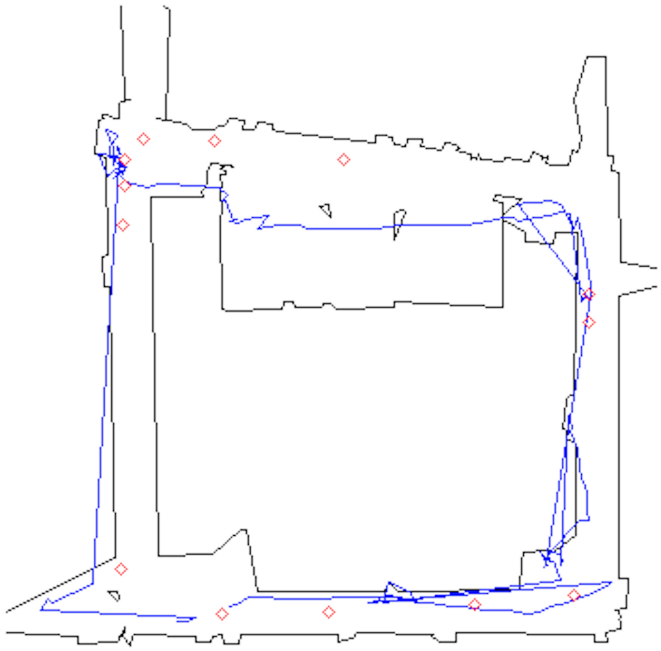


Figure 7: Trajectory obtained on Vuzix M100 in real time using inertial sensors and WiFi using WiFi connection.

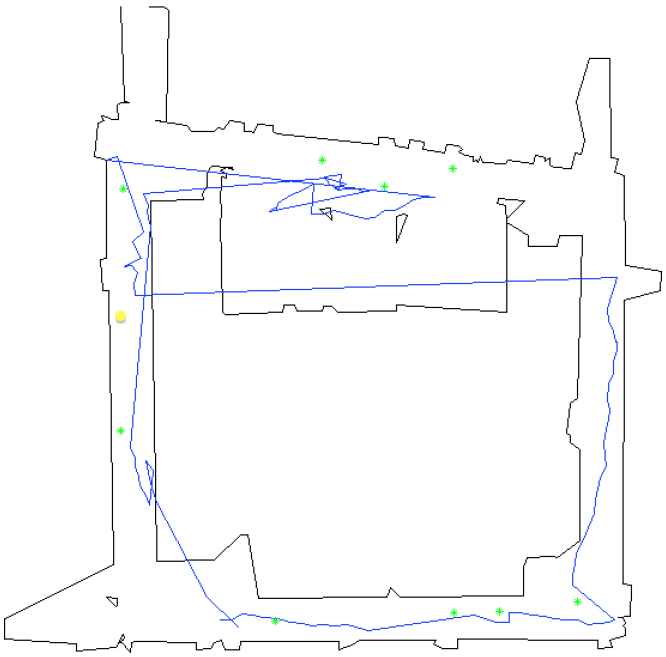


Figure 8: Trajectory obtained on Galaxy S4 in real time using inertial inertial sensors and images using 4G connection.

Figure 9: Trajectory obtained on Galaxy S4 in real time through integration of sensors, WiFi measurements as well as images using 4G connection.

is only one incorrectly retrieved image. Since in this case, we use 4G communication as a back-up when WiFi connectivity is lost, transmission of images to the server creates contention at the WiFi antenna, delaying and disrupting the necessary WiFi scans for WiFi localization. This explains the numerous jumps for the combined WiFi and image localization method. One possible solution would be to always communicate the pictures on 4G; however, in our scenario, there is no 4G connectivity for approximately 25% of the path. Another remedy is to increase the compression rate to reduce image transmission time. This could however adversely affect image retrieval and matching accuracy. Another possibility is to cache image database on the mobile device as the user roams in order to minimize WiFi contention [32]. Clearly with future generation of communication networks and mobile devices, combination of the above approaches can be used to alleviate WiFi contention problem.

A major advantage of an image based localization approach, especially as it relates to augmented reality, is that it can recover both position and orientation of the user. Even though in this paper we primarily used images for positioning, the approach in [4, 5] is fully capable of recovering user's orientation with 10 degree accuracy 90% of the time. Future work consists of integrating orientation recovery into our system.

Pictures taken by AR glass or smartphone's low-cost cameras often suffer from blur due to user's unsteady hand movements and walking motion. In our case, since the exposure time could be set to a very short duration i.e. close to 1 millisecond, the major culprit of image blur is user's movement e.g. walking and involuntary shaking of the user's hand. Blur could potentially cause some of the otherwise useful images unusable. We plan to deal with blur by estimating the user motion and deblurring images using point spread function obtained from motion data. The deblurred image could then be fed into image localization algorithm for feature extraction. Thus, the system can become more useful in situations when the user walks fast.

We might also be able to achieve better performance on AR glass by updating to the latest version of Vuzix M100 or Google Glass. Better hardware especially WiFi reception capability and 5GHz band support is essential for boosting WiFi localization accuracy.

REFERENCES

[1] H. Lin, Y. Zhang, M. Griss, and I. Landa, "WASP: An enhanced indoor locationing algorithm for a congested wifi environment," *Mobile Entity Localization and Tracking in GPS-less Environments*, vol. 5801, pp. 183–196, 2009.

[2] O. J. Woodman, "An introduction to inertial navigation," Tech. Rep. UCAM-CL-TR-696, University of Cambridge, Computer Laboratory, Aug. 2007.

[3] F. Li, C. Zhao, G. Ding, C. Liu, and F. Zhao, "A reliable and accurate indoor localization method using phone inertial sensors," in *UbiComp '12, Pittsburgh, PA, USA*. IEEE, 2012, pp. 421–430.

[4] J. Z. Liang, N. Corso, E. Turner, and A. Zakhor, "Image based localization in indoor environments," *International Conference on Computing for Geospatial Research and Application (COM.Geo) '13, San Jose, CA, USA*, 2013.

[5] J. Z. Liang, N. Corso, E. Turner, and A. Zakhor, "Reduced-complexity data acquisition system for image-based localization in indoor environments," *International Conference on Indoor Positioning and Indoor Navigation (IPIN '13), Montbeliard-Belfort, France*, 2013.

[6] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, pp. 91–110, 2004.

[7] T. C. Hales, "The jordan curve theorem, formally and informally," *The Mathematical Association of America Monthly*, vol. 114, 2007.

[8] N. Corso and A. Zakhor, "Indoor localization algorithms for an ambulatory human operated 3d mobile mapping system," *Remote Sensing*, vol. 5, pp. 6611–6646, 2013.

[9] E. Turner and A. Zakhor, "Floor plan generation and room labeling of indoor environments from laser range data," *GRAPP '14, Berlin, Germany*, vol. 5, 2014.

[10] F. Evennou and F. Marx, "Advanced integration of wifi and inertial navigation systems for indoor mobile positioning," *EURASIP Journal on Applied Signal Processing*, vol. 2006, pp. 1–11, 2006.

[11] N. Naikal, J. Kua, and A. Zakhor, "Image augmented laser scan matching for indoor dead reckoning," *International Conference on Intelligent RObots and Systems (IROS), St. Louis, MO, USA*, 2009.

[12] G. Chen, J. Kua, S. Shum, N. Naikal, M. Carlberg, and A. Zakhor, "Indoor localization algorithms for a human-operated backpack system," *3D Data Processing, Visualization, and Transmission, Paris, France*, 2010.

[13] T. Liu, M. Carlberg, G. Chen, J. Chen, J. Kua, and A. Zakhor, "Indoor localization and visualization using a human-operated backpack system," *International Conference on Indoor Positioning and Indoor Navigation (IPIN), Busan, Korea*, 2010.

[14] J. Kua, N. Corso, and A. Zakhor, "Automatic loop closure detection using multiple cameras for 3d indoor localization," *IS&T/SPIE Electronic Imaging*, 2012.

[15] P. Levchev, C. Yu, M. Krishnan, and A. Zakhor, "Indoor wifi localization with a dense fingerprint model," *Submitted to Global Communications Conference (GlobeCom) '14, Austin, TX*, 2014.

[16] A. Hallquist and A. Zakhor, "Single view pose estimation of mobile devices in urban environments," *IEEE Workshop on the Applications of Computer Vision (WACV), Clearwater, FL, USA*, 2013.

[17] J. Zhang, A. Hallquist, E. Liang, and A. Zakhor, "Location-based image retrieval for urban environments," *International Conference on Image Processing (ICIP), Brussels, Belgium*, 2011.

[18] J. Z. Liang, N. Corso, E. Turner, and A. Zakhor, "Image-based positioning of mobile devices in indoor environments," in *Multimodal Location Estimation*, Sprinter Gerald Friedland, Ed., 2014.

[19] K. Chen and K. R. Vadde, "Design and evaluation of an indoor positioning system framework," *U.C. Berkeley course project for CS262A*, 2012.

[20] J. T. Quah and L. Lim, "Location cluster with nearest neighbors in signal space: An implementation in mobile service discovery and track-

Table 1: Comparison of paths obtained via different tracking methods.

	Number of Jumps	Total Jump Length(m)	Number of Backward Jumps	Total Path Length(m)
WiFi only (WiFi)	11	72.8	2	189.4
WiFi only (4G)	7	50.8	2	184.8
Vuzix	11	122.0	1	243.0
Image only	11	121.9	1	207.1
Image + WiFi	24	173.6	4	252.9

ing," *International Conference on Wireless Networks (ICWN '11), Las Vegas, NV, USA*, 2011.

[21] J. Biswas and M. Veloso, "Wifi localization and navigation for autonomous indoor mobile robots," *IEEE International Conference on Robotics and Automation (ICRA), Anchorage, AK, USA*, 2010.

[22] D. M Lambeth, "Design considerations for an indoor location service using 802.11 wireless signal strength," M.S. thesis, Department of Aeronautics and Astronautics, MIT, 2009.

[23] C. Yoon J. So, J. Lee and H. Park, "An improved location estimation method for wifi fingerprint-based indoor localization," *International Journal of Software Engineering and Its Applications*, 2013.

[24] N. Kothari, B. Kannan, and M. B. Dias, "Robust indoor localization on a commercial smart-phone," Tech. Rep. CMU-RI-TR-11-27, The Robotics Institute, Carnegie-Mellon University, Aug. 2011.

[25] M. Holcik, "Indoor navigation for android," *Masters thesis, Faculty of Informatics, Masaryk University*, 2012.

[26] Y. Jin, H. Toh, W. Soh, and W. Wong, "A robust dead-reckoning pedestrian tracking system with low cost sensors," *IEEE International Conference on Pervasive Computing and Communications (PerCom), Seattle, WA, USA*, 2011.

[27] A. Serra, T. Dessi, D. Carboni, V. Popescu, and L. Atzori, "Inertial navigation systems for user-centric indoor applications," *Networked & Electronic Media (NEM '10) Summit, Barcelona, Spain*, 2010.

[28] H. Wang, H. Lenz, A. Szabo, J. Bamberger, and U. D. Hanebeck, "WLAN-based pedestrian tracking using particle filters and low-cost MEMS sensors," *Positioning, Navigation and Communication, 2007. WPNC '07. 4th Workshop on, Hannover, Germany*, 2007.

[29] Widyawan, M. Klepal, and S. Beauregard, "A backtracking particle filter for fusing building plans with PDR displacement estimates," *Positioning, Navigation and Communication, 2008. WPNC '08. 5th Workshop on, Hannover, Germany*, 2008.

[30] V. Renaudin, M. Susi, and G. Lachapelle, "Step length estimation using handheld inertial sensors," *Sensors Open-Access Journal*, 2012.

[31] F. H. T. Pinto, "An indoor localization solution for mobile devices," M.S. thesis, FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO, 2011.

[32] Gabriel Takacs, Yingen Xiong, Radek Grzeszczuk, Vijay Chandrasekhar, Wei chao Chen, Kari Pulli, Natasha Gelfand, Thanos Bismpigiannis, and Bernd Girod, "Outdoors augmented reality on mobile phone using loxel-based visual feature organization," in *In Proceeding of ACM international conference on Multimedia Information Retrieval*, 2008, pp. 427–434.