2022 IEEE International Conference on Robotics and Automation (ICRA)
May 23-27, 2022. Philadelphia, PA, USA

Single User WiFi Structure from Motion in the Wild

Yiming Qian[1] Hang Yan[2] Sachini Herath[3] Pyojin Kim[4] Yasutaka Furukawa[3]

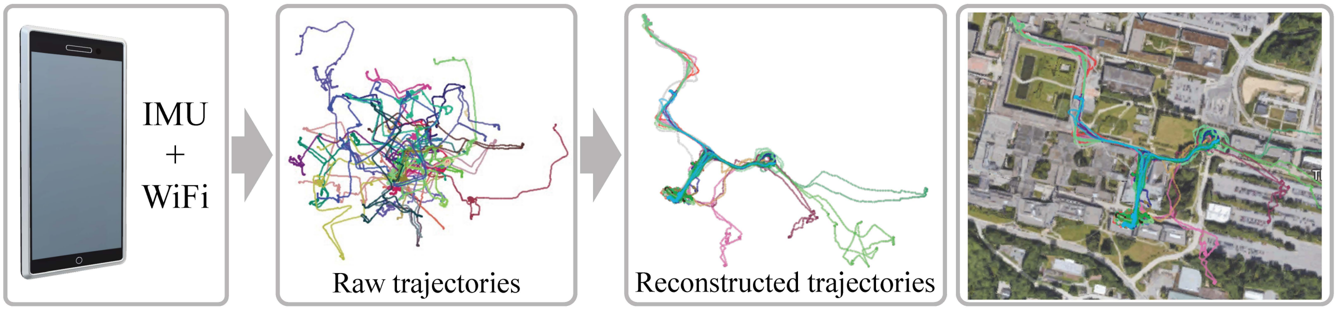

Fig. 1. This paper proposes a novel motion estimation algorithm, dubbed WiFi Structure-from-Motion. The approach takes WiFi and IMU sensor data from a smartphone during normal day-to-day activities, estimates raw trajectories by inertial navigation, and reconstructs the final trajectories and a radio fingerprint map for indoor positioning. At the right, we manually overlay the reconstruction with an aerial view of the environment (Google Maps).

***Abstract*— This paper proposes a novel motion estimation algorithm using WiFi networks and IMU sensor data in large uncontrolled environments, dubbed "WiFi Structure-from-Motion" (WiFi SfM). Given smartphone sensor data through day-to-day activities from a single user over a month, our WiFi SfM algorithm estimates smartphone motion trajectories and the structure of the environment represented as a WiFi radio map. The approach 1) establishes frame-to-frame correspondences based on WiFi fingerprints while exploiting our repetitive behavior patterns; 2) aligns trajectories via bundle adjustment; and 3) trains a self-supervised neural network to extract further motion constraints. We have collected 235 hours of smartphone data, spanning 38 days of daily activities in a university campus. Our experiments demonstrate the effectiveness of our approach over the competing methods with qualitative evaluations of the estimated motions and quantitative evaluations of indoor localization accuracy based on the reconstructed WiFi radio map. The WiFi SfM technology will potentially allow digital mapping companies to build better radio maps automatically by asking users to share WiFi/IMU sensor data in their daily activities.**

I. Introduction

The Internet-of-Things (IoT) devices like smartphones are ubiquitous in everyday life, providing opportunities to track motions and localize the position of an individual from low-energy anytime anywhere sensors. With the wide coverage of wireless networks in human-centric environments, WiFi becomes a reliable cue for localization thanks to its low energy cost and flexibility of working anytime anywhere. Effective positioning systems benefit numerous location-aware applications spanning indoor GPS, augmented reality, robotics, and contact-tracing.

However, indoor positioning systems typically require laborious site-surveys. Early methods construct the database of the geo-coordinates of WiFi access points (AP), while using trilateration for positioning [1], [2]. Recent attempts bypass site surveying under the framework of WiFi SLAM, whereas creating high-quality radio maps typically requires a single continuous data acquisition and is limited to small/medium-scale environments [3], [4]. Crowdsourcing [5], [6] has been an efficient solution for large-scale scenes but still imposes challenges to the data acquisition process (*e.g.*, all users have to start from the same position and the smartphone is always held in front of users [7]).

This paper takes WiFi positioning systems to the next level, turning WiFi/IMU sensor data from normal day-to-day activities into a WiFi radio map. Our goal is to estimate a radio fingerprint map, namely geo-coordinates and received signal strength indicators (RSSI) from *unstructured* and *distributed* motion recordings of a single user. The constructed WiFi maps will enable the positioning of other users.

Concretely, our system (1) uses an inertial navigation algorithm (RoNIN [8]) to estimate the relative motion trajectory from IMU recordings for each day, (2) establishes trajectory correspondences using WiFi RSSIs as features, (3) performs bundle adjustment (BA) to correct each trajectory and aligns all trajectories into the same coordinate system, and (4) trains a self-supervised neural network to further extract motion constraints for BA. We have developed an Android app to collect WiFi and IMU measurements of a single individual all-day in a university campus for 38 days, spanning 235 hours and 45 kilometers. The quantitative and qualitative evaluations on the WiFi-based localization task show that our method is able to produce 43% more accurate than the state-of-the-art system [7].

[1] University of Manitoba. yiming.qian@umanitoba.ca
[2] Washington University in St. Louis. yanhangxiong@gmail.com
[3] Simon Fraser University. {sherath, furukawa}@sfu.ca
[4] Sookmyung Women's University. pjinkim@sookmyung.ac.kr

This work was done when Y. Qian and P. Kim were a postdoc at SFU and when H. Yan was with WUSTL. This research is partially supported by NSERC Discovery Grants, NSERC Discovery Grants Accelerator Supplements, and DND/NSERC Discovery Grant Supplement.

978-1-7281-9680-0/22/$31.00 ©2022 IEEE

Note that our system does not use any external geo-localization information such as satellite GPS or commercial localization APIs (e.g., Google Fused Location Provider [9]). Rather, this work is to build a radio map to enable such geo-localization techniques, crucial in the emerging markets.

II. Related Work

A. Structure from motion

Visual SLAM [10] has flourished in the past decade with the emergence of real-time products such as Google Project Tango and Apple ARKit. These systems require *continuous* video capturing by constantly holding a smartphone. Visual SfM is a relevant technique and works for *unstructured* image collections [11]. For example, the state-of-the-art method is capable of recovering an entire 3D city (*e.g.*, Rome) from millions of unstructured images parsed from Internet, thanks to the advances in image matching and 3D reconstruction [12]. This paper extends such success to WiFi SfM, using mobile WiFi and IMU sensors that are energy-efficient and work anywhere anytime as opposed to a camera.

B. Localization

Satellite GPS has been successful for outdoor localization but does not work in indoor environments. Researchers have explored alternative cues such as vision, LiDAR, compass, WiFi, and Bluetooth [13]. Geo-localization is typically achieved by matching a query signal against a database of geo-localized signals. Google Fused Location provider (FLP) is a widely-used commercial API utilizing multi-modal sensor data [9].

With the rapid growth of wireless networks in the early 21st century, WiFi positioning systems become the most prominent methods given the superiority of reliability, ubiquity, and flexibility, which can be categorized into two major groups: signal-strength-based and fingerprinting-based.

Signal-strength-based methods assume that pairs of a MAC address and a geo-coordinate are available for WiFi access points [14], typically constructed by dedicated site surveys. Localization is performed by trilateration [1], [2], which, unfortunately, is not robust because of the multipath fading issue.

Fingerprinting-based systems reply on a radio fingerprint map, which maps a RSSI vector into a geo-coordinate. The map can be constructed by an on-site survey or techniques such as visual odometry and WiFi SLAM [3], [4], [15]. For localization, analytic RSSI-to-distance models (*e.g.*, Gaussian process [3], [15]) or fingerprint-based location lookups [16], [17] are employed. WiFi SLAM requires a single continuous data sequence, whereas our WiFi SfM enables radio map creation from distributed and unstructured motion data.

C. Mobile crowdsensing

Crowdsensing via ubiquitous smartphones with rich built-in sensors creates a new powerful paradigm for large-scale distributed data collection for various sensing tasks, including navigation [8], [18], indoor localization [6], [19], and floorplan reconstruction [20], [21]. In fact, crowdsencing has been introduced to help the construction of AP database [22] or radio fingerprint map in larger environments [7], [23], [24]. However, to the best of our knowledge, no work has been tested with real data in the wild [7], [23], [24]. For example, a phone was carried by a hand, in a pocket, or in a bag, but ten users were simply asked to randomly walk inside one floor for an hour [24]. All our data come from a smartphone under standard daily activities, where one may read/write emails, browse websites, order foods, or simply carry in a pocket, which span more than 6 hours per day on the average over 38 days inside an entire university campus.

III. Problem definition and dataset

A. WiFi Structure-from-Motion

WiFi SfM is the task of taking WiFi and IMU sensor measurements over multiple trajectories, potentially across different dates by different users, then reconstructing their motions as well as a WiFi radio map in a single coordinate frame. This paper focuses on sensor data in the wild by a single individual, acquired through daily activities. Note that the task does not take any external geo-localization information such as satelite GPS, but rather reconstructs a radio map to enable such geo-localization services.

B. Dataset

We developed an Android app to record IMU and WiFi sensor data, which is adopted from [25] and collects accelerations (200Hz), angular velocities (200Hz), and device orientations (100Hz). The WiFi receiver records RSSIs and the MAC addresses at 1Hz.

We installed the data capture app to an android smartphone (i.e., Samsung Galaxy S9) of a single individual, where the individual turns on the app upon arriving the Simon Fraser university campus and turning off the app before leaving. The smartphone is handled by the individual naturally as a part of daily activities. Note that the data were captured before the COVID shutdown in the middle of a busy semester. The dataset spans 38 days with the average duration of 6.2 hours and the average travel distance of 1.2 kilometers per day, covering approximately 550m × 580m space.

IV. Algorithm

Our WiFi SfM algorithm draws an inspiration from visual SfM with similar system components as depicted in Figure 2. Our algorithm is also similar in spirit to WiFi SLAM, which 1) estimates a relative motion per trajectory based on pedestrian dead reckoning (PDR); 2) finds frame-to-frame correspondences based on the RSSI similarities; and 3) bundle-adjusts (BA) the trajectories based on the correspondence constraints [3], [4], [7].

There are two key distinctions in our approach. First, we exploit repetitive behavior patterns to obtain motion constraints. For example, we walk the same routes to the same office and use the same restroom everyday. Second, if the estimated motions were correct, a standard RSSI-based positioning should be accurate. We train a neural network

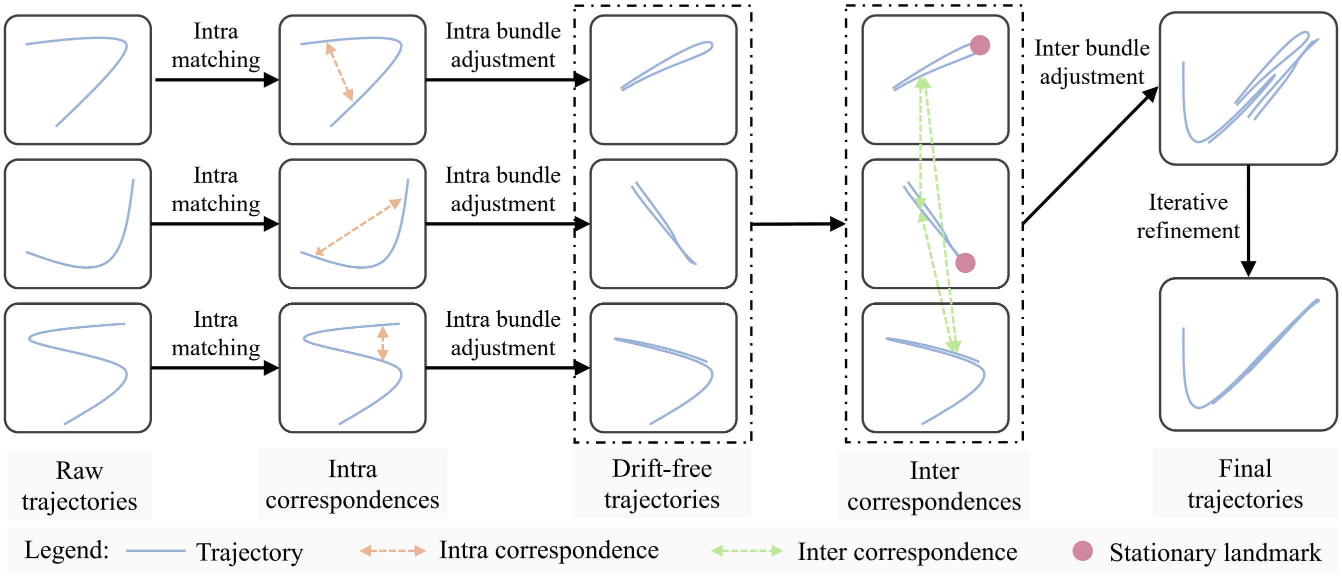

Fig. 2. System overview. Our approach turns IMU sensor data into relative motion trajectories by RoNIN [8]. We then use intra-trajectory correspondences for correcting drifting errors in each single trajectory, and use inter-trajectory correspondences and clustered stationary frames (i.e., stationary landmarks) to align trajectories into the same coordinate. The iterative refinement produces the final trajectories.

to perform RSSI-based positioning with self-supervision, then derive new constrains enforcing that the reconstructed motions are consistent with the RSSI-based positioning.

Concretely, given a sequence of IMU sensor data for each day, we first use a state-of-the-art neural inertial navigation algorithm RoNIN [8] to estimate a relative motion trajectory. Second, we establish correspondences based on WiFi signals for stationary (i.e., no motions) and non-stationary frames (Section IV-A). Given the correspondences, we perform intra- and inter-trajectory bundle adjustment (Sections IV-B and IV-C). Finally, an iterative refinement is performed by enforcing the consistency between the motion estimations and the RSSI-based positioning (Section IV-D). Below we present the details of each component.

A. WiFi-based correspondence matching

1) Stationary frames: Humans do not keep moving 24 hours a day, and tend to stay at certain landmarks, which could be a desk, a meeting room, or a favorite cafe. We detect stationary frames (*i.e.*, no motion) whose magnitudes of the angular velocity vectors from the Android API are below 0.1. This process turns a trajectory into intervals of stationary frames and non-stationary frames. We treat non-stationary intervals less than 3 minutes as stationary to filter out noise, then group/match stationary intervals across all the trajectories by mean-shift clustering (scikit-learn implementation), while treating the average RSSI vector within each interval as the feature.

This process matches stationary intervals as a landmark, which will be used as frame-to-frame correspondences. Since the stationary frames account for 87% of the data, we drop them and concatenate the remaining frames to shorten each trajectory for the subsequent pose estimation.

2) Non-stationary frames: Following a popular approach, we use the cosine similarity metric between WiFi RSSI vectors to establish frame-to-frame correspondences [7], [26]. Given a trajectory, for each frame, we find the other frame with the most similar RSSI vector that is more than F/10 frames away, where F is the total number of frames in the trajectory (i.e., intra-trajectory correspondence). Similarly, given a pair of trajectories, for each frame, we find the other frame in the other trajectory with the most similar RSSI vector (i.e., inter-trajectory correspondence). Unreliable correspondences are eliminated by the forward and backward consistency check, popular in the computer vision literature [27]: In order for frame f and g to be a match, f must be the closest to g, and vice versa.

B. Intra-trajectory bundle adjustment

Intra-trajectory correspondences enable non-linear least squares optimization to refine the shape of each trajectory, mitigating the rotational accumulation errors caused by gyroscope bias [8], [28], [29]. Specifically, RoNIN estimates the relative motion represented as a pair of velocity magnitude v^i_f and heading direction angle θ^i_f at frame f for the ith trajectory. We solve for the angle correction $\Delta\theta^i_f$:

$$\min_{\{\Delta\theta^i_f|1\le f\le F\}} \sum_{(f,t)\in\Phi^i} \rho\left(\|\mathbf{x}^i_f-\mathbf{x}^i_t\|^2\right)+\sum_{f=2}^{F-1}\rho\left((\Delta\theta^i_f-\Delta\theta^i_{f-1})^2\right)$$

$\mathbf{x}^i_f$ denotes the 2D position at frame f and is computed as $\mathbf{x}^i_f=\mathbf{x}^i_0+\sum_{t=1}^{f} v^i_t[\cos(\theta^i_t+\Delta\theta^i_t),\sin(\theta^i_t+\Delta\theta^i_t)]^\top$. $\mathbf{x}^i_0$ is the starting position and set to $[0,0]^\top$ in RoNIN. Φ^i denotes the intra-trajectory correspondences for trajectory i. [1] The

[1] We use non-stationary correspondences while excluding stationary correspondences here for simplicity, which did not change results.

first term ensures the spatial consistency between matched frames. The second term enforces the smoothness of angle displacements at adjacent frames. ρ is the Cauchy loss function, which is robust and there are no scalar weights for the two terms.

After optimization, we filter out unreliable trajectories whose spatial consistency error is above 8.0 or number of frame correspondences is below 13. In our experiments, 15 out of 38 trajectories are removed by this process.

C. Inter-trajectory bundle adjustment

Inter-trajectory correspondences enable non-linear least squares optimization to align all the trajectories into the same coordinate system by solving for the angle corrections and the starting points simultaneously. Compared to the bundle adjustment in IV-B, here the formulation has three terms: 1) The spatial consistency term in the same form, including both intra- and inter- trajectory correspondences; 2) Exactly the same smoothness term; and 3) The stationary consistency term $\sum_{\mathcal{C}\in\Omega}\sum_{(f,i)\in\mathcal{C}}\rho\left(\|\mathbf{x}^i_f-\bar{\mathbf{x}}_{\mathcal{C}}\|^2\right)$, enforcing that the stationary frames in the same cluster are close. Ω is the set of clusters and $\bar{\mathbf{x}}_{\mathcal{C}}$ denotes the average location of cluster $\mathcal{C}$. Again, we use the robust Cauchy loss and there are no scalar weights for the three terms.

D. Iterative refinement

The last step enforces the consistency between the reconstructed motions and the RSSI-based positioning results. Given a reference frame, we find top-10 closest frames based on the position distances, then use a weighted average of the positions of the neighbors to predict the location of the reference frame. If reconstructed motions were accurate, the prediction should be consistent with the location of the reference frame.

Concretely, the process iterates two steps: 1) Training a neural network, predicting the weights of the neighbors; and 2) Using the network to predict the locations and running the final bundle-adjustment to refine the motions while minimizing the inconsistencies between the predictions and the current motion reconstructions.

Weight network: We use a four layer multi-layer-perceptron (MLP) to predict the weights of the 10 neighbors with respect to a reference. A RSSI vector is represented as an N-dimensional vector, where N is the number of access points in a scene ($N = 535$ in our case). The input to the MLP is a $10\times 2N$ tensor, where each row is a concatenation of the RSSI vectors from a neighbor and the reference. Note that the reference vector is repeated across rows. The output is a 10-dimensional weight vector for the 10 neighbors. The two hidden layers has 2048 and 1024 nodes, respectively. All layers are followed by a ReLU function, except the output layer that is followed by a softmax function. The loss measures the consistency between the prediction and the location: $\|\mathbf{x}^i_f-\hat{\mathbf{x}}^i_f\|^2$, where $\hat{\mathbf{x}}^i_f=\sum_{s=1}^{10}w_s\mathbf{x}_s$ is the weighted average of $\mathbf{x}^i_f$. The PyTorch library is employed for neural network training with an Adam optimizer. The learning rate is 0.0001 and the batch size is 16.

TABLE I

QUANTITATIVE COMPARISONS AGAINST TWO STATE-OF-THE-ART METHODS AND ONE VARIANT. COLUMN "WIFI ONLY?" INDICATES IF A METHOD USES WIFI DATA ONLY FOR LOCALIZATION. THE LAST TWO COLUMNS SHOW THE LOCALIZATION ERRORS (MEAN AND MEDIAN) IN METERS. THE COLORS CYAN AND ORANGE REPRESENT THE BEST AND THE SECOND BEST.

Method	WiFi only?	Mean	Median
C-SLAM-RF [7]	✓	21.0	15.1
FLP [9]		8.8	7.2
Ours w/o refine	✓	18.3	13.0
Ours	✓	11.9	8.7

Final bundle adjustment: The same bundle adjustment process refines the angle displacement $\Delta\theta_f$ and the starting point $\mathbf{x}_0$ for each trajectory, subject to the same smoothness term and the above loss function:

$$\min_{\mathbf{x}_0,\{\Delta\theta_f|1\leq f\leq F\}}\sum_{f=1}^{F-1}\rho\left(\|\mathbf{x}_f-\hat{\mathbf{x}}_f\|^2\right)+\sum_{f=2}^{F-1}\rho\left((\Delta\theta_f-\Delta\theta_{f-1})^2\right)$$

We repeat the iteration 20 times to produce the final result.

V. EXPERIMENTAL RESULTS

We have implemented our approach in C++ and Python. For inertial navigation, the RoNIN ResNet model from the official website is used [8]. The Ceres library is used for the bundle adjustment steps [30]. The algorithm takes about 10 hours in total on a workstation with an Intel Core i7-7740X CPU and an NVIDIA GTX 1080Ti GPU with 11GB RAM.

A. Experimental setup

Test dataset: We use a Google Tango phone (Asus Zenfone AR) to collect data in the areas covered by our estimated radio map. We recorded WiFi RSSIs (1Hz), Google FLP results [9] (1Hz), and 6 DoF poses from visual SLAM (200Hz), which is used as ground-truth. Sensor measurements are asynchronous, and we use nearest neighbor interpolation to compute sensor measurements at arbitrary time-stamps. The test set consists of 22 raw trajectories, with an average duration of 2.7 minutes, which are all acquired independently from our dataset in Section III.

Evaluation metric: An ideal evaluation metric would be the accuracy of WiFi localization results based on a reconstructed radio map. However, our radio map is reconstructed up to a rigid transformation, which cannot be compared with the ground-truth. Instead of manually aligning our reconstruction to the map for ground-truth evaluation, we perform the following automatic process for the quantitative evaluation. Concretely, given a test sequence, we 1) use the nearest neighbor search with the radio map to localize each of the RSSI vectors, 2) align the results with the Tango trajectory by iterative closest point (ICP), and 3) calculate the average Euclidean distance between the corresponding locations.

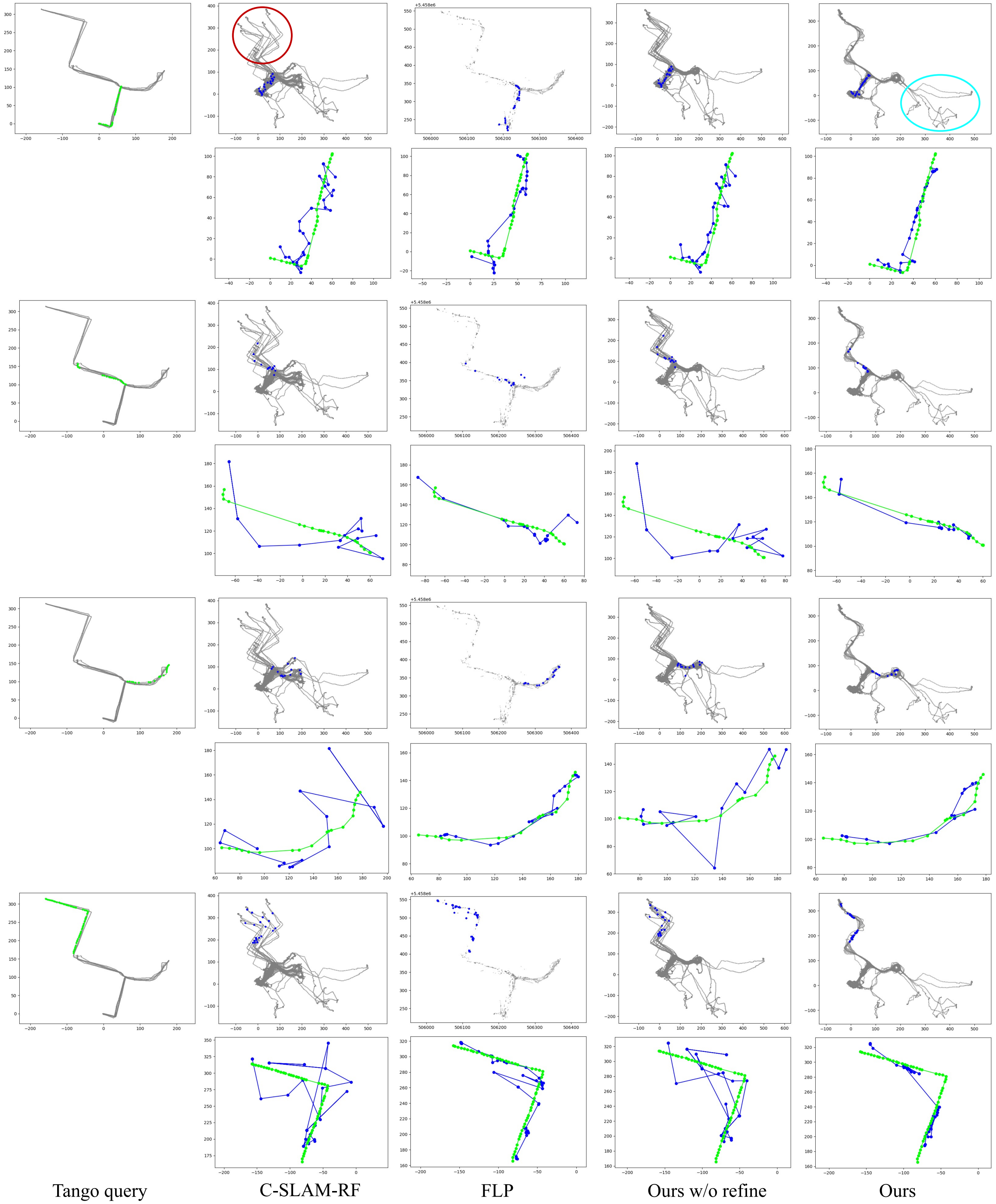

Fig. 3. Qualitative evaluations. The left column shows the four query sequences from a Tango device in green. The gray dots show all the query sequences. In the right four columns, gray dots show the reconstructed trajectories by C-SLAM-RF [7], FLP [9], and our system without or with the refinement step. Note that FLP shows the coordinates by the Google FLP API. In the top row, blue dots show the localization results, where RSSI-based nearest neighbor is used for C-SLAM-RF and our methods, while the result of Google API is simply shown for FLP. In the bottom row, green is again the query sequence from Tango, while blue shows the localization results that are aligned with the query via ICP. The red oval highlights large motion errors by C-SLAM-RF. The cyan oval highlights an area with large errors by our method, which is less frequently visited and lack enough constraints.

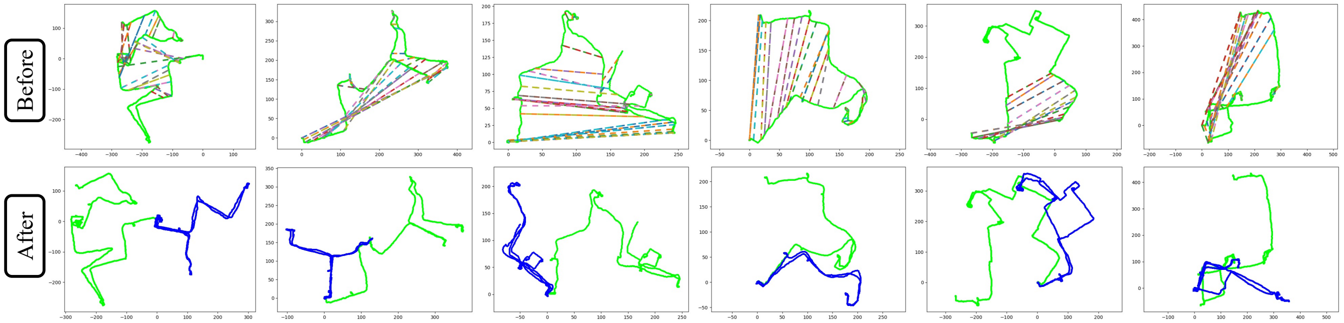

Fig. 4. Effectiveness of intra-trajectory bundle adjustment (Section IV-B). The top row shows the raw motion trajectories (green) by RoNIN [8], where the dashed lines show the intra-trajectory correspondences. The bottom row shows the drift-free trajectories (blue) after the correction.

B. *Competing methods*

We compare against two competing methods and one variant of our system.

- **C-SLAM-RF** [7] is a state-of-the-art WiFi SLAM system using WiFi-based loop closing. No implementation is available and we reproduced the system with two minor modifications: (1) Inertial navigation trajectories after the intra-trajectory bundle adjustment including the filtering (IV-B) are used as input, whereas the original version uses clean input trajectories without accumulation errors; (2) For fair comparison, the same Ceres is used as the optimizer.
- **Fused Location Provider (FLP)** [9] is a commercial geo-localization service by Google. We call the FLP API at 1Hz while collecting the test data.
- **Ours w/o refine** is a variant of our approach without the iterative refinement step.

C. *Quantitative evaluation*

Table I shows the main quantitative evaluations. C-SLAM-RF is the state-of-the-art system but has the largest localization errors (inferior to ours by 9.1 meters). In their experiments, all the trajectories started from the same location and the phone was always hand-held. Furthermore, we did not observe severe drifting errors in their data, possibly because the sensors were calibrated carefully before every data acquisition. Our experiments run on real data in the wild all day every day through normal day-to-day activities, posing real challenges. FLP achieved the best performance, which is understandable as FLP is a multi-modal system utilizing GPS, WiFi, IMU, and potentially rich digital mapping information (*e.g.*, a WiFi access point named Starbucks must be at the Starbucks store) with offline manual data cleaning. Our system is fully automatic and takes only a single RSSI vector for the localization task. Nonetheless, our approach achieves comparable errors (differs by 3.1 and 1.5 meters for mean and median) with a much simpler framework.

Our WiFi SfM system consists of three major steps: correspondence matching, bundle adjustment, and iterative refinement. While first two are inspired by visual SfM, iterative refinement is a new idea. To demonstrate the effectiveness of the refinement, we conduct an ablation study by evaluating our system without the refinement step. The last two rows in Table I show that the iterative refinement reduces the mean error by 6.4 meters (35%) and median error by 4.3 meters (33%), validating our contribution.

D. *Qualitative evaluation*

Figure 3 compares the reconstructed motion trajectories and the localization results. C-SLAM-RF suffers from erroneous alignment everywhere as highlighted by the red oval. The localization accuracy also degrades, as seen in the fluctuating blue curves in the second column. Our approach (the last column) significantly outperforms C-SLAM-RF and obtains smooth localization results, which align well with the Tango trajectory. Visually, FLP and our method are comparable, which is supported by the quantitative evaluations in Table I. The last two columns clearly show the effectiveness of our iterative refinement.

Figure 4 shows the reconstructed motions before and after the intra-trajectory bundle adjustment. In each of the six trajectories, the first and the last frames must be at the same location. However, severe rotational drifts are present, as is typical in inertial navigation with real data, which are all successfully corrected by the intra-trajectory bundle adjustment.

VI. Conclusion

This paper proposes a novel WiFi Structure from Motion system that reconstructs a radio map for an unknown environment by aligning unstructured motion trajectories across multiple days from a single user. Following the successful visual SfM pipeline, our method finds intra- and inter-trajectory correspondences using WiFi RSSIs and reconstructs motion trajectories by bundle adjustment. A novel iterative refinement scheme is also proposed for accuracy boost. Quantitative and qualitative evaluations demonstrate the effectiveness of our approach over the competing methods. WiFi SfM could allow us to construct the radio map for every single building in the world by mobile crowdsensing. Our future work includes more experimental validations with more data as well as true multi-user crowdsensing towards city-scale WiFi SfM, both of which are becoming possible as the pandemic is coming to an end.

References

[1] J. Yang and Y. Chen, "Indoor localization using improved rss-based lateration methods," in *GLOBECOM 2009-2009 IEEE Global Telecommunications Conference*. IEEE, 2009, pp. 1–6.

[2] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti, "Spotfi: Decimeter level localization using wifi," in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, 2015, pp. 269–282.

[3] B. Ferris, D. Fox, and N. D. Lawrence, "Wifi-slam using gaussian process latent variable models." in *IJCAI*, vol. 7, no. 1, 2007, pp. 2480–2485.

[4] J. Huang, D. Millman, M. Quigley, D. Stavens, S. Thrun, and A. Aggarwal, "Efficient, generalized indoor wifi graphslam," in *2011 IEEE international conference on robotics and automation*. IEEE, 2011, pp. 1038–1043.

[5] D. Sikeridis, B. P. Rimal, I. Papapanagiotou, and M. Devetsikiotis, "Unsupervised crowd-assisted learning enabling location-aware facilities," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4699–4713, 2018.

[6] C. Wu, Z. Yang, and Y. Liu, "Smartphones based crowdsourcing for indoor localization," *IEEE Transactions on Mobile Computing*, vol. 14, no. 2, pp. 444–457, 2014.

[7] R. Liu, S. H. Marakkalage, M. Padmal, T. Shaganan, C. Yuen, Y. L. Guan, and U.-X. Tan, "Collaborative slam based on wifi fingerprint similarity and motion information," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 1826–1840, 2019.

[8] S. Herath, H. Yan, and Y. Furukawa, "Ronin: Robust neural inertial navigation in the wild: Benchmark, evaluations, & new methods," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 3146–3152.

[9] "Fused location provider api." [Online]. Available: https://developers.google.com/location-context/fused-location-provider

[10] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[11] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: exploring photo collections in 3d," in *ACM siggraph 2006 papers*, 2006, pp. 835–846.

[12] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski, "Building rome in a day," *Communications of the ACM*, vol. 54, no. 10, pp. 105–112, 2011.

[13] F. Zafari, A. Gkelias, and K. K. Leung, "A survey of indoor localization systems and technologies," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2568–2599, 2019.

[14] "WiGLE - wireless geographic logging engine." [Online]. Available: https://wigle.net/

[15] H. Xiong and D. Tao, "A diversified generative latent variable model for wifi-slam," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, 2017.

[16] P. Bahl and V. N. Padmanabhan, "Radar: An in-building rf-based user location and tracking system," in *Proceedings IEEE INFOCOM 2000. Conference on computer communications. Nineteenth annual joint conference of the IEEE computer and communications societies (Cat. No. 00CH37064)*, vol. 2. Ieee, 2000, pp. 775–784.

[17] C. Wu, Z. Yang, Y. Liu, and W. Xi, "Will: Wireless indoor localization without site survey," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 4, pp. 839–848, 2012.

[18] R. Faragher, C. Sarno, and M. Newman, "Opportunistic radio slam for indoor navigation using smartphone sensors," in *Proceedings of the 2012 IEEE/ION Position, Location and Navigation Symposium*. IEEE, 2012, pp. 120–128.

[19] L. Chen, K. Yang, and X. Wang, "Robust cooperative wi-fi fingerprint-based indoor localization," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 1406–1417, 2016.

[20] R. Gao, M. Zhao, T. Ye, F. Ye, Y. Wang, K. Bian, T. Wang, and X. Li, "Jigsaw: Indoor floor plan reconstruction via mobile crowdsensing," in *Proceedings of the 20th annual international conference on Mobile computing and networking*, 2014, pp. 249–260.

[21] S. Chen, M. Li, K. Ren, and C. Qiao, "Crowd map: Accurate reconstruction of indoor floor plans from crowdsourced sensor-rich videos," in *2015 IEEE 35th International conference on distributed computing systems*. IEEE, 2015, pp. 1–10.

[22] Y. Zhuang, Z. Syed, J. Georgy, and N. El-Sheimy, "Autonomous smartphone-based wifi positioning system by using access points localization and crowdsourcing," *Pervasive and Mobile Computing*, vol. 18, pp. 118–136, 2015.

[23] C. Luo, H. Hong, and M. C. Chan, "Piloc: A self-calibrating participatory indoor localization system," in *IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*. IEEE, 2014, pp. 143–153.

[24] Z. Li, X. Zhao, Z. Zhao, and T. Braun, "Wifi-rita positioning: Enhanced crowdsourcing positioning based on massive noisy user traces," *IEEE transactions on wireless communications*, vol. 20, no. 6, pp. 3785–3799, 2021.

[25] S. Herath, S. Irandoust, B. Chen, Y. Qian, P. Kim, and Y. Furukawa, "Fusion-dhl: Wifi, imu, and floorplan fusion for dense history of locations in indoor environments," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021.

[26] P. Vorst, A. Koch, and A. Zell, "Efficient self-adjusting, similarity-based location fingerprinting with passive uhf rfid," in *2011 IEEE International Conference on RFID-Technologies and Applications*. IEEE, 2011, pp. 160–167.

[27] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International journal of computer vision*, vol. 47, no. 1, pp. 7–42, 2002.

[28] H. Yan, Q. Shan, and Y. Furukawa, "Ridi: Robust imu double integration," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 621–636.

[29] C. Chen, P. Zhao, C. X. Lu, W. Wang, A. Markham, and N. Trigoni, "Oxiod: The dataset for deep inertial odometry," *arXiv preprint arXiv:1809.07491*, 2018.

[30] S. Agarwal, K. Mierle, and Others, "Ceres solver," http://ceres-solver.org.