Multi-Level Scene Modeling and Matching for Smartphone-Based Indoor Localization

Lidong Chen* [1,2], Yin Zou[1], Yaohua Chang[2], Jinyun Liu[2], Benjamin Lin[2] and Zhigang Zhu[2,3]

[1] College of Systems Engineering, National University of Defense Technology, Changsha, Hu'nan, P.R. China

[2] Department of Computer Science, The City College of New York, New York, USA

[3] Department of Computer Science, The CUNY Graduate Center, New York, USA

ABSTRACT

Accurate indoor positioning has attracted a lot of attention for a variety of indoor location-based applications, with the rapid development of mobile devices and their onboard sensors. A hybrid indoor localization method is proposed based on single off-the-shelf smartphone, which takes advantage of its various onboard sensors, including camera, gyroscope and accelerometer. The proposed approach integrates three components: visual-inertial odometry (VIO), point-based area mapping, and plane-based area mapping. A simplified RANSAC strategy is employed in plane matching for the sake of processing time. Since Apple's augmented reality platform ARKit has many powerful high-level APIs on world tracking, plane detection and 3D modeling, a practical smartphone app for indoor localization is developed on an iPhone that can run ARKit. Experimental results demonstrate that our plane-based method can achieve an accuracy of about 0.3 meter, which is based on a much more lightweight model, but achieves more accurate results than the point-based model by directly using ARKit's area mapping. The size of the plane-based model is less than 2KB for a closed-loop corridor area of about 45m*15m, comparing to about 10MB of the point-based model.

Keywords: Indoor localization, lightweight model, multi-level mapping, region segmentation, plane matching.

Index Terms: Computing Methodologies－Mixed / Augmented Reality; Computing Methodologies－Computer Vision－Vision and Scene Understanding

1 INTRODUCTION

In the past two decades, accurate indoor localization and tracking has attracted a great deal of attention for its widespread applications in large indoor environments, such as airports, supermarkets and hospitals. Once accurate location information is available in real time, various practical location-based services could be provided, such as indoor navigation services, location information push, advertising market services, etc. Unfortunately, compared with the commonly used GPS technology in outdoor environments, there is still an absence of a standard positioning system that can be widely applied in indoor environments [1].

Since GPS signal does not work in indoor environments, a variety of radio-based methods that use signal frequencies or strengths, such as Wi-Fi [2], WLAN, Bluetooth [3], UWB or RFID, are widely researched to realize indoor positioning. However, due to the variations of radio signals in complicated indoor environments, all these positioning methods tend to have large fluctuations, leading to poor accuracy. Moreover, extra infrastructure must be deployed ahead in indoor environments, and additional signal receiving devices are needed at users' side.

E-mail: nudtdong11@163.com

On the other hand, with rapid development of the mobile device industry, smartphones now are equipped with various kinds of powerful on-board sensors, including accelerometers, gyroscopes, compasses, proximity sensors, depth sensors, cameras, etc.[4]. Pedestrian dead reckoning (PDR) [5] is a well-known technique to track a user's current position based on previous positions, step length and motion direction [6]. Furthermore, in cooperation with visual sensors, the Visual-Inertial Odometry (VIO) technique [7] can achieve much better performance on motion tracking, due to the complementary characteristics of these two sensing modalities. However, the major disadvantage of this kind of methods is the accumulative drift error. For long-distance and long-term tracking, additional global mapping and/or other physical constraints are necessary to eliminate the cumulative error.

So far, highly accurate and practical smartphone-based indoor localization remains an open problem. In this paper, a hybrid indoor localization method is proposed solely based on a single smartphone, which takes full advantage of various on-board sensors. Based on the powerful high-level APIs of ARKit platform [8] provided by Apple Inc., the proposed method integrates visual-inertial odometry (VIO), area mapping based on point cloud, and 3D space modeling based on plane detection. The main contributions of this paper include: (1) A real-time implementation solely on a smartphone, by fully utilizing the functionalities well developed in the AR platform ARKit. (2) Multi-level area mapping and localization by leveraging different types of features (point-based and plane-based) in an indoor environment. (3) Highly lightweight models leading to more efficient performance in storage space and processing time yet achieving high accurate localization.

2 RELATED WORK

Modern smartphones are equipped with a variety of sensors, which can be combined to obtain precise positioning results. Meanwhile, powerful computing performance of smartphones also makes real-time mobile localization possible. According to the types of utilized on-board sensors and the measuring principles, smartphone-based indoor positioning methods can be classified into two groups: Inertial based methods, further cooperated with visual information, and radio-based methods, typically using fingerprinting technology.

2.1 Inertial based methods

Using a combination of on-board inertial sensors, i.e. inertial measurement units (IMUs), consisting of an accelerometer and a gyroscope, sometimes also a magnetometer, the current position of a user can be determined based on its previous position, step length and motion direction. This widely used technique is called pedestrian dead reckoning (PDR) [5]. The motion of the user is measured step by step, and current position is tracked relative to the starting point without the need of any extra physical infrastructure deployed in the environment. Unfortunately, the accuracy of position estimation would be low if the motion is too rapid or violent. Visual information can then be introduced to tackle the problem of violent movement by the utilization of an on-board camera, via Visual-Inertial Odometry (VIO) algorithms. VIO algorithms can be classified into two categories: filter-based [9] and optimization–based [10]. However, the major disadvantage of PDR or VIO methods is their cumulative error, which limits their applications in a very short distance. Sensor drift is the main drawback, which makes it impossible to extend to long-distance localization and long-term tracking, even using the EKF method [11]. Additional global maps and physical constraints are needed to reduce the accumulative drift error, especially for orientation drift error. Otherwise, the sensor drift will be accumulated and become more and more severe along the walking distance, if only the relative information is leveraged.

2.2 Radio-based methods

Generally, all these methods in this category are built upon a wireless network deployed in an indoor environment in advance. Multiple access points or signal transmitters are distributed in the space. The characteristics of the received radio signals, including time of flight (TOF), frequency or received signal strength (RSS), are used to estimate the position of the user carrying a mobile device. Apart from using traditional trilateration measurement in very simple scenarios, normally the measuring principle of these methods needs a prior process of training in the target space, which is called the fingerprinting technique [12]. The RSS-based fingerprinting method firstly collects a database of features in many locations of the scene, and then online measurements at current location will be matched with the closest fingerprint of a priori location. The main challenge to the location fingerprinting technique is that the received signal strength could be affected by diffraction, reflection, and scattering in a propagation indoor environment. Besides, it also suffers from multi-path problems. Hence, the accuracy of radio-based methods is still limited at around meter-level. Therefore, some researchers propose to fuse radio-based and PDR techniques to achieve better localization accuracy [13]. The improvement of accuracy is achieved upon the increasing of computation complexity. Moreover, extra infrastructure and prior training process are also needed.

In the last few years, some large technology companies have been committing to develop augmented reality (AR) platforms, most prominently Tango [14] and ARCore from Google [15], and ARKit from Apple [8]. These platforms provide high-level APIs on world tracking, space modeling, scene understanding and so on. Based on a basic survey of these platforms, we choose an Apple iPhone that supports ARKit as our smartphone device to realize indoor localization. ARKit is a mobile AR platform for developing augmented reality apps on iOS. It provides a high-level API interface containing a powerful set of features. Firstly, tracking based on visual inertial odometry (VIO) is the core functionality of ARKit. It is the ability to track the mobile device in real time, which provides the ability to get the device's relative position in physical environments. Moreover, ARKit also provides an ability of real-time plane detection for scene understanding, which determines surfaces or planes in the physical environment. In ARKit 2 released on June 2018, the ARWorldMap class provides the ability to save world maps representing physical 3D space, and then these maps can be reloaded the next time when a user visits the same physical environment to obtain a “geo-referenced” localization experience. ARWorldMap offers a powerful foundation to realize indoor relocalization easily, which will be detailed in the next section as one of our baselines, although its ability is still limited.

3 Hybrid Modeling and Localization Approach

The system diagram of the proposed hybrid modeling and localization approach is illustrated in Figure 1. The whole system is implemented on a standard iPhone device based on the ARKit platform developed by Apple Inc. The main work for achieving indoor localization in this paper can be divided into two parts: hybrid mapping (modeling) and hybrid matching (localization).

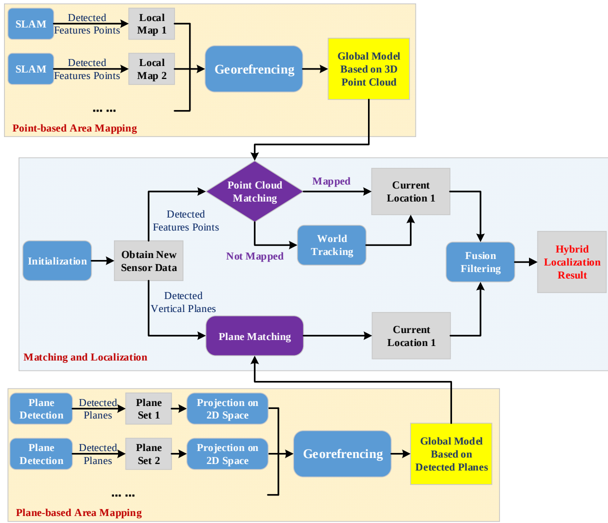


Figure 1: The proposed hybrid modeling and localization approach

In the hybrid mapping part, two different modules for *modeling* a 3D indoor scene are presented: point-based area mapping and plane-based area mapping. The point-based area mapping module is based on the traditional SLAM (Simultaneous Localization and Mapping) method [16]. The ARKit platform has provided a powerful feature ARWorldMap in its newer version ARKit 2. The ARWorldMap object stores all the raw feature points as you scanned, which represents mapping of physical 3D space. Notably, the environment will have to be rich with visual features (such as doorplates, posters, fixtures, etc.) to generate a usable point-based area map. Then, the local area map stored in ARWorldMap object could be retrieved and loaded to realize relocalization next time. The point-based approach is used as one of the two modules with two purposes: as a baseline model to evaluate our plane-based module in this paper, and the point-level module for our multi-level fusion in the future. The plane-based area mapping module, on the other hand, is based on plane detection in 3D space. In the current implementation, we use vertical planes that are rich and unique in typical indoor environments for space localization, such as walls, windows, pillars, posters, etc. Moreover, ARKit also provides a basic API to determine spatial surfaces as geometric planes automatically in physical environments. The plane detection ability of ARKit can also extend and update the ranges of detected planes simultaneously. All these local area maps are integrated into a global model aligned with the same indoor floor plan.

After the modeling of the indoor environment, the hybrid matching part is used for localization, through the matching between the current scene information captured by various sensors and the saved global models. Since there are two different area models, one is at point-based and the other is at plane-based, we propose a hybrid multi-level matching and localization method. Each of the two models and its corresponding matching results have advantages and disadvantages in different scenic conditions: the point-based model is preferable when rich and unique visual features present in the indoor scene, whereas the plane-based model is desirable when the scene is populated with various distinguished vertical planes. The two matching and localization results could be fused by using filtering methods to get a better final result, such as EKF [17] or particle filtering [18].

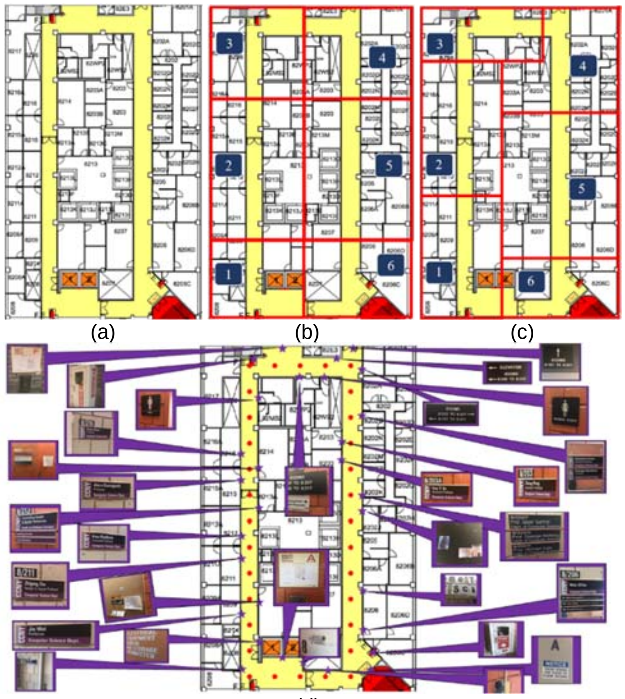
3.1 Region segmentation for hybrid mapping

Point-based area mapping. Generally, it is difficult to store the whole scene information of a large area (such as a building) into only one point-based area map. The size of model will be too large, and it will bring a great challenge to computational efficiency while doing matching for localization. Specifically, for point-based area mapping, by conducting numerous real-scene experiments on the ARKit platform, we have found that the size of a point-based area map stored in an ARWorldMap object is limited to only a few Megabytes. Therefore, we need to divide the whole area into multiple local areas, and then scan each area to generate the corresponding local model. Finally, a georeferencing process is needed for integrating local area models into a global model, which means to align all the local models into a unified world coordinate system using a 3D rigid transformation. Details of the alignment process will be described below. Then, the global model of the whole physical space is generated.

Figure 2(a) shows a floor plan of a campus building. As shown in Figure 2(b), the easiest way in region segmentation is to divide the whole area into several regular rectangle regions. However, we have found that the point cloud area maps created by ARWorldMap in ARKit API can only find matches in the localization stage at few spots where there are salient visual features in such an indoor scene. As shown in Figure 2(d), there are only about twenty spots along a close-loop corridor with a length of more than 100 meters. Thereby, an improved strategy of irregular region segmentation is proposed as shown in Figure 2(c). Each boundary between two neighboring local areas is located at somewhere with salient visual features, which ensures that the system can quickly find a right match and perform localization based on the new-loaded map when the user enters a new area. Furthermore, while scanning the real scene, each two consecutive regions have overlaps to achieve a smooth area switching.

Plane-based area mapping. Planes are very common in real indoor environments, such as walls, pillars, posters, doors, and ground. The spatial structure of the 3D space can be well represented by a set of dominant planes. Fortunately, ARKit provides the ability of plane detection to determine both horizontal and vertical planes in physical environments. For indoor localization, vertical planes such as walls and sides of pillars are rather unique in positioning the user with the phone.

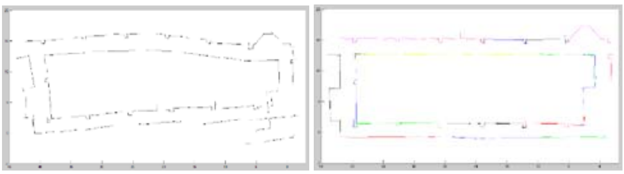
For the plane-based area mapping, it is feasible to store all the detected planes in one model, because we just need to store the necessary parameters of each plane, which is highly lightweight. Here, to simplify the modeling and mapping process, we first project all the detected vertical planes onto the 2D floor plan, i.e. the *xoz* plane while the coordinate system's *y*-axis is parallel to the direction of gravity. However, if we use the plane detection and tracking results directly from ARKit, as shown in Figure 3(a), accumulative drift error will cause very low accuracy of the area model. Therefore, the aforementioned strategy of region segmentation is also employed here for plane-based modeling. As shown in Figure 3(b), each different color of the lines indicates one-time scanning and modeling process of the real environment. Finally, all the plane-based local models consisting of several projected lines will be aligned to a unified world coordinate system using 2D similarity transformations to the floor plan, thus creating a "geo-referenced" plane-based global model.



(a) (b) (c)

(d)

Figure 2: Illustration of region segmentation for 3D point cloud modeling. (a) A 2D floor plan of one floor of a campus building. (b) Regular region segmentation. (3) Adaptive region segmentation based on salient feature spots. (4) Salient feature spots along the corridor.



(a) (b)

Figure 3: Illustration of region segmentation for plane-based modeling (a) Results with only one-time area mapping process. (2) Results with region segmentation and global alignment.

Details on geo-referenced models. For the georeferencing process of both the above two modeling methods, several "landmark" points (ground-truth points on the floor) are set in each local area in advance. The accurate physical positions of each landmark point are measured accurately and recorded in advance in the pre-defined and unified world coordinate system. While scanning the real scene and modeling the local area, the estimated location at each landmark point is recorded in the corresponding local coordinate systems. Then, a transform matrix between each local coordinate system and the unified world coordinate system can be calculated based on the local and global coordinates of these landmark points. Since the indoor localization problem in this paper is simplified by projecting the 3D space onto 2D xoz plane (the floor plan), only two landmark points for each local area are needed to calculate a 2x3 similarity transform matrix.

Compared with the point-based area model, the plane-based model is much more lightweight. For each plane, which is projected on the ground (the floor plan), we only need to store the coordinates of the two end points of its projected line segment on the 2D *xoz* floor plane.

3.2 Hybrid matching and localization

For the two different levels of space models, the corresponding matching and localization methods are also different, which are detailed as follows.

3.2.1 3D point cloud mapping and relocalization

An overview of the proposed mapping and localization method based on 3D point cloud modeling is presented in Figure 4. Firstly, the initial position of the device need to be acquired by some initialization techniques, such as GPS location at the entrance of an indoor environment. Here, the initial position is not necessary to be very accurate. We just need a rough location to determine at which local area (region) the current location is. Then, the corresponding local area map will be loaded, i.e. a partial 3D point cloud enhanced with visual feature points in this area. Once the new map is loaded, a new AR session will be reconfigured and started. Then, each current frame while scanning the physical space will be checked simultaneously whether it can be mapped with somewhere stored in the point-level area model or not. This mapping process is automatically realized by ARKit. Once mapped successfully, the current position in the local coordinate system of the local area model is acquired. Meanwhile, the world tracking functionality of ARKit is launched. The device's relative position is continuously tracked based on the captured images, as well as motion data from on-board inertial sensors. The real-time tracking and localization is realized upon the technique of Visual Inertial Odometry (VIO) [7].

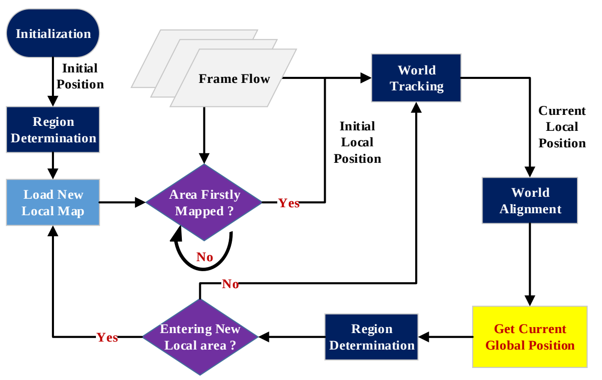


Figure 4: Diagram of point-based matching and localization method

Since the estimated local position at each frame has been aligned to a pre-defined world physical space based on the corresponding transform matrix of the current local map, a geo-reference position of the current frame can be determined and therefore is drift-free. Furthermore, the real-time positioning process will be checked at each frame whether the device has entering a new local area or not. If so, the corresponding new local map will be loaded. Once the new local map is firstly mapped, a new loop of mapping and localization continues.

3.2.2 Plane detection and simultaneous matching for localization

A simplified RANSAC strategy is employed in the plane-based indoor localization algorithm. The algorithm has the following steps:

1) Randomly select two line segments l_i and l_j (representing two vertical planes) in the current set of detected planes *Cur_subset*, then check whether the two lines satisfy the following two conditions simultaneously: ①The orientation difference of the two lines is large enough (e.g., $> 45^\circ$); ② The distance between the two line segments is within a certain range. If not, select again.

2) Randomly search a pair of line segments L_m and L_n in a subset *subModel* of planes in the global model. Here, the subset is filtered within a certain range near the current estimated position. The two line segments L_m and L_n should satisfy both of the following two conditions: ①The angle deviation between the match l_i and L_m is the same as the angle deviation between the match l_j and L_n (within a small error tolerant range, e.g. 5 degrees); ②The distance between l_i & L_m and the distance between l_j & L_n are both within a certain error range. Then, we assume the two pairs l_i & l_j and L_m & L_n as a candidate match.

3) Calculate the transform matrix T between *Cur_subset* and *subModel*, based on the orientation difference of the two pairs of matching lines (l_i & l_j and L_m & L_n) and the offset between the intersection points of the two line pairs. Here, we treat T as a 2D similarity transformation for simplicity.

4) Apply the transform matrix T to all the line segments in the current detected set *Cur_subset*, and then the corresponding transformed line set *Cur_subset_T* is obtained.

5) For each transformed line segment in the set *Cur_subset_T*, find the best matching line segment in *subModel*. Record the highest matching score of each line segment in the set *Cur_subset_T*, while the matching score between two line segments is defined as:

$$ms_{ij} = AngleDev_{ij} * Dist_{ij}$$

Here, $AngleDev_{ij}$ is the orientation difference between the two line segments, and $Dist_{ij}$ is the distance between the two line segments.

6) If the above matching score is less than a threshold, the two line segments can be treated as a potential matching, and then the line segment *Cur_subset_Ti* in the set *Cur_subset_T* is temporarily labeled as an "inlier". Count the total number of inliers for the current candidate matching T, and sum the matching score of each potential matching.

7) Repeat the above steps 1)-6), then select one of the candidate matching which has the largest number of inliers as the best matching between *Cur_subset* and *subModel*. If there are two candidate matches with the equal number of inliers, choose the one that has better matching score as the best matching.

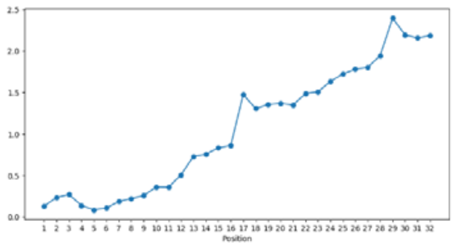
8) The current position in the global coordinates system can be calculated by multiplying the original position with the best-matching transform matrix T.

4 Experiments and Performance Evaluation

The proposed hybrid modeling and localization approach is evaluated in a typical indoor environment with a closed-loop corridor as shown in Figure 5(a). The size of the whole area is about 45m*15m. We uniformly set 32 ground-truth landmark points on the floor of the area; the green line in the figure shows the looping route starting from the bottom-left corner. The distance between two neighboring landmark points is 3.35m, and the positions of all these landmark points were accurately measured in advance to be treated as the ground truth of indoor localization.

Firstly, a simple experiment is implemented to illustrate the effect of accumulative drift error of SLAM. The localization results are directly obtained by the world tracking module based on visual inertial odometry (VIO) provided by ARKit, without any modeling process. As shown in Figure 5(b), the radius of the green circle at each landmark point equals to the localization error at this point. We can see that, the localization error increases steadily with the increasing walking distances from the starting point. After a loop route along the corridor with a length of about 100 meters, the cumulative error will be more than 2 meters. Even worse, the error will continue to grow much larger loop after loop. The angular error is more pronounced as see in Figure 5(c), the orientation difference of the rectangular corridor is as large as 25 degrees between the first loop and the third loop of the travels of the corridor. Figure 5(d) shows the location error of the first loop of the corridor travel, from landmark #1 to landmark #32.

(a) (b) (c)



(d)

Figure 5: Illustration of experimental setup and accumulative drift error. (a) Plane map and landmark locations; (b) Location errors on the floor plan; (c) Orientation errors in multiple loops of trajectories; (d) Plot of the location error of the first loop.

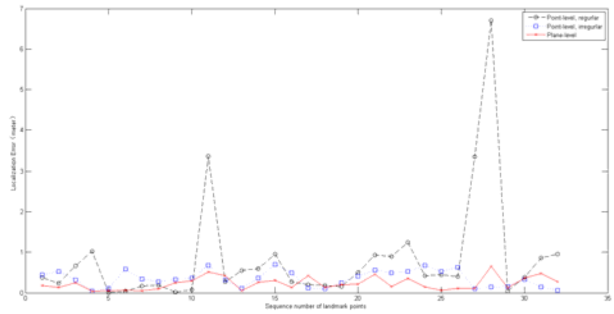
Therefore, creating area maps (models) of the indoor environment in advance is necessary to eliminate the cumulative error. Three experiments using different modeling schemes are implemented for indoor localization, as detailed below. The Localization error distribution of different methods is illustrated by the radii of green circles, as shown in Figure 6.

(a) (b) (c)

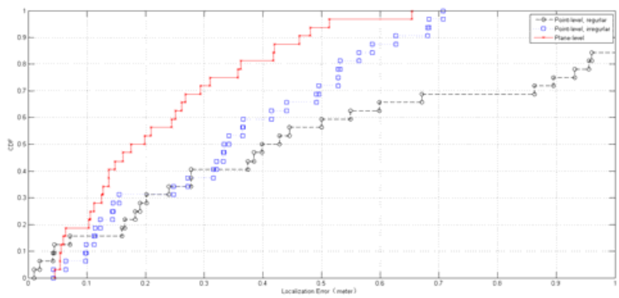
Figure 6: Localization error distribution of different modeling and matching methods. (a) Point-based method using regular region segmentation; (b) Point-based method using adaptive region segmentation; (c) Plane-based method

The first experiment is based on point-level modeling and matching with regular region segmentation (Baseline 1). We can see from Figure 6(a) that, most of the landmark points are located well with a reasonable error less than one meter. Cumulative errors can be significantly reduced at any point where the current view of the smartphone can be mapped successfully to the point-level model stored in an ARWorldMap object. However, at some of the landmark locations, such as #11, #27 and #28, localization error is as large as over 3 meters. The reason is that each of these landmark points is nearing the boundary between two neighboring local areas, but there are no salient visual features around the boundary. While crossing the boundary to enter a new local area, the new local area map will be loaded, and then the positioning output will remains at the boundary for several seconds. The reason is that the transform relationship between the current local coordinate system and the global coordinate system has not been established before the new local map is mapped successfully for the first time.

To overcome this problem, the second experiment based on point-level modeling with irregular region segmentation is implemented (Baseline 2). Each boundary between two neighboring local areas is located at somewhere very close to the spots with salient visual features. Consequently, the new loaded local model will be mapped quickly after crossing the boundary to enter a new local area, and then world tracking of ARKit and simultaneous global localization will be renewed soon. The localization results are much better as shown in Figure 6(b).



(a)



(b)

Figure 7: Quantitative comparison of three different localization methods: point-based using regular region segmentation, point-based using adaptive region segmentation and plane-based. (a) Error measured over the 32 landmarks. (b) The cumulative distribution function (CDF) of localization error.

The third experiment is based on plane-level modeling and matching. Vertical planes in 3D space are detected by utilizing ARKit API, and then a global plane-level model as shown in Figure 3(b) is generated in advance. This experiment acquires much better localization results as shown in Figure 6(c). More importantly, the plane-level space model is very lightweight. For each detected vertical plane, we only need to store the coordinates of the two end points of its projected line segment on the 2D xoz plane. If we use a single-precision floating-point format to store one coordinate value, the storage size of one plane will be 4*4=16 bytes. Then, the total size of the plane-level model of the 45m*15m experimental space is only 1,952 Bytes. In contrast, the size of the above point-based model of the same area is 10.2 Megabytes.

The quantitative comparison of localization error of the above three experiments is shown in Figure 7(a), and their root mean square error (RMSE) of localization are 1.549 m, 0.405 m and 0.278 m, respectively. The cumulative distribution function (CDF) of localization error is shown in Figure 7(b).

Furthermore, the qualitative comparison between the above two different modeling and matching methods at different levels, i.e. point-level and plane-level, is discussed in Table 1.

Table 1. Comparison between point-level and plane-level methods

Methods	Advantages	Disadvantages
Point-level modeling and localization using adaptive region segmentation	✓ Easy to implement based on ARKit platform ✓ Autonomous mapping and world tracking	o Rely heavily on salient visual features o Could get stuck while entering a new local area o Much larger file size of model
Plane-level modeling and localization	✓ Higher accuracy ✓ Lightweight model ✓ Suitable for areas with very few salient visual features	o Rely heavily on detection of vertical planes o May have ambiguity in plane matching

5 CONCLUSION

In this paper, a hybrid mobile indoor localization method is proposed based on multi-level scene modeling and matching solely using single off-the-shelf smartphone. By utilizing high-level APIs provided by the popular augmented reality developing platform ARKit, both point-level and plane-level scene modeling and matching methods are conducted simultaneously on a single smartphone. Region segmentation and physical alignment are researched to realize global modeling of real environment. A simple and high-efficiency plane matching algorithm based on RANSAC strategy is designed. The experimental results verify that the proposed hybrid method has a high localization accuracy of about 0.3m, and the plane-level model is very lightweight with a total size of only 1.9 KB for a 45m*15m experimental space. The real-time localization can be implemented based on the world tracking functionality provided by ARKit. Experimental results show the main merits of our method: high accuracy in localization, lightweight models, and solely smartphone-based without any extra infrastructure.

Both of the two different levels of modeling, i.e. point-level and plane-level, have its own advantages and disadvantages in different scenic conditions, as discussed in experimental section. Our ongoing research focuses on fusing point-level and plane-level localization methods together to achieve complementarity of each other. In the application side, we will develop a navigation app using the hybrid localization approach for providing navigation guidance for people in need, such as individuals who are blind or visually impaired.

6 ACKNOWLEDGMENTS

This work was mostly done with the first author was visiting The City College of City University of New York, supported by both the National University of Defense Technology and the City College Visual Computing Laboratory. The work was also funded by the National Science Foundation via awards #CNS-1737533 and #IIP-1827505, as well as NYSID via the CREATE program.

REFERENCES

[1] A. Correa, M. Barcelo, A. Morell, and J. Vicario. A review of pedestrian indoor positioning systems for mass market applications. *Sensors*, 17(8): 1927, 2017.

[2] N. Chang, R. Rashidzadeh, and M. Ahmadi. Robust indoor positioning using differential Wi-Fi access points. *IEEE Transactions on Consumer Electronics*, 56(3): 1860–7, 2010.

[3] Y. Zhuang, J. Yang, Y. Li, et al. Smartphone-based indoor localization with bluetooth low energy beacons. *Sensors*, 16(5): 596, 2016.

[4] H. H. Hsu, J. K. Chang, W. J. Peng, et al. Indoor localization and navigation using smartphone sensory data. *Annals of Operations Research*, 265(2): 187-204, 2018.

[5] W. Kang and Y. Han. SmartPDR: Smartphone-based pedestrian dead reckoning for indoor localization. *IEEE Sensors journal*, 15(5): 2906-2916, 2015.

[6] M. Alzantot and M. Youssef. UPTIME: Ubiquitous pedestrian tracking using mobile phones. *Proceedings of* 2012 *IEEE the Wireless Communications and Networking Conference*, pp. 3204–3209, 2012.

[7] J. Gui, D. Gu, S. Wang, et al. A review of visual inertial odometry from filtering and optimisation perspectives. *Advanced Robotics*, 29(20): 1289-1301, 2015.

[8] Apple Inc. https://developer.apple.com/videos/play/wwdc2018/602/

[9] Z. Chen, H. Zou, H. Jiang, et al. Fusion of WiFi, smartphone sensors and landmarks using the Kalman filter for indoor localization. *Sensors*, 15(1): 715-732, 2015.

[10] S. Leutenegger, S. Lynen, M. Bosse, et al. Keyframe-based visual–inertial odometry using nonlinear optimization. *The International Journal of Robotics Research*, 34(3): 314-334, 2015.

[11] Z. Chen, H. Zou, H. Jiang, et al. Fusion of WiFi, smartphone sensors and landmarks using the Kalman filter for indoor localization. Sensors, 15(1): 715-732, 2015.

[12] M. Azizyan, I. Constandache, and R. R. Choudhury. SurroundSense: mobile phone localization via ambience fingerprinting. *Proceedings of the 15th annual international conference on Mobile computing and networking*, pp. 261-272, ACM, 2009.

[13] Q. Tian, Z. Salcic, K. I-K. Wang, et al. A hybrid indoor localization and navigation system with map matching for pedestrians using smartphones. *Sensors*, 15(12): 30759-30783, 2015.

[14] V. Nair, M. Budhai, G. Olmschenk, et al. ASSIST: personalized indoor navigation via multimodal sensors and high-level semantic information. *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[15] Google Inc. https://developers.google.com/ar/

[16] H. Durrant-Whyte, T. Bailey. Simultaneous localization and mapping: part I. *IEEE robotics & automation magazine*, 13(2): 99-110, 2006.

[17] Z. A. Deng, Y. Hu, J. Yu, et al. Extended Kalman filter for real time indoor localization by fusing WiFi and smartphone inertial sensors. *Micromachines*, 6(4): 523-543, 2015.

[18] J. Luo, and S. Qin. A fast algorithm of simultaneous localization and mapping for mobile robot based on ball particle filter. *IEEE Access*, 6: 20412-20429, 2018.