

Fusion-DHL: WiFi, IMU, and Floorplan Fusion for Dense History of Locations in Indoor Environments

Sachini Herath¹ Saghar Irandoost¹ Bowen Chen¹ Yiming Qian¹ Pyojin Kim² Yasutaka Furukawa¹

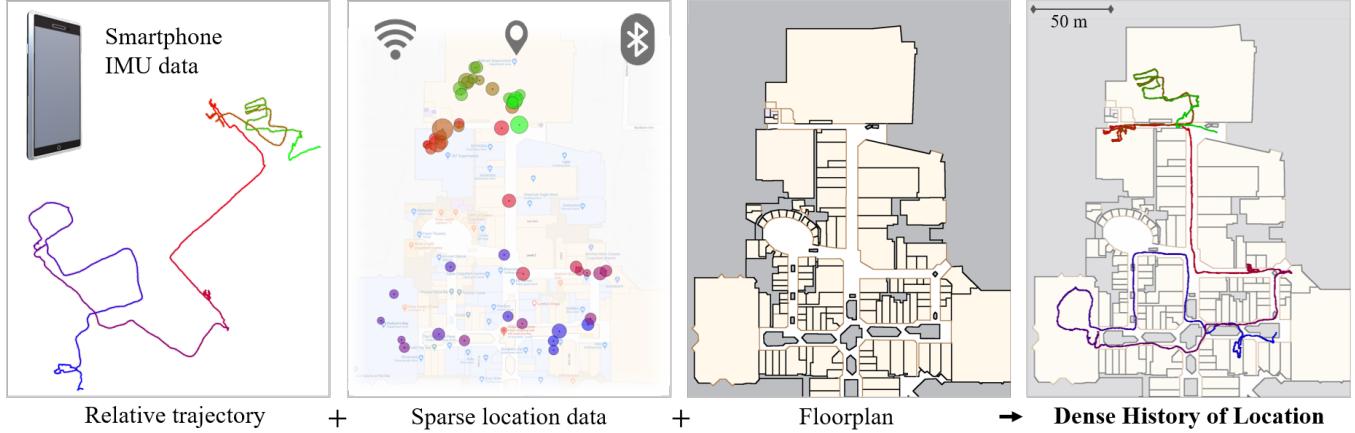


Fig. 1. The paper proposes a novel multi-modal sensor fusion algorithm that fuses 1) a relative motion trajectory by inertial navigation algorithm based on IMU sensor data; 2) sparse location data by geo-localization system based on WiFi; and 3) a floorplan image. The system works with any modern smartphone and is able to produce dense history of location (DHL) with minimal additional energy consumption on the smartphone. (The color of the trajectory and points, blue→red→green, encodes time.)

Abstract—The paper proposes a multi-modal sensor fusion algorithm that fuses WiFi, IMU, and floorplan information to infer an accurate and dense location history in indoor environments. The algorithm uses 1) an inertial navigation algorithm to estimate a relative motion trajectory from IMU sensor data; 2) a WiFi-based localization API in industry to obtain positional constraints and geo-localize the trajectory; and 3) a convolutional neural network to refine the location history to be consistent with the floorplan. We have developed a data acquisition app to build a new dataset with WiFi, IMU, and floorplan data with ground-truth positions at 4 university buildings and 3 shopping malls. Our qualitative and quantitative evaluations demonstrate that the proposed system is able to produce twice as accurate and a few orders of magnitude denser location history than the current standard, while requiring minimal additional energy consumption. We will publicly share our code and models.

I. INTRODUCTION

Indoor positioning has been a long-awaited technology, enabling mobile location-aware services, augmented reality applications, and location history collection. The location history is, in particular, getting more attentions for contact-tracing upon COVID pandemic. The location history further enables space utilization studies for indoor space planning or traffic flow analysis for transportation optimization.

¹ Simon Fraser University. {sherath, sirandou, bowenc, yimingq, furukawa}@sfu.ca

² Sookmyung Women's University. pjinkim@sookmyung.ac.kr

This research is partially supported by NSERC Discovery Grants, NSERC Discovery Grants Accelerator Supplements, and DND/NSERC Discovery Grant Supplement.

Contemporary indoor positioning systems utilize multi-modal sensor data such as global positioning system (GPS), inertial measurement unit (IMU), and building dependent infrastructures such as WiFi, Bluetooth or magnetic fingerprints. WiFi map is the most informative, which holds pairs of access-point IDs (i.e., MAC address) and their global positions (i.e., latitude/longitude degrees). The quality of WiFi-maps directly influence the accuracy of indoor positioning systems, where the community [1] and digital mapping companies (e.g., Google Maps and Apple Maps) are competing to build better WiFi maps.

This paper takes the indoor positioning technology to the next level, specifically focusing on the location history estimation as an offline task. Our idea is the novel fusion of WiFi, IMU, and floorplan data. Concretely, we use 1) a state-of-the-art inertial navigation algorithm (RoNIN [2]) to estimate a relative motion trajectory from IMU; 2) Google Fused Location Provider API, which mostly relies on WiFi, to obtain sparse global positions and align the inertial trajectory to a map; and 3) a convolutional neural network to refine the location history to be consistent with a floorplan. Our system works with any modern smartphone, while requiring minimal extra energy consumption on the device, where energy-efficient IMU is the only additional data acquisition.

We have built a new benchmark with WiFi, IMU, and floorplan data at 4 university buildings (training sites) and 3 shopping malls (testing sites), spanning 15.2 hours and 42.3 kilometers. Dense ground-truth positions are obtained by manually geo-localizing relative motion trajectories from a

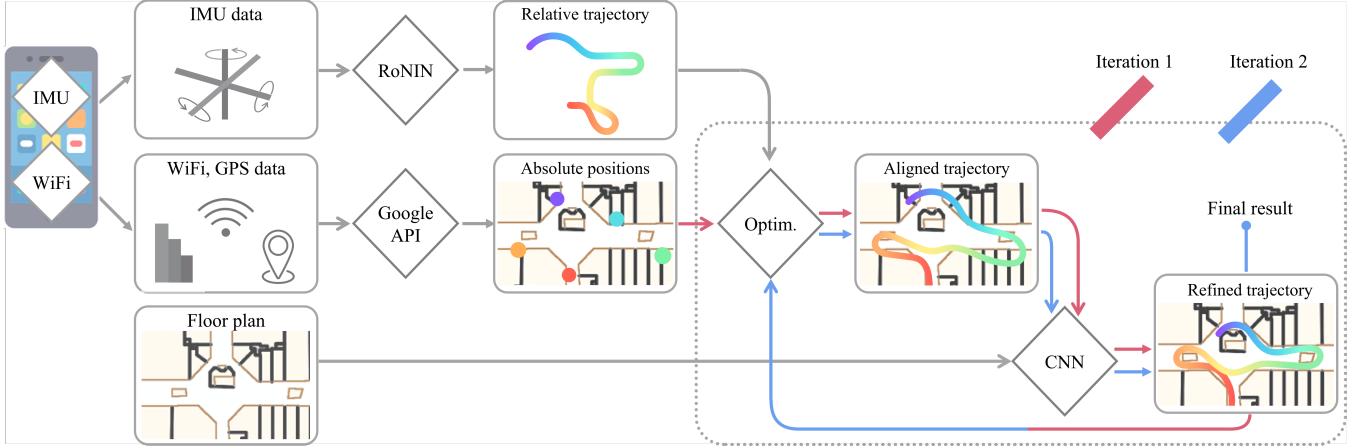


Fig. 2. System pipeline: We use a data-driven inertial algorithm to estimate relative motion trajectories from IMU sensor data. We geo-localize the trajectory with a floorplan by solving an optimization problem subject to the constraints from the state-of-the-art Indoor GPS system (i.e., Google Fused Location Provider). We use a convolutional neural network to refine the trajectory (time coded by color, blue→red) to be consistent with a floorplan. The system repeats the optimization and the CNN application once more to further improve the accuracy of the location history.

visual inertial SLAM algorithm. For long sequences through shopping malls in the wild (i.e., as long as 45 minutes), the ground-truth positions are obtained for a sparse set of points by manual specifications. Our qualitative and quantitative evaluations demonstrate that the proposed system is able to produce twice as accurate (i.e., RMSE roughly 5m instead of 12m) and a few orders of magnitude denser (i.e., 50Hz instead of 1/60Hz) location history than the current state-of-the-art system.

II. RELATED WORK

We discuss related works in three areas: inertial navigation, indoor localization via wireless networks, and indoor localization via multi-modal sensor fusion.

Inertial navigation: IMU is energy efficient and works anytime anywhere. Inertial navigation has been a dream technology for many years. Classical algorithms (also known as Pedestrian Dead Reckoning) are either physics based (e.g., IMU double integration), or heuristic based (e.g., zero speed update [3] or step counting [4]). With the surge of deep learning, data-driven approaches [2], [5], [6] have made a breakthrough. A state-of-the-art system achieves the positional accuracy of 5 meters per minute under natural activities in the wild [2]. The major error source is the rotational drift due to gyroscope bias.

Indoor localization via wireless networks: Wireless communication networks (e.g., WiFi [7] or Bluetooth [8]) are the most reliable information for indoor localization. A system requires either 1) an access-point database (i.e., access-point IDs and their geo-coordinates), which allows trilateration or weighted averaging of access-point coordinates [1]; or 2) a fingerprint database (i.e., received signal strength indicators and geo-coordinates at their observations), which allows fingerprint-based geo-coordinate look-ups [9], [10]. Google Fused Location Provider API (FLP) [11] is a widely used system, which we believe is mostly WiFi-based. However,

the system is energy-draining for a dense location history and often exhibits positional errors of more than 10 meters.

Indoor localization via Multi-modal sensor fusion: Modern smartphones are equipped with a suite of sensors. WiFi-SLAM [12]–[15] minimizes accumulation errors in inertial navigation by WiFi-based loop-closing constraints. Magnetic field distortions are specific to building infrastructures and provide additional cues for the fingerprint-based system [16]. Map-Matching algorithms align motion trajectories with a floorplan via conditional random fields [17], hidden Markov models [18], [19], particle filtering [20], or dynamic programming [17]. The paper proposes a novel fusion of WiFi, IMU, and floorplan data by a combination of optimization and a convolutional neural network (CNN).

III. SYSTEM: DENSE HISTORY OF LOCATION INFERENCE

Our system, coined DHL, works with any modern smartphone, utilizing an IMU and a WiFi receiver (See Fig. 2). The input to the system is IMU sensor data, global positions by a geo-localization API such as Google FLP at 1/60Hz, and a geo-localized floorplan. This section provides the system overview, which consists of the following steps.

- A data-driven inertial navigation algorithm (RoNIN [2]) estimates a relative motion trajectory at 50Hz from IMU.
- A non-linear least squares optimization is used to geo-localize the trajectory subject to positional constraints by FLP and regularization terms penalizing the deviations from the original trajectory with a sensor-centric error model. Besides the geo-localization, this alignment removes scale and/or rotational drifts inherent to the inertial navigation.
- A convolutional neural network (CNN) is trained to estimate a correction to the aligned trajectory for being consistent with a floorplan.
- Lastly, the system runs the optimization and the CNN steps once more, while changing the positional constraints from FLP to the current location history estimation, subsampled once every 200 frames.

IV. DENSE LOCATION HISTORY DATASET

The first contribution of the paper is a novel multi-modal sensor dataset with ground-truth positions. Our dataset contains 9.6 hours (26.1 km) of motion data over 4 university buildings (6 floorplans) and 5.7 hours (16.2 km) of motion data over 3 shopping malls (6 floorplans). Shopping malls are the indoor environments with immense business opportunities and our interests, which are kept for testing. University data are used for training a CNN module, where a single trajectory spans 5-10 minutes. Mall data are much longer (i.e., 15-45 minutes) and hence more challenging.

IMU data consists of accelerations (200Hz), angular velocities (200Hz), and device orientations (100Hz). A subject walks on a single floor of a building, while carrying a smartphone naturally by a hand or in a pocket (either Samsung Galaxy S9 or Google Pixel 2 XL). We have not performed gyro-calibration (e.g., placing on a table for 10 seconds) in advance to simulate the real-world scenarios, which causes severe “bending” due to rotational accumulation errors.

WiFi-based geo-positions are collected by the Google Fused Location Provider (FLP). We call the FLP API at 1Hz to obtain geo-positions as much as possible for evaluation. This rate is impractical for real-world applications as the API drains battery. We subsample FLP data at a realistic rate (1/60Hz) as an input to a system.

Floorplan images are created based on blueprints for university buildings and Google maps for shopping malls (See Fig. 1). Corridors, rooms, and unwalkable areas are in white, yellow, and grey, respectively. Open boundaries (e.g., walls with doors and storefronts) are in brown. The other walls are in black. University and shopping-mall floorplans are scaled so that 5 and 2.5 pixels correspond to a meter, respectively. We align the floorplans with the global coordinates by manually specifying correspondences between the floorplans and the Google Maps.

Ground-truth (GT) positions for shopping malls are obtained as either a sparse set of positions from real-time user clicks or dense trajectories from visual inertial SLAM. First, a set of ground-truth positions are obtained through an Android app where a subject clicks a floorplan on a smartphone real-time during the data acquisition. At total, 206 GT positions are collected for 5.0 hours of data from the three buildings, where a single trajectory spans 27.3 minutes on the average. Second, in one of the floorplans, we use visual inertial SLAM of Asus Zenfone (Google Tango phone) to collect relative trajectories, which are geo-localized by a rigid transformation with manual offline specifications to generate dense GT data. Due to the memory limitation on the visual SLAM system, the trajectories with dense GT positions are much shorter and hence less challenging, where a single trajectory spans 4.2 minutes on the average.

University data (9.6 hours) are used only for training a CNN module, where dense GT positions are obtained by geo-localizing inertial navigation trajectories with real-time user clicks by an optimization algorithm in our system (See Sect.V-A for details). Note that GT positions may not be

accurate for frames away from the user clicks. However, the university data are used only for training, and do not affect our evaluation. In practice, this training data allows the CNN model to generalize reasonable well to shopping malls.

V. ALGORITHM: WiFi, IMU, AND FLOORPLAN FUSION

The section explains the details of the non-linear least squares optimization, which fuses time-synchronized WiFi and IMU data, and the convolutional neural network, which fuses floorplan information for refinement.

A. IMU and WiFi Fusion by Optimization

A relative motion trajectory by RoNIN is a sequence of speed s_f and motion direction angle θ_f at every frame f . The position of the trajectory at frame f is calculated as $\sum_{i=1}^f s_i [\cos \theta_i, \sin \theta_i]^T$, assuming that motions are 2D on a single floor and start from the origin. The units of the positional and angular quantities are meters and radians in our formulation, respectively.

Inertial navigation exhibits two types of errors: 1) accumulation errors in the device orientation due to gyroscope bias; and 2) scale inference errors (i.e., the walking speed varies per person). We solve for the scale correction Δs_f , the angle correction $\Delta \theta_f$, and the positional displacement at the first frame \vec{P}_0 , where the corrected position is calculated as

$$\vec{P}_f = \vec{P}_0 + \sum_{i=1}^f s_i \cdot \Delta s_i [\cos(\theta_i + \Delta \theta_i), \sin(\theta_i + \Delta \theta_i)]^T.$$

Note that the scale (resp. angle) correction terms should be smooth, and we define a correction term every 100 (resp. 20) seconds, while using linear interpolation to derive the amount of correction at an arbitrary frame [5]. In reality, angle corrections should be on the device axes, but we make a simplified assumption and directly model corrections in the heading angle around the gravitational axis.

Concretely, we formulate a non-linear least squares optimization problem subject to the positional constraints from FLP and the regularization constraints ($w_1 = 10, w_2 = 200$):

$$\begin{aligned} \sum_{f \in \mathcal{F}_{FLP}} \max(\|\vec{P}_f - \vec{P}_f^{FLP}\|^2 - r_f^{FLP}, 0) + w_1 E_1^{REG} (\{\Delta s_f\}) \\ + w_2 E_2^{REG} (\{\Delta \theta_f\}) \end{aligned}$$

The first term is the sum of squared distances between the corrected position (\vec{P}_f) and the geo-position by FLP. We only penalize if the error is larger than accuracy reported by FLP, r_f^{FLP} . \mathcal{F}_{FLP} denotes the set of frames with FLP constraints. The second term is the regularization over the scale corrections, where Δs_f should be close to 1.0. The term is $\max(\Delta s_f, \frac{1}{\Delta s_f})^2$, summed over all the variables. We also set $s_f \geq 0$ as constraints. The last term is the regularization over the angle corrections. We simply use squared magnitudes of the first order and the second order derivatives via finite differences over the variables. The second order magnitude is further amplified by a factor of 1.5.

Before the optimization, we solve for a rigid transformation by linear least squares that aligns the FLP positions and

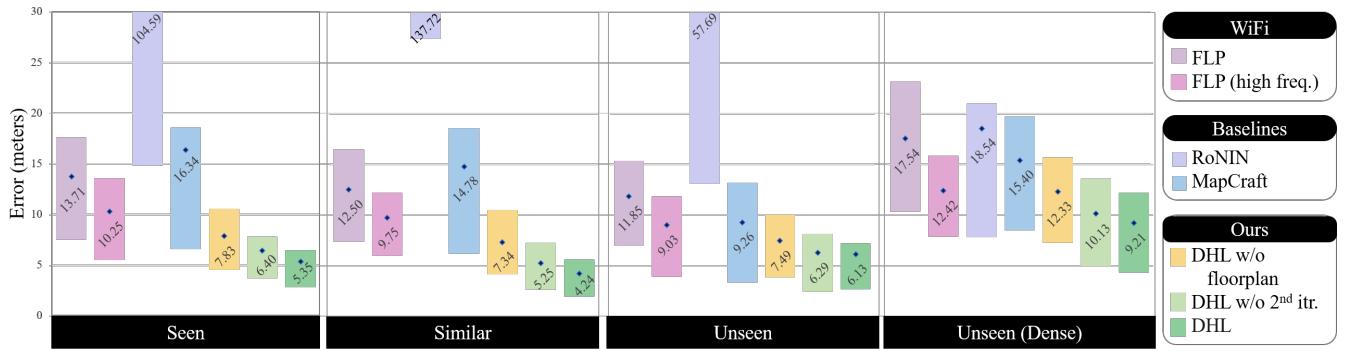


Fig. 3. Quantitative evaluation: Localization accuracy for 4 groups of test data (data groups described in VI-A). We compare the proposed approach (DHL) against four competing methods and two variants of our system. Each bar shows the mean and the first/third quartiles of the localization errors.

the corresponding frames in the inertial navigation trajectory. The rigid transformation is used to initialize the angle correction and the positional displacement variables, while keeping the scale correction variables to be 1.0.

B. Floorplan fusion by CNN

FLP positional errors are often 10 meters or more. We refine the aligned trajectory based on the floorplan via a convolutional neural network (CNN). The section explains the data representation, the network architecture, and the pre-processing steps for training and testing.

Data representation: We represent each trajectory segment as an image by scatter-plotting the curve in the black background with disks of 3 pixel-radius and with a rainbow colormap representing time (see Fig. 1). Being concatenated with a RGB floorplan image, the input is a 6-channel image of size 250×250 . The output is a “correction flow”, which is a 2D positional displacement vector, only defined for each pixel along the input trajectory segment (other pixel values are disregarded when computing loss and generating the corrected trajectory). The correction flow is represented as a 2-channel image of size 250×250 , where the numeric values are in the unit of a pixel.

Network architecture: We use U-Net [21], in particular, an implementation for brain image segmentation [22] with one modification. We added a bilinear layer to our input image, which increase the number of channels from 6 to 18. We use the Smooth L1 Loss with a default PyTorch implementation at every pixel along the input segment, where no loss is defined at the remaining pixels.

Training data pre-processing: Our training data come from university buildings, while testing data come from shopping malls. Both have similar error characteristics on the inertial trajectories, but FLP behaves differently. We use university data to generate training samples that generalize well to shopping mall data by simulating FLP noise.¹

To study the generalization capability of the system over different building types (i.e., from university buildings to shopping malls), we also generate synthetic training data by drawing trajectories on one mall floorplan by a Python GUI. Specifically, we use cubic splines to generate a continuous

and constant-velocity trajectory, which is treated as the ground-truth. The input “corrupted” trajectories are generated in the same way as before, after adding synthetic bias/scale errors. In total, we have generated 4,556 training samples from university buildings and 825 samples from shopping malls (synthetic).

Testing data pre-processing: An inertial trajectory can be as long as 45 minutes. We divide into overlapping shorter segments by 1) either taking the first 4 minutes or a segment whose axis aligned bounding box does not fit inside 250×250 pixel square of the floorplan, whichever comes first; and 2) repeat the process after removing the three quarters of the segment to create overlaps. For each segment, we crop a 250×250 pixel region of a floorplan so that the bounding box of the trajectory is centered. Pixels in the overlapping regions have multiple predictions from the CNN model, where we take their weighted average. Suppose a trajectory segment spans a time interval of $[0, T]$. The weight of a pixel is given by a normal distribution $\mathcal{N}(T/2, T/4)$ based on its time.

VI. EXPERIMENTAL RESULTS

We have implemented the proposed system in Python and Pytorch, and used a workstation with a GeForce RTX 2080 Ti GPU. For CNN training, we have used Adam optimizer [23]. The learning rate has been initialized to 0.001 and reduced by a factor of 10 after every 10 epochs when the loss does not decrease. The total training time is roughly 4 hours. For RoNIN software, we have downloaded the trained model

¹First, we randomly select a reference frame from the first 85% of an inertial navigation trajectory from a university building, then geo-localize the trajectory by aligning the position and the motion direction at the reference frame with the ground-truth. We follow the trajectory from the reference and crop when it reaches the last frame or within 5 pixels from the border of the 250×250 pixel square centered at the reference frame. We simulate FLP positional error by a 2D Gaussian function with a standard deviation set to 25 pixels (= 5 meters), because FLP at shopping malls exhibits roughly 10 meters positional errors. More concretely, we take the trajectory segment, perturb the first and the last positions of the cropped segment by the Gaussian noise, and solve for a scaled rigid transformation based on the two points to warp the trajectory. We repeat the process and generate twenty samples from each inertial navigation trajectory. Lastly, we apply random horizontal flipping and a random rotation.



Fig. 4. Qualitative evaluation: Visualization of inferred dense location history. DHL is the proposed system at the right. We compare against three other methods in close-ups. The color of the trajectory (blue→red→green) encodes time. For FLP, the trajectory estimation is drawn as a piecewise linear curve, connecting the FLP positions, which are highlighted as circles. For each figure, the ground-truth positions are shown as yellow 'x's, and the corresponding inferred positions are highlighted with square markers. The average localization error is shown by the number at the bottom.

from the official website [2]. Scipy least-squares function has been used for the optimization [24], [25].

A. Quantitative Evaluation

We have compared the localization accuracy against the four competing methods (See Fig. 3):

- **RoNIN** is the inertial navigation trajectory, geo-localized based on the first minute of FLP constraints.
- **FLP-high-freq** is a piece-wise linear trajectory, connecting the FLP locations that are queried at every second.
- **FLP** is the same as FLP-high-freq, except that FLP is queried once every minute for energy efficiency.

• **Mapcraft** is a state-of-the-art map-matching system [17].²

Our test-sites are shopping malls, where the dataset contains 3 buildings with 2 floorplan images each, where we create four testing setups for assessing the generalization capability over different building types. Let us denote the six floorplan images as (A1, A2), (B1, B2), and (C1, C2),

²The code is not publicly available and we implemented locally with a few modification to improve accuracy: 1) Using the RoNIN result as the velocity input; 2) Increasing the search neighborhood by a factor of 1.5 to handle scale inaccuracies in inertial navigation; and 3) Using FLP positions as a feature. MapCraft is memory intensive and does not scale to large floorplans. We developed a heuristic to crop the floorplan based on the FLP positions. The edge of a graph is set to 1 meter.



Fig. 5. Floorplan fusion by CNN. The top and the bottom shows the trajectories (time coded by color, blue→red→green) before and after the CNN application, respectively. CNN is effective in avoiding collisions with walls or inaccessible areas (first 4 columns). The last two columns show failure cases due to large initial alignment errors.

where a letter denotes a building. Suppose synthetic training data was generated on A1.

- **Seen** group is (A1: 1.7 hours).
- **Similar** group is (A2: 1.4 hours), as the overall building structure is similar (i.e., 1st floor vs. 2nd floor).
- **Unseen** group is (B1, B2, C1, C2: 2.0 hours), where buildings are different.
- **Unseen-dense** is (B1, B2: 0.7 hours), one unseen building with dense ground-truth by visual inertial SLAM.

Fig. 3 shows the mean and the first/third quartiles of the localization errors in meters. DHL (w/o Floorplan) is the result of our system after the WiFi/IMU fusion. In comparison to FLP, whose primary information source is WiFi, the result demonstrates the effectiveness of integrating inertial navigation, where the localization accuracy improves by roughly 5 meters (40%). DHL (w/o 2nd itr.) is our system with one iteration of optimization and CNN, which is already better than all the competing methods. DHL shows that an extra iteration further improves the accuracy for about a meter in many cases.

FLP-high-freq indicates the limit of the current industrial state-of-the-art when one affords to call the API every second, which drains battery. The method improves the accuracy over FLP (API call every minute) by roughly 3 meters, but is inferior to DHL by 3 to 5 meters (DHL improves by 25% to 50%), demonstrating the effectiveness of our WiFi/IMU/floorplan fusion approach. MapCraft suffers from discretization errors and the lack of holistic image understanding capabilities like CNN, where the result is inferior to even FLP-high-freq. RoNIN is the raw inertial navigation trajectory, which suffers from severe bending due to the accumulation errors in the device orientation, where the localization errors are worse than 50 to 100 meters.

Unseen-dense evaluation also supports the claims where the fusion of WiFi, IMU, and floorplan consistently improves the localization accuracy, where our method (DHL) is the most accurate. Note that FLP accuracy is worse in the Unseen-dense setup especially for FLP, probably because the 3D tracking phone uses Android 8 while others use Android

10. Furthermore, the data was collected about a year before for Unseen-dense.

Lastly, we have evaluated the contribution of the synthetic training data. The average localization error of DHL increases from 5.57 meters to 6.85 meters on the average over the three main setups (Seen, Similar, and Unseen) when the synthetic data is not used.

B. Qualitative Evaluation

Figure 4 demonstrates how the multi-modal sensor fusion improves the results qualitatively. DHL w/o floorplan shows that WiFi (i.e., FLP positions) successfully removes the severe bending and roughly geo-localizes the inertial navigation trajectory. DHL in comparison to DHL w/o floorplan shows that the floorplan information further refines the trajectory to be more accurate. Figure 5 directly compares the results before and after the CNN-based floorplan fusion. Our CNN effectively avoids walking through walls/escalators and keeps the trajectory within the walkable areas. Our CNN is sometimes able to correct the trajectory over 15 pixels (6 meters). High frequency local motion detail is another difference in our results in contrast to FLP and MapCraft. As demonstrated in the supplementary video, together with the body orientation estimation by RoNIN system, DHL could potentially allow us to calculate the wall that receives the most human gazes and hence is the best place to put advertisement. Please refer to the supplementary material for more renderings.

VII. CONCLUSION

The paper proposes a multi-modal sensor fusion algorithm that fuses WiFi, IMU, and floorplan information to infer dense location history, together with the dataset with the ground-truth. In contrast to the current industrial state-of-the-art (i.e., Google Fused Location Provider), our result is a few orders of magnitude denser and twice as accurate. The current system is designed as an offline process generating location history as opposed to real-time Indoor GPS system. Our future work is the development of real-time Indoor GPS system via multi-modal sensor fusion.

REFERENCES

- [1] "WiGLE - wireless geographic logging engine." [Online]. Available: <https://wgle.net/>
- [2] S. Herath, H. Yan, and Y. Furukawa, "RoNIN: Robust neural inertial navigation in the wild: Benchmark, evaluations, & new methods," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 3146–3152.
- [3] A. R. Jimenez, F. Seco, C. Prieto, and J. Guevara, "A comparison of pedestrian dead-reckoning algorithms using a low-cost mems imu," in *2009 IEEE International Symposium on Intelligent Signal Processing*. IEEE, 2009, pp. 37–42.
- [4] A. Brajdic and R. Harle, "Walk detection and step counting on unconstrained smartphones," in *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*. ACM, 2013, pp. 225–234.
- [5] 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.
- [6] C. Chen, P. Zhao, C. X. Lu, W. Wang, A. Markham, and N. Trigoni, "Oxioid: The dataset for deep inertial odometry," *arXiv:1809.07491[cs.RO]*, 2018.
- [7] C. Yang and H.-R. Shao, "Wifi-based indoor positioning," *IEEE Communications Magazine*, vol. 53, no. 3, pp. 150–157, 2015.
- [8] Z. Chen, Q. Zhu, and Y. C. Soh, "Smartphone inertial sensor-based indoor localization and tracking with ibeacon corrections," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 4, pp. 1540–1549, 2016.
- [9] P. Bahl and V. N. Padmanabhan, "Radar: An in-building rf-based user location and tracking system," in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2. Ieee, 2000, pp. 775–784.
- [10] 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.
- [11] "Fused location provider api." [Online]. Available: <https://developers.google.com/location-context/fused-location-provider>
- [12] 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.
- [13] P. Mirowski, T. K. Ho, S. Yi, and M. MacDonald, "Signalslam: Simultaneous localization and mapping with mixed wifi, bluetooth, lte and magnetic signals," in *International Conference on Indoor Positioning and Indoor Navigation*. IEEE, 2013, pp. 1–10.
- [14] 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.
- [15] 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.
- [16] A. Solin, M. Kok, N. Wahlström, T. B. Schön, and S. Särkkä, "Modeling and interpolation of the ambient magnetic field by gaussian processes," *IEEE Transactions on robotics*, vol. 34, no. 4, pp. 1112–1127, 2018.
- [17] Z. Xiao, H. Wen, A. Markham, and N. Trigoni, "Lightweight map matching for indoor localisation using conditional random fields," in *IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*. IEEE, 2014, pp. 131–142.
- [18] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, and J. Eriksson, "Vtrack: accurate, energy-aware road traffic delay estimation using mobile phones," in *Proceedings of the 7th ACM conference on embedded networked sensor systems*, 2009, pp. 85–98.
- [19] P. Newson and J. Krumm, "Hidden markov map matching through noise and sparseness," in *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*, 2009, pp. 336–343.
- [20] P. Davidson, J. Collin, and J. Takala, "Application of particle filters for indoor positioning using floor plans," in *2010 Ubiquitous Positioning Indoor Navigation and Location Based Service*. IEEE, 2010, pp. 1–4.
- [21] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, ser. LNCS, vol. 9351. Springer, 2015, pp. 234–241, (available on arXiv:1505.04597 [cs.CV]). [Online]. Available: <http://imb.informatik.uni-freiburg.de/Publications/2015/RFB15a>
- [22] M. Buda, A. Saha, and M. A. Mazurowski, "Association of genomic subtypes of lower-grade gliomas with shape features automatically extracted by a deep learning algorithm," *Computers in biology and medicine*, vol. 109, pp. 218–225, 2019.
- [23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: <http://arxiv.org/abs/1412.6980>
- [24] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.
- [25] J. J. Moré, "The levenberg-marquardt algorithm: implementation and theory," in *Numerical analysis*. Springer, 1978, pp. 105–116.