STEPS - InDoor Visual Navigation Framework for Mobile Devices

Yael Landau[1], Revital Marbel[1], and Boaz Ben-Moshe[1]

[1]Dept. of Computer Science, Kinematics and Computational Geometry lab. Ariel University, Israel

Abstract. This work presents a vision-based navigation system designed for Indoor localization. The suggested framework operates as a standalone $3D$ positioning system by fusing a sophisticated optical-flow pedometry with map-constrains using an advanced particle filter. The presented method requires no personal calibration and works on standard smart-phones with relatively low energy consumption. Preliminary field experiments on Android smart-phones show that the expected $3D$ error is about $1-2$ meters in most real-life scenarios.

1 Introduction

Indoor positioning is an important capability for a wide range of applications including: location base services (LBS), public safety (first responders) and autonomous robotics (indoor navigation). While LBS related applications mainly target smart-phone users navigating in a shopping mall [1, 2], first responders may be using foot mounded pedometer (see [3–5]). Although such solutions were presented by several research-groups in the last two decades - the robustness and accuracy of existing indoor positioning systems (IPS) are often insufficient [6].

Utilizing particle filter for localization problem is common, both for indoor and outdoor scenarios [7]. In essence, the **sense**, **action** and **re-sample** functions of each implementation differs one algorithm from another. Moreover, harnessing the smart-phone internal sensors can be performed in a wide range of techniques. Although there are many different types of applications which require indoor pedestrian positioning, it seems that the following properties should be optimized with respect to almost any such method:

Accuracy: Often the main and foremost parameter which is being tested.

Keep It Simple: Simplicity is a key factor: the system should work automatically with no manual overhead operation or calibration.

Real-time: For a natural and intuitive positioning results.

Privacy: The suggested solution should be able to work in an "off-line" mode (i.e., "flight-mode" or "standalone" mode).

Bring your own device: The suggested solution should work on existing $COTS$ smart-phones.

1.1 Our Contribution

This work presents a smart-phones indoor positioning system (IPS) based on recent AR and MR (Augmented and Mix Reality) tools such as Google's $ARCore$ or Apple's $ARKit$. The AR tools are used as visual pedometry (scaled optical-flow) sensor, which is then fused with an advanced version of localization particle filter to produce a both accurate and robust solution for various indoor positioning applications. The presented method allows a simple and efficient mapping solution that, combined with the localization particle filter, allows 1-2 meter positioning accuracy in most standard indoor scenarios.

2 The basis of Indoor Position

The user global position can be retrieved from existing geolocation services (e.g., Google Maps Geolocation API). Such user location is commonly approximated using RF signals ($4G-3G$, $WLAN$, BLE) and even weak global navigation satellite system ($GNSS$). The accuracy of such methods is considered to be "building level" (10-30 meter) or "room level" (5-10 meter).

The user relative position is often computed using a pedometer. Smart-phone based pedometer is composed of two major virtual-sensors: (i) "Step-counter": which detects discrete step-events. (ii) Orientation sensor: which approximates the user global / relative direction. Combined the two parts allow a step based relative path computations. Naturally such method tends to drift in time (and steps). Many modern IPS are combining the above two positioning method in order to allow an accurate and global localization (with no drift).

2.1 Basic Particle Filter for Localization

This section discusses possible naive particle filter algorithm for localization estimation. Since particle filter method represents the posterior distribution of a set of particles P ($|P| = n$) on a given map, the result of such algorithm (for each step) is a new set of particles P' with a (slightly) different distribution. The goal of this algorithm is to get all the particles to converge into a single (compact) region on the map in few steps (re-sampling). Figure 1 presents a basic demonstration of such process. After converging, the algorithm computes the expected position using some variant of weighted average over the particles (or simply reports the location of the particle with the highest grades / likelihood). Before presenting the algorithm some terms should be clarified:

- **Map:** The particle filter methods estimates the internal state in a given region of interest (ROI). Thus, the input of this algorithm is a $2D$, $2.5D$ or $3D$ map of the region, this map should include as many constraints as possible (for example rooms, walls, doors, stairs, etc'). The map constrains are used to determine each particle grade as particles with impossible location on the map will be downgraded.

- **Particle:** At the beginning of the localization process we "spread" a set of particles P on the map. Each particle $x_i \in P$ will have these attributes: location: $< x, y, z >$, orientation: w and grade: g. In each step all particles location and orientation will be modified as well as their grades. Since these particles represent the internal state-distribution, the sum of P particles grade is 1 in each step. At the initial step each particle x_i grade is $\frac{1}{|P|}$. The grade of each particle will be set higher as its location on the map seems most likely to represent the internal state.
- **Move function (Action function):** With each step all the particles in the map should be relocated according to the internal movement. Hence, for each step we calculate the movement vector (in $2D$ or $3D$) and the difference in orientation, then we move all the particles accordingly. The movement of each step is computed by the pedometer (step counter with orientation) as commonly used in smart-phone.
- **Sense function:** The sensors of the device are used to determine each particle grade. The sense method predict each particle sense for each step and then grade it with respect to the correlation between the particle prediction and the internal sense. In the simplified case, the sense function computes the distance from each particle to the general estimated position (computed by some geo-location service) and evaluate each particle accordingly, e.g., if the distance from some particle p is larger than the estimated error (of the geo-location service) reduce the grade of p. The map constrains are also used in order to evaluate the probability (i.e., grade) of each particle.
- **Re-sampling:** The process of choosing a new set of particles P' from P. The re-sampling process can be done using various methods but the purpose of this processes is common; to choose the particles with high weight (i.e., grade likelihood) over the low weight ones.
- **Get best**: The method that compute the output of the particle filter algorithm - the estimated position. Traditionally performed by one of three ways: return the best particle position, return the weighted average position of the particles or the combined approach, return the weighted average position of the particles that are in the range of some determined radius from the best particle.

Algorithm 1 presents the process of $2D$ localization using particle filter method with mobile pedometry sense (see Figure 1).

The naive algorithm is relatively time efficient, however, its precision might be insufficient in cases of large areas with few constraints. In the next section we proposes an improved version of the particle filter algorithm which support $3D$, has better accuracy and improved robustness.

3 Advanced Algorithm

In this section an advanced localization algorithm is suggested: An improved map-constrains combined with adjusted sense function allows better accuracy.

Input: Black and white $2D$ map of the navigation area.
Init: generate a set P of n particles, each with grade $\frac{1}{|P|}$. For every $x_i \in P$ a random location $< x, y >$ is set in uniform distribution over the map.
Result: Estimated location: $pos =< x, y >$
for *Each step* **do**

1. Calculate the step vector s_i.
2. Apply the Move function on all particles in P by s_i.
3. Apply the sense function on each particle in P according to the current geo-location likelihood position.
4. Evaluate the weight of each particle according to its new position on the map.
5. Re-sample all particles into P'.
6. Estimating the current position by calculating the particles' average location in P', considering their weights.

end

Algorithm 1: Generic particle filter localization algorithm: a black and white map is used in order to present the geo-constrains used by the particle filter.

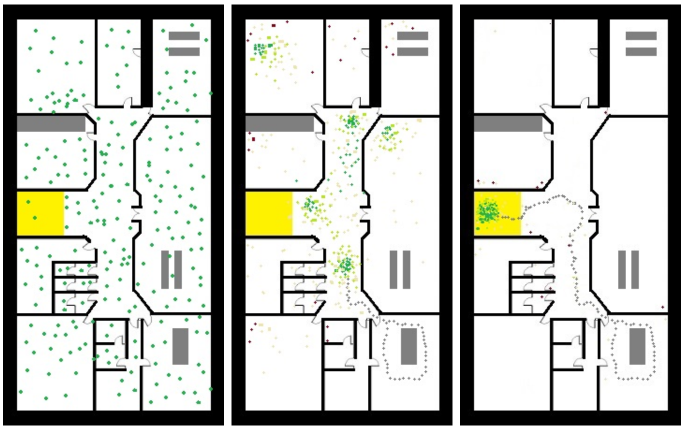

Fig. 1. Particle Filter for localization. Left: Init state, the particles are uniformly distributed. Middle: using the short motion vector the particles are beginning to organize in few clusters. Right: the particles converged to a single position cluster - mainly due to floor change detection. The lower graph shows the barometer raw measurements (PSI) in time.

The next subsections explain the improved mapping process and the advanced particle filter algorithm.

3.1 Mapping - multi floors and $3D$ position

The advanced particle filter algorithm relies on the existence of a pre-made map of the region of interest. Such map assembled by our system using to the following technique:

1. AR measurements tools surface detection, which allows us to conclude the sampled region of interest boundaries.
2. The map is represented in the form of painted image, using the defined colors: A, B, C, D to represent the verity of the different constrains.

The colors will be placed on the map according to the following logic:

– **A**: Accessible area.
– **B**: Unaccessible area, such as walls, fixed barriers, etc. Sensed by AR tool.
– **C**: Partially accessible regions. This area represents locations with relatively low probability for a user to at (e.g., tables)
– **D**: Floor changing regions, such as stairs, escalators and elevators.

A $2.5D$ map such as presented in Figure 2 will be the base for the particle filter algorithm, and will later on be used to determine the particles grade.

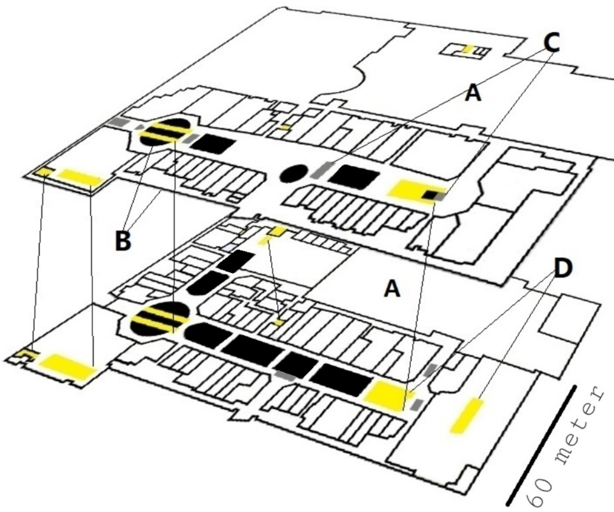

Fig. 2. $2D$ multicolor map example used in the advanced algorithm. The Color white is (marked by the letter A) represent accessible areas , the black color (marked as B) represent the fixed ares (in this case walls), the grey (marked as C) represent dynamic unaccessible areas (tables in this case) and the yellow part (marked as D) represent stairs and elevators.

3.2 Improved Velocity Estimation in $3D$

Indoor navigation methods often use the device IMU sensor in order to implement a pedometer which detect the device global orientation and count "steps". Yet, such method introduces significant inaccuracy both in the distance measured and in the orientation. Therefore, we use optical flow with plan and range detection [8] in order to estimate the user movement in high sampling rate, this allow us an improved distance approximation and fusing optical features to reduce IMU drifts. Combined with a barometric pressure sensor the vertical speed can be computed - allowing us to detect a floor change.

3.3 Improved Sense Function

The naive and the advanced particle filter algorithm differ mainly by their sense functions. While the naive algorithm simply evaluate the weight of the particles according to their map location (a particle in B or A area), the advanced algorithm performs actual sense to determine how far each particle is from the truth. The sense performed by AR measurement tool detects the front plan region and compare it to the front plan region of each particle. This comparison gives us the ability to re-weight the particles in more precise way.

3.4 Improving Compass Accuracy

The orientation reported by smartphones often suffers from significant errors due to magnetic interference. In order to reduce the orientation inaccuracy related to compass noise and bias, the particle-state may also include additional dimension to estimate the compass original bias, and current drift. Initially, each particle starts with some Gaussian random value of compass bias. During the re-sampling process, each new particle will be assigned a compass related state according to the values of its nearest neighbors, with some minor noise. Each particle will use the smartphone's compass measured data combined with its bias and drift for the move function.

4 Preliminary Results

In this section we present an accuracy evaluation of suggested indoor positioning method. The main result addresses the Microsoft Indoor Localization Competition $IPSN2018$ in which a preliminary version of the suggested algorithm was implemented which allows us a 1-2 meter accuracy in relative complicated $3D$ scenario. We conclude with few implementation remarks regarding the IPIN2018 results in which the suggested method took part and although got to the first place, its accuracy was insufficient.

4.1 Study case: Microsoft Indoor Localization Competition

Since 2014 an annual Indoor Localization Competition is being organized by Microsoft , see [6] for 2014-2017 indoor localization evaluations. In the 2018 competition We have implemented a preliminary version of the suggested particle filter (named "STEPS"). In general, the system was design to improve existing indoor positioning services (such as Google's indoor maps api) with expected accuracy 10-20 meter accuracy to a 1-2 meter ($3D$) accuracy. Overall the system performed as expected allowing a rapid converges of the particle filter - within 10-15 seconds (of 15-20 steps). Figure 3 presents the $2D$ evaluation of the system with respect to the ground truth (GT). Figure 4 shows the convergence nature of particle filter regarding the $2D$ case (when the floor is given).

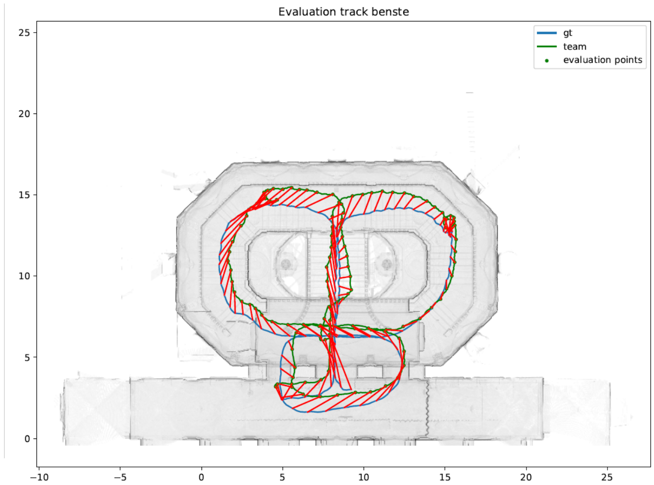

Fig. 3. The "STEPS" path with the $2D$ error with respect to a centimeter level lidar-based ground truth.

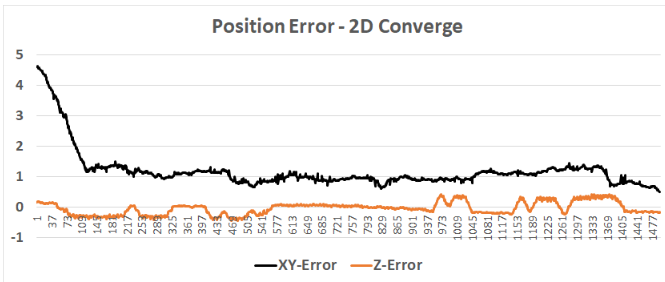

Fig. 4. The Particle Filter $2D$ convergence: Assuming the correct floor is known the horizontal position converge from an error of 4.5 meters to about 1.3 meters within 10 seconds (about 15 steps). During the rest of the test the horizontal error is about 1 meter, while the vertical error is (on average) below half a meter.

4.2 Study case: $IPIN2018$

During September 2018 an indoor positioning competition was held in a large shopping mall at Nantes, France, as part of the $IPIN2018$ Conference on Indoor Positioning [9]. The on-site competition had two tracks (with and without a camera). Naturally, we took part in the "Camera based Positioning" (Track 1). A preliminary version of the algorithm was implemented on a Tango based Android Phone. The initial starting position given to the competitors. The evaluation was conducted over about 70 known waypoints (each with a known $3D$ global position), the path was conducted on 3 floors - with over 1 km long. Along the path our algorithm has used few $GNSS$ momentary positioning (via the mall sky-lights) for global (inaccurate) position. The particle filter localization algorithm was able to maintain a relative [4-12] meter accuracy (7.2 meter on average). The overall evaluation of our Algorithm lead us to first place in the competition.

References

1. Zahid Farid, Rosdiadee Nordin, and Mahamod Ismail. Recent advances in wireless indoor localization techniques and system. *Journal of Computer Networks and Communications*, 2013, 2013.
2. Yanying Gu, Anthony Lo, and Ignas Niemegeers. A survey of indoor positioning systems for wireless personal networks. *IEEE Communications surveys & tutorials*, 11(1):13–32, 2009.
3. Antonio Ramón Jiménez, Fernando Seco, José Carlos Prieto, and Jorge Guevara. Indoor pedestrian navigation using an ins/ekf framework for yaw drift reduction and a foot-mounted imu. In *Positioning Navigation and Communication (WPNC), 2010 7th Workshop on*, pages 135–143. IEEE, 2010.
4. Bernhard Krach and Patrick Roberston. Cascaded estimation architecture for integration of foot-mounted inertial sensors. In *Position, Location and Navigation Symposium, 2008 IEEE/ION*, pages 112–119. IEEE, 2008.
5. Bernhard Krach and Patrick Robertson. Integration of foot-mounted inertial sensors into a bayesian location estimation framework. In *Positioning, Navigation and Communication, 2008. WPNC 2008. 5th Workshop on*, pages 55–61. IEEE, 2008.
6. Dimitrios Lymberopoulos and Jie Liu. The microsoft indoor localization competition: Experiences and lessons learned. *IEEE Signal Processing Magazine*, 34(5):125–140, 2017.
7. Roi Yozevitch and Boaz Ben-Moshe. Advanced particle filter methods. In *Heuristics and Hyper-Heuristics-Principles and Applications*. InTech, 2017.
8. Julien Valentin, Ivan Dryanovski, Joao Afonso, Jose Pascoal, Konstantine Tsotsos, Mira Leung, Mirko Schmidt, Onur Guleryuz, Sameh Khamis, Vladimir Tankovitch, Sean Fanello, Adarsh Kowdle, Shahram Izadi, Christoph Rhemann, Jonathan T. Barron, Neal Wadhwa, Max Dzitsiuk, Michael Schoenberg, Vivek Verma, and Eric Turner. Depth from motion for smartphone ar. volume 37, pages 1–19, 12 2018.
9. *International Conference on Indoor Positioning and Indoor Navigation (IPIN-2018)*, Nantes, France, 2018.