Pedestrian Dead Reckoning Navigation with the Help of A* -Based Routing Graphs in Large Unconstrained Spaces

Fadoua Taia Alaoui, David Betaille, Valerie Renaudin

Hindawi Publishing Corporation
Journal of Wireless Communications and Mobile Computing
Special Issue on Advances in Enabling Technologies for Next Generation 3D Network Localization, Tracking, and Navigation 2017
http://

Pedestrian dead reckoning navigation with the help of A*-based routing graphs in large unconstrained spaces

F. Taia Alaoui, David Betaille, Valerie Renaudin,
IFSTTAR, COSYS, GEOLOC, F-44344 Bouguenais, France

Conflict-of-interest statement: The authors declare that there is no conflict of interest regarding the publication of this paper.

An A*-based routing graph is proposed to assist PDR indoor and outdoor navigation with handheld devices. Measurements are provided by inertial and magnetic sensors together with a GNSS receiver. The novelty of this work lies in providing a realistic motion support that mitigates the absence of obstacles and enables the calibration of the PDR model even in large spaces where GNSS signal is unavailable. This motion support is exploited for both predicting positions and updating them using a particle filter. The navigation network is used to correct for the gyro drift, to adjust the step length model and to assess heading misalignment between the pedestrian's walking direction and the pointing direction of the handheld device. Several data sets have been tested and results show that the proposed model ensures a seamless transition between outdoor and indoor environments and improves the positioning accuracy. The drift is almost cancelled thanks to heading correction in contrast with a drift of 8% for the non-aided PDR approach. The mean error of filtered positions ranges from 3 to 5m.

Keywords: A* path-planning; Pedestrian Dead Reckoning; Map-aided; Particle filter.

1. Introduction

Pedestrian Dead Reckoning (PDR) is widely adopted in the field of pedestrian navigation with handheld devices. It is particularly adapted to smartphone-based localization as inertial sensors can be designed in a MEMS (Micro-Electro Mechanical Sensors) technology, enabling them to be embedded in lightweight devices. Unlike GNSS receivers, inertial sensors are especially useful indoors as they allow standalone localization without sky visibility. Yet, due to gyro drift and step detection limitations, additional information is required to assist the PDR positioning process. For foot-mounted sensors, zero velocity update (ZUPT) calibration is exploited to adjust the positioning parameters by detecting stance phases within the gait cycle (static phase). Though, this calibration is not possible with handheld devices because of free hand motion and an increased difficulty to detect the stance phase. Outdoors, PDR can still be aided by GNSS [1], but this is not feasible indoors because of signal unavailability and further measurements are needed. These could be provided by radio beacons or visual information. The first approach requires infrastructure deployment and training [2], while the second necessitates a camera and further image processing for feature recognition [3]. A third possibility is to constrain the pedestrian's position using map information. Two main paradigms can be retained from previous work. Either walkable space is given by 2D maps delimited by obstacles [4], or it is given by a routing graph network that transforms the positioning process into a piecewise 1D model [5]. In the first case, space is better explored but the map is not exploited further than for detecting static obstacles (e.g. walls). This means that no calibration is performed unless an obstacle is hit. On the contrary, routing graphs are much more constraining because the motion model is directly given by the graph network. Hence, their use has greater impact on the shape and accuracy of the trajectory and they have to be realistic enough to limit positioning errors.

This paper focuses on routing graph-assisted PDR. In fact, routing graphs involve a simple motion model that allows both obstacle avoidance and the calibration of walking directions within straight line travels [5]. Their use can even be extended to calibrating the step length model as reported in our previous work [6]. Though, two major drawbacks make their use quite impractical and sometimes ineffective. First, there is no ubiquitous process that generates combined indoor-outdoor pedestrian graph networks. Their construction can be time-consuming and sometimes inadequate with pedestrian motion. While the graph is expected to counterbalance PDR limitations, it may introduce additional errors due to inconsistencies with real displacement. Second, they do not handle "pseudo-random" trajectories within obstacle-free areas and during the transition between outdoor and indoor spaces as the freedom of motion increases. In these cases, routing graphs become inefficient as no motion assumptions can be relied on to design a realistic network. Grid-based models are well-suited for exploring large spaces [7]. Yet, no calibration is possible with this approach. This even leads to overestimate the travel distance because linearity of movement is lost due to the grid structure. As a result, the PDR process is no longer assisted and the basic issues of gyro drift and uncalibrated travel distance not solved.

A solution to the vulnerability of routing graph assisted PDR navigation in obstacle free spaces is investigated in this study. The proposal is to make use of A* algorithm, which is commonly used for path planning within virtual worlds or for vehicle route guidance, in order to design a realistic routing graph in obstacle-free spaces. The graph is constructed on the basis of a set of waypoints that are crucial for pedestrian navigation. In fact, the latter are expected to balance the absence of geometrical constraints by providing strategic locations between which path computation with A* algorithm would be relevant. Therefore, the first contribution of this

study is to mitigate the drift in the PDR approach even in large spaces. Moreover, the issue of accumulating errors during the transition between indoor and outdoor spaces is addressed by improving the routing graph relevance. This is accomplished through the computation of the likely paths a pedestrian may take from outdoor strategic locations to reach buildings entrance doors. In addition, the choice of A* algorithm is well-suited for handling different mobility profiles (e.g. personal disease that impacts the path of the target user) so that the approach can be customized for the specific needs of each pedestrian. Indeed, this is ensured thanks to a user-relative weighting of the map which is directly involved in the graph construction. This is because A* algorithm computes optimal paths according to walkability rates given by a map of navigation (e.g. walkability rate according to slopes, pavement, etc.). For instance, one would take the shortest path to cross an open area, whereas a disabled person may follow another path such as walking near walls or dedicated tracks.

2. A* pathfinding within navigation meshes

2.1. Overview of Navigation Mesh generation. A navigation mesh (NavMesh) is a set of several 2D or 3D polygons reachable by some given user [8]. This structure can be obtained automatically using GIS software. Others make use of RECAST [9], an open source library within the community of virtual world designers. Several studies were conducted on how to construct navigation meshes from raw 3D models [10], [11]. NavMesh construction allows assigning weights that translate the polygon walkability rates according to some given characteristics (slope, type = {stairs, flat ground…}) and to the target user's mobility profile (ability to walk, to climb, to take the stairs, etc.). The weighting of the NavMesh is pretty important for considering the user's mobility profile in order to compute suitable routing paths. DETOUR [12] is an example of open source software that allows A* pathfinding calculations on the basis of a NavMesh.

2.2. A shopping mall NavMesh generation. The proposed method was applied to a shopping mall (Fig. 1). The latter is mapped in Google Maps with the names of stores, the mall principal gates and the set of walkable zones. We used QGIS 2.12.1 software to extract the map in the form of a raster with an 11cm resolution. Walkable zones were extracted by digitalizing the map and eliminating obstacles. Delaunay triangulation was used to create the NavMesh. All polygons were assigned the same weight as all experiments are conducted by healthy persons. This implies that A* computed paths will minimize distance and may result in a shape that can be obtained with different approaches [10], [13].

2.3. A pathfinding algorithm.* A* is an improved version of the Dijkstra algorithm [14] . The latter aims at calculating the optimal path between two points according to one given cost function. Optimization applies to the travel distance, the time of travel, the expended energy, etc. This adaptability allows dealing with the issue of reduced mobility due to a disease or handicap. The A* optimization process is a discrete search scheme where space is modelled by a grid composed of cells. Each cell is explored according to an adjacency graph that models connections within the grid. Once visited, a cell is stored together with its assigned cost until the target cell is reached. The optimal path is given by the sequence of cells for which the sum of costs is minimal. A* is an extension to the principle of Dijkstra that calculates the cost of visited cells according to their distance from the starting point (which is the same as the Dijkstra cost) but also to their assumed distance to the target-cell in a heuristic approach. It is the a priori cost-to-go assumed lower than the actual (unknown) cost from the current cell to the target cell. A* is faster than Dijkstra algorithm due to the sorted exploration of cells according to their costs-to-go. The A* pathfinding scheme is applicable to any rasterized NavMesh where polygon weights are inherited by the raster cells.

Fig. 1: Walkable zones have been meshed (green polygons). Blank spaces are also walkable but the NavMesh is restricted to our zone of interest. The gray blocks are the stores considered as obstacles.

3. A* based routing graph generation

3.1. Routing graph generation on the basis of waypoints. A waypoint is a punctual element that intervenes significantly in the pathfinding process. The significance of a waypoint is related to the role it plays in the pedestrian's decision making during her/his travel. Two situations are worth considering. The first is when the pedestrian intends to reach a known destination within a familiar environment. In this case, it is obvious that (s)he follows the itinerary best suited to her/his mobility profile. Generally it is the shortest one but it could be any path depending on the cost-function that defines the user's mobility profile. This situation is handled by A* algorithm using a weighted NavMesh. The target destination is known if it is visible from the pedestrian's current position, for example if the pedestrian is located at a graph node that is visible from her/his target node. The second situation is when the pedestrian's destination is invisible due to obstacles. In this case, waypoints are built in order to discretize the possible target destinations which may be either final or intermediate destinations.

To better understand the implementation of waypoints, space has to be considered in relation to human spatial cognition. Indeed, walkable areas within a building floor could be corridors, rooms or halls. Outdoors, walkable areas are the space comprising sidewalks, footpaths, squares, car parks, etc. (Roads belong to the drivable areas.). This intuitive classification of space components allows defining scenes and extends what is called decision scenes to outdoor space and corridors. Decision scenes have been defined previously in [15] as the places that "can be entered and left and [are] physically bounded by buildings and other solid obstacles that prevent movement". The extension of decision scenes to corridors and footpaths in this paper is motivated by the fact that these elements have borders (physical borders for corridors and geometrical borders for footpaths) and that they can be accessed and left through specific points. For example

two corridors are two separate scenes. They can be accessed and left through their intersection which is a building corner; a corridor and hall are also distinct scenes between which traffic flow is possible through their intersection. Two elements are important for pedestrian travel inside and between scenes. The first is related to the isovist, which is visible space from the pedestrian's current position. The isovist is mainly influenced by the presence of obstacles that have an impact on both visibility and the pedestrian's trajectory [16]. Second element is the set of portals that allow flows of pedestrians from scene to scene [15] such as corridor extremities, building gates, store entries or footpath extremities. Indeed, key elements in the pedestrian wayfinding behavior depend mainly on portals and obstacle borders (which determine the isovist). They materialize the fact that visibility and purpose (destination) are the parameters that give a shape to one's trajectory. In this study, the set of waypoints is composed of portals (corridor extremities, footpath extremities, outdoor/indoor doors) and obstacle corners. The graph construction based on waypoints is obviously more realistic than kernel-based methods where scenes are modelled by their centers [17], implying erroneously that the pedestrian walks systematically through the center of the scene. Besides, the structure of the graph is not entirely determined by the geometry of space but handles behavioral-based paths generated according to waypoints and reflecting pedestrian travel strategies.

Once waypoints have been constructed, they can be exported in a database that will assist the pathfinding process. Each pair of waypoints are related by a set of itineraries that are generated with A* algorithm and are part of the final routing graph. Multi-hypotheses motion is handled as in classical routing graphs by exploring several paths and keeping the ones that are best adapted to IMU measurements. The graph structure is stored in the form of a GIS database that contains the graph segments identifiers, their extremity nodes coordinates, their length and their connections within the graph in a node-connectivity approach. Each edge of the graph is oriented and has an entry node A and an exit node B. The node connectivity design is involved in the motion model used for filtering.

3.2. A path planning for seamless transition between outdoor and indoor spaces and within obstacle-free areas.* Previous work demonstrated that indoor layout-based graphs are efficient for enhancing the PDR localization within geometrically constrained areas [5], [6]. Hence, the proposed method is applied only to the area that precedes the mall principal entrance as well as in the obstacle-free hall (*Fig. 2*). The latter represent the critical GNSS-deprived places where map information fails to provide a calibration to the PDR positioning process, emphasizing the main issue being addressed in this paper. Outdoors, the routing graph is constructed according to the geometry of sidewalks, crosswalks and footpaths.

Fig. 3 shows the generated routing graph within the test area. Three waypoints materialize the building entrance gate (drawn is black in *Fig. 2*). The focus is made on the waypoint where red paths intersect. It represents the left side of the mall entrance. Red paths are the A*-calculated itineraries relating the left side of the gate to other places of interest such as stores entries or the building's main corridors. The latter are modelled by a series of straight lines according to corridors main directions. Three paths relate the outdoor to the indoor space through the left side of the mall entrance.

This representation shows that straight line travels are privileged for practical and fast displacement. They are also more realistic in regards with pedestrian motion. In fact, some hypotheses are eliminated such as walking towards walls or making a series of turns to attend a place that can be reached straight ahead. Moreover, this provides a measurement of walking directions given by the graph segments in obstacle-free zones, which would be impossible if a grid-based approach had been applied, requiring additional information such as radio beacons signal to compensate for the PDR errors.

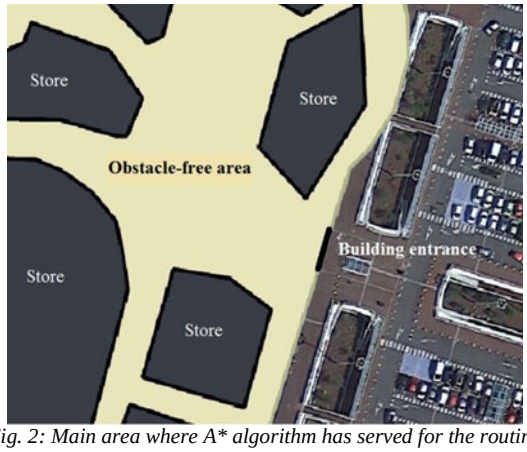


Fig. 2: Main area where A algorithm has served for the routing graph construction*

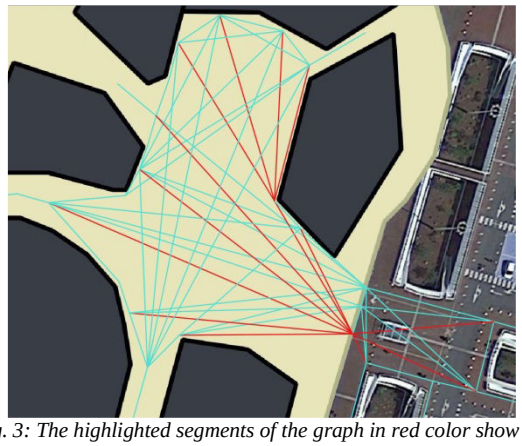


Fig. 3: The highlighted segments of the graph in red color show the A itineraries relating a portal waypoint (left side of the building entrance door) to different destinations. The latter are either extremities of corridors or footpaths, obstacle corners, or store entries.*

4. IMU fusion with GNSS and the routing graph with a particle filter

4.1. Step detection and heading calculation. The step detection is realized after motion classification according to [18] by detecting peaks on the acceleration signal using an adaptive threshold algorithm [19]. According to the same reference, a generic model is used to estimate the step length. This model relies on a set of three parameters trained on 10 subjects and is given by

$$s = k.(h(af + b) + c), \quad (1)$$

where s is the step length, h is the user's height, f is the step frequency, $\{a,b,c\}$ the generic model parameters and k a scale factor that is expected to calibrate the model on the pedestrian.

Headings have been calculated with MAGYQ attitude estimation filter [20] that fuses signals from a tri-axis accelerometer, a tri-axis gyroscope and a tri-axis magnetometer. Heading calculation considers different carrying modes such as the swinging mode or the texting mode. The device orientation in 3D-space is then obtained and the yaw angle deduced, giving the orientation of the device relative to the true North.

4.2. Calibration of the PDR parameters using GNSS positions and the routing graph. The PDR parameters to be calibrated are the step length and headings. The step length model needs to be adjusted to each user as the model parameters are dependent on the pedestrian's physiological features and gait cycle, whereas headings are potentially misaligned with the actual walking direction because of gyro drift and the device carrying mode. In fact, the device may be oriented towards a direction which is different from the walking direction. These errors are compensated by fusing the IMU with GNSS and the routing graph. The fusion is realized thanks to a particle filter that models the state (vector of unknown variables) by a set of particles (a set of sampled state vectors). The state vector contains necessary variables for determining the user's position and is presented in detail in section *4.3*.

The step length model is adjusted to the pedestrian using both the graph and GNSS positions when available. In fact, the graph allows keeping the positions on plausible path(s). This directly impacts the travel distance. Besides, GNSS decreases the particles dispersion by bringing them next to the GNSS position. On the other hand, walking directions are given by path headings. They are particularly reliable indoors as the paths are calculated by the A*-algorithm or are directly given by the corridors main directions. The routing graph-derived walking direction is then compared to the IMU pointing direction and the most likely path is selected. The difference between both headings gives the IMU angular misalignment with the pedestrian's walking direction.

The step length model is calibrated using a scale factor k. The latter is assumed constant as it depends mainly on the pedestrian's physiological parameters. It is modelled by a Gaussian variable with 1m mean and low variations as the step length is restricted to realistic values comprised between (0.60m – 1m). k is expected to converge within straight line travels where an optimal calibration is guaranteed.

Heading misalignment β is the angular difference between the actual walking direction and the pointing direction of the IMU. It is time-dependent because it varies with hand motion. Uniformly distributed samples of heading misalignment values are drawn with variations up to 30°.

4.3. Model implementation with a particle filter. The state vector is given by

$$\{Id,\ Dp,\ \delta,\ k,\ \beta\ ,\ w\}_t^p \qquad (2)$$

where Id is the edge identifier in the graph database, Dp is the curvilinear abscissa, δ represents the travel direction, which can be 1 if the edge is walked forward (from A to B) and -1 otherwise (from B to A), k is the scale factor that calibrates the step length model, β is the misalignment between the walking direction and the pointing direction of the handheld device, w is the particle weight, p and t refer respectively to the particle and time.

Each particle is moved along the current path according to the travel distance given by the step length. The particle filter introduces white noise on each parameter to spread the particles and optimize the graph exploration. The particle position is determined by the identifier of the graph edge and its curvilinear abscissa. If the latter is above the edge length, one of its connected edges will be explored. The transition model is

$$DU = (k \cdot s) \cdot \delta . \alpha + n_G + Dp_{t-1} \qquad (3)$$

where DU is compared to the current edge length L_{Id} to compute the curvilinear abscissa Dp (Fig.4), s is the step length, and n_G is the modeling error represented by a Gaussian noise. $\alpha \in \{0,1,2\}$ and determines the number of steps to be made by one particle. It allows detecting over/mis-detected steps.

Fig.4 gives the transition test process that allows computing the curvilinear abscissa *Dp* when an exit node B is exceeded (i.e. DU > 0). t is the time index. BA/ BB connectivity translates the search in the graph database in order to determine if the edge being explored is connected to the exceeded one via an A or a B node. The same logic is adopted if an entry node A is exceeded (i.e. DU <0).

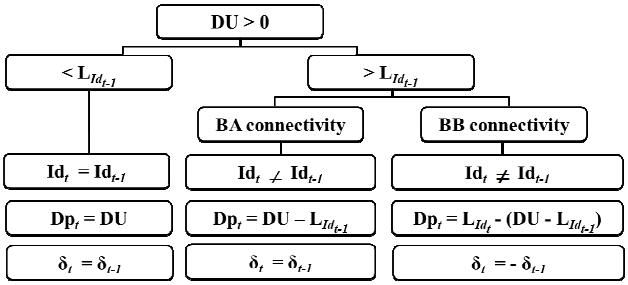


Fig. 4: Dp computation according to DU

When GNSS positions are accurate enough, they are used together with the graph headings in order to calibrate both the step length model and PDR headings. The graph headings provide a measurement of walking directions which are compared to PDR headings, taking into account potential angular misalignment. This is performed according to

$$w_t = w_{t-1} . \exp\left(-\frac{1}{2} . \left(\frac{|\theta_{PDR} - (\theta_{Id} + \beta)|^2}{(\sigma_{\theta_{Id}}{}^2 + \sigma_{\theta_{PDR}}{}^2)} + \Delta' \sum{}^{-1} \Delta \right) \right) \qquad (4)$$

$$\Delta = \begin{bmatrix} E_P - E_{GPS} \\ N_P - N_{GPS} \end{bmatrix} \qquad (5)$$

$$\Sigma^{-1} = \begin{bmatrix} 1/(\sigma_{E_{GPS}}{}^2 + \sigma_E^{map\,2}) & 0 \\ 0 & 1/(\sigma_{N_{GPS}}{}^2 + \sigma_N^{map\,2}) \end{bmatrix} \qquad (6)$$

where w_t is the particle weight, E_P and E_{GPS} are respectively the predicted East coordinate of a particle and its corresponding GPS observation. N_P and N_{GPS} are the same parameters along the North direction. $\sum^{-1}$ is the weighting matrix considering the graph and GPS position accuracies.

Indoors, some GNSS positions are still available but the latter are generally unreliable and are rejected. The rejection test is made by thresholding the signal to noise ratio (SNR) and the horizontal dilution of precision (HDOP). Only the

graph headings are exploited in the weighting process. The weighting equation indoors is

$$w_t = w_{t-1}.\exp\left(-\frac{1}{2}\times\frac{|\theta_{PDR} - (\theta_{Id} + \beta)|^2}{(\sigma_{\theta_{Id}}{}^2 + \sigma_{\theta_{PDR}}{}^2)}\right) \quad (7)$$

where θ_{PDR} is the walking direction estimated by the PDR algorithms, θ_{Id} is the predicted heading of the current path, β is the heading misalignment between the path direction and the PDR estimate, $\sigma_{\theta_{Id}}$ is the standard deviation of the path heading, and $\sigma_{\theta_{PDR}}$ is the standard deviation of the PDR walking direction estimate.

Once the particles have been weighted, some of them are assigned low weights and become useless in the process. Resampling is performed to duplicate the particles with high weights and delete the others. The particles amount is kept constant and their weights equal after each update in order to explore enough hypotheses of motion.

5. Experiments

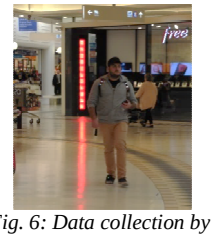
5.1. Data collection with a HSGNSS and IMU in hand. Three healthy volunteers (2 men, 1 woman) collected data with *ULISS* (Ubiquitous Localization unit with Inertial Sensors and Satellites) device (*Fig. 5*) held in hand (*Fig. 6*). Data were collected in both outdoor and indoor environments for two different device carrying modes. Two acquisitions were made in a texting mode and one in both swinging and texting modes. The average duration of acquisitions was about 14 minutes and the walking distance for each trial almost 1.5 km.

Different scenarios were chosen to perform the experiments. These will be explained in detail in section *6.1* where the text is enhanced by figures.

5.2. Input data. ULISS device [21] comprises a 9 degrees of freedom inertial mobile unit, a high sensitivity GNSS (HSGNSS) receiver and antenna, a memory card and a battery. It delivers measurements that are time-stamped in GPS time. Inertial sensors and magnetometers provide measurements at a 200 Hz frequency.

The HSGNSS receiver operates in a standalone mode and delivers positions in real time at a 5 Hz frequency. Delivered positions are time-stamped in GPS time and have metric accuracies ranging from 2m up to 10m near buildings and tree shades. In this work, GNSS positions were interpolated at the step frequency in order to be fused with the PDR estimates of headings and step lengths.

Fig. 5: Ubiquitous Localization Unit with Inertial Sensors and Satellites



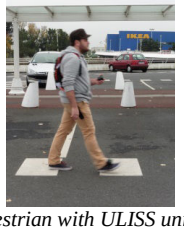


Fig. 6: Data collection by a pedestrian with ULISS unit in hand

5.3. Reference trajectories. Besides collecting data with *ULISS* device in hand, all volunteers were equipped with an independent GNSS receiver carried in their backpacks and a small antenna attached to their caps. GNSS measurements were then post-processed in order to calculate reference trajectories by differential GNSS. This was performed using RTKLIB 2.4.2p12 software [22]. Measurements from the embarked GNSS receiver and from a nearby base station were used to perform relative GNSS positioning. Obtained positions had decimetric accuracies up to several meters near buildings and other elevated features. Afterwards, thresholding was applied to three parameters in order to reject some outlier position estimates. These parameters are first the number of visible satellites, second, the ratio factor of ambiguity resolution and final, the horizontal dilution of precision. The resulting positions had precisions below 1m and were adequate for accuracy assessment in this work.

6. Results

6.1. Trajectory analysis. Fig. 7, 8 and 9 show the estimated trajectories (red) with the A*-generated routing graph and the particle filter described in section 4.3. The blue pattern corresponds to the PDR trajectory, while the green one is the reference trajectory obtained by differential GNSS.

Fig. 7 and *8* correspond to acquisitions performed in the texting mode. The starting point for both acquisitions lies on the top right extremity of the trajectories. Both subjects made a closed loop around the building, with an intermediate outdoor travel (Bottom side of the figures) before reaching the starting point back.

For both acquisitions, the travel distance seems to be overestimated. Yet, heading determination is more accurate for the first acquisition as the shape of the trajectory is more faithful to the building structure. The drift is higher for the second dataset and can be visually observed at the end of the PDR trajectory.

Where the drift is most important, the PDR trajectory presents major inconsistencies with the map. According the above figures, the drift has been corrected as the estimated trajectories are more compliant with the building structure and with footpaths in outdoor space. This has been achieved thanks to the proposed particle filter and to an increased conformity with pedestrian motion; demonstrating that the positioning accuracy for the texting scenario has been significantly enhanced using our approach.

Fig. 7: Estimated trajectory (in red). The Blue trajectory gives the PDR position estimates and the green one gives the reference trajectory (Only Texting)

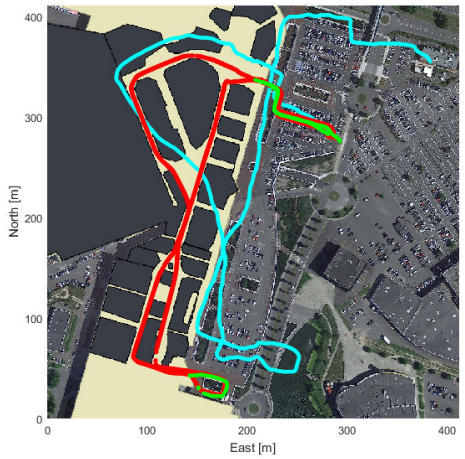


Fig. 8: Trajectories for the second dataset (Only texting)

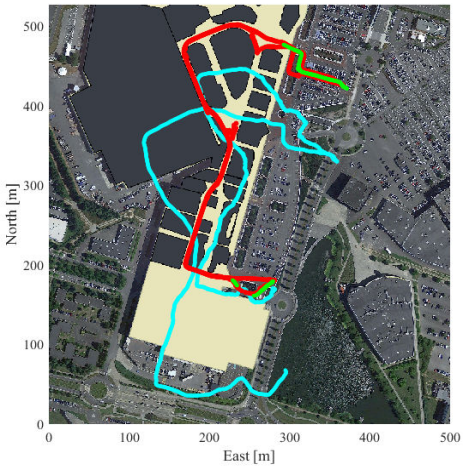


Fig. 9: Trajectories for the third dataset (Swinging, texting)

Fig. 9 corresponds to data collected for both the swinging and texting modes. The acquisition started at the bottom of the figure where the reference and the filtered trajectories overlap. Starting from this position, the subject entered the building and then went outside through the North-East building entrance. This travel was made in the swinging mode. The top right extremity of the trajectory underlines a U-turn before the subject entered the building back to reach the starting point. This part of the travel was performed in the texting mode.

Unlike the two first acquisitions, there is a gap in the PDR trajectory because the subject did not go around the building as the two first subjects did. This gap is retrieved in the filtered trajectory.

Obviously, this scenario implies a less accurate heading determination and even an alteration in the travel distance estimation. In fact, the walking distance is underestimated for the swinging mode (First part of the travel until the subject reached the outdoor) and overestimated for the texting mode (Second part of the travel until the ending point). These errors can be noticed in the PDR trajectory.

The filtered trajectory shows that the drift has been corrected, resulting in a shape that is more compliant with the map and with the reference trajectory outdoors. Yet, the positioning accuracy seems to be decreased as compared with the two first datasets. Following section discusses the accuracy of estimated trajectories.

6.2. Error computation. In order to assess the accuracy of our positioning method, filtered positions were compared to the reference positions interpolated at the step frequency. The average plane error ranges from 4 to 5 meters for the three datasets (*Fig. 10*). Accuracy is dependent on the quality of the PDR trajectory. Therefore, computed errors are more important for the second dataset considering the texting scenario. For the third acquisition that includes both swinging and texting, the accuracy is significantly decreased and more outliers (precisions above 15m) are detected due to mismatching errors (i.e. choosing wrong edges of the graph). These errors are mainly due to uncalibrated PDR parameters and are discussed in the following sections.

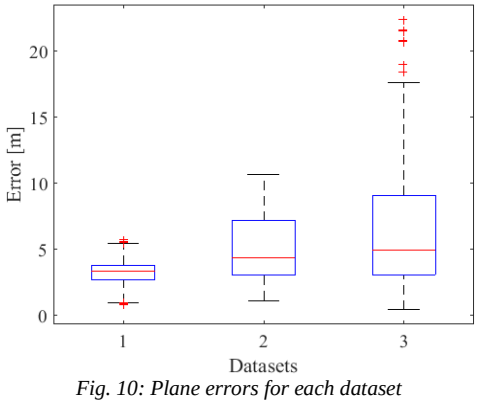


Fig. 10: Plane errors for each dataset

6.3. Heading misalignment estimation. The figures below show the estimated heading misalignment values for each dataset. For the first trial (*Fig. 11*), angular misalignment is comprised between -10° and +10° and varies around an approximate mean value of 0°. According to this distribution, the angular difference between walking directions and the pointing direction of the device is minimal. Hence, applied corrections compensate only for the gyro drift; which is rather logical regarding the texting mode scenario.

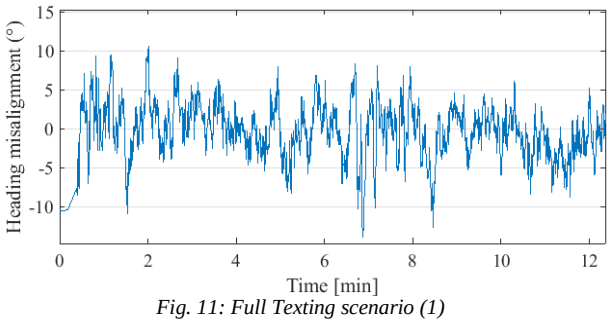


Fig. 11: Full Texting scenario (1)

For the second dataset (*Fig. 12*), estimated heading misalignment values are comprised between -15° and +12°. They are not equally distributed around 0° (e.g. between the 1st and 2nd minutes the mean value is over 5°). From this analysis, non-negligible hand motion can be assumed even if the subject intended to perform the experiment in the texting mode.

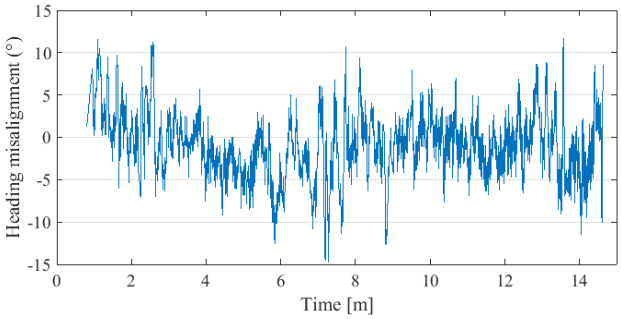


Fig. 12: Full texting scenario (2)

Fig. 13 gives the estimated heading misalignment for the third trial. The first part (until 9.5min) of the travel was performed in the swinging mode. Hence, the estimated values vary significantly (-15° to +20°). Afterwards, oscillations between ± 5° over a mean value of 0° are retrieved, reflecting the texting mode of the acquisition.

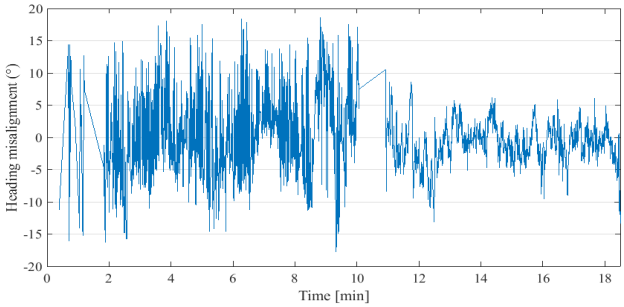


Fig. 13: Swinging-Texting scenario

6.4. Step length model calibration. In this paper, a scale factor was introduced on the step length in order to calibrate the walking distance. Though, due to degraded GNSS signal and to short walking periods in outdoor space, the scale factor was not calibrated. In fact, regular and accurate GNSS positions are needed through straight line travels as cited in our introduction in order to calibrate the walking distance. These conditions were not verified during our experiments.

As a result, only corrected headings were relied on in the selection process. Therefore, distance calibration occurred only at junctions of the graph when a change in heading was detected. This explains the fact that the accuracy of filtered trajectories is still enhanced and compliance with the map improved.

Though, the uncalibrated walking distances caused some mismatching errors because direction change was detected too late or too early in the process. This can be noticed in the Northern part of *Fig. 9*. Indeed, while the pedestrian was intending to exit the building, the filtered trajectory indicates that he was walking towards a corridor. This happened for two reasons. First, the real trajectory is quite unusual in terms of pedestrian behavior. In fact, there is a change in heading (observed in the PDR trajectory) that is independent from space configuration, invalidating the assumptions that allowed constructing our navigation network. Second, the uncalibrated walking distance prevented the particles from reaching outdoor space at the right time. Another mismatching error occurred at the middle of the building (Between 300m and 400m North) because direction change was detected too late due to uncalibrated walking distance. Later on, the particle filter corrected for this error and converged over the right corridor thanks to the particle dispersion over the graph and to the multi-hypothesis approach.

7. Conclusions

A map-aided PDR approach where a routing graph is used as motion model has been proposed. Main contribution of this paper is A* algorithm adaptation to elaborate a pedestrian network that is capable of cancelling the gyro drift and the misalignment between the device orientation and the walking direction even in large spaces. These are GNSS-deprived and obstacle free areas where the limitations of map-aided PDR algorithms are most important. In fact, widespread map-aided PDR approaches do not compensate for these errors when pedestrian motion is unconstrained, mainly during the transition between outdoor and indoor spaces and when obstacles are absent. The A*-based routing graph mitigates the lack of obstacles thanks to a set of waypoints implemented according to human spatial cognition and to a weighted navigation mesh. This allows building a realistic motion model that meets the requirements of map-aided localization. Indeed, the proposed routing graph is well exploited because it gives prior knowledge about the pedestrian's destination and provides reliable measurements of walking directions. Results show that it is adequate for a seamless transition between outdoor and indoor environments and for enhancing the positioning accuracy even in large spaces. Achieved accuracies range from 3 to 5 meters and the drift is almost cancelled with the help of the routing graph. Though, some mismatching errors due to uncalibrated walking distance, especially while carrying the device in the swinging mode, might induce important positioning errors. Indeed, proper conditions of sky visibility and sufficient period of outdoor walking are prerequisite for the step length calibration before the pedestrian reaches indoor space.

8. Acknowledgements

This research work was done as part of the HappyHand (2015-2018) project which is funded by the 19th "Fonds Unique Interministériel" (French national R&D fundings).

9. References

[1] J. B. Bancroft and G. Lachapelle, “Data Fusion Algorithms for Multiple Inertial Measurement Units,” *Sensors*, Jun-2011.

[2] Y. Hu, L. Sheng, and S. Zhang, “Design of Continuous Indoor Navigation System Based On INS and Wifi,” in *Sensors, Measurement and Intelligent Materials, Pts 1-4*, vol. 303–306, Y. H. KIm and P. Yarlagadda, Eds. Stafa-Zurich: Trans Tech Publications Ltd, 2013, pp. 2046–2049.

[3] T. Lee *et al.*, “An Indoor Positioning System Using Vision aided Advanced PDR technology without image DB and with motion recognition,” *Proc. Ion 2013 Pac. Pnt Meet.*, pp. 490–497, 2013.

[4] P. Hafner, T. Moder, M. Wieser, and T. Bernoulli, “Evaluation of Smartphone-based Indoor Positioning Using Different Bayes Filters,” in *2013 International Conference on Indoor Positioning and Indoor Navigation (ipin)*, New York: Ieee, 2013.

[5] T. Willemsen, F. Keller, and H. Sternberg, “A topological approach with MEMS in smartphones based on routing-graph,” in *Indoor Positioning and Indoor Navigation (IPIN), 2015 International Conference on*, 2015, pp. 1–6.

[6] F. T. Alaoui, D. Betaille, and V. Renaudin, “A multi-hypothesis particle filtering approach for pedestrian dead reckoning,” in *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2016, pp. 1–8.

[7] T. Fetzer, F. Ebner, F. Deinzer, L. K?ping, and M. Grzegorzek, “On Monte Carlo smoothing in multi sensor indoor localisation,” in *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2016, pp. 1–8.

[8] M. R. F. Mendonça, H. S. Bernardino, and R. F. Neto, “Stealthy Path Planning Using Navigation Meshes,” in *2015 Brazilian Conference on Intelligent Systems (BRACIS)*, 2015, pp. 31–36.

[9] “Recast: Detail API documentation for the members declared in Recast.h… · recastnavigation/recastnavigation@6f5c9f9,” *GitHub*. [Online]. Available: https://github.com/recastnavigation/recastnavigation/commit/6f5c9f9b82418efc44b85974e604095b95354ada. [Accessed: 28-Jul-2016].

[10] I. Afyouni, C. Ray, and C. Claramunt, “Spatial models for context-aware indoor navigation systems: A survey,” *JOURNAL OF SPATIAL INFORMATION SCIENCE*, pp. 85–123, May-2012.

[11] W. van Toll, A. F. Cook, and R. Geraerts, “Navigation meshes for realistic multi-layered environments,” in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 3526–3532.

[12] “DETOUR,” *GitHub*. [Online]. Available: https://github.com/recastnavigation/recastnavigation/tree/master/Detour/Source. [Accessed: 08-May-2017].

[13] F. Mortari, S. Zlatanova, L. Liu, and E. Clementini, “(IGNM): A novel approach for deriving connectivity graphs for indoor navigation,” *ISPRS Tech. Comm. IV Symp. ISPRS Ann. - Vol. II-4 Suzhou China 14-16 May 2014*, 2014.

[14] N. O. Eraghi, F. López-Colino, A. de Castro, and J. Garrido, “Path length comparison in grid maps of planning algorithms: HCTNav, A #x2217; and Dijkstra,” in *Design of Circuits and Integrated Systems*, 2014, pp. 1–6.

[15] C. Gaisbauer and A. U. Frank, “Wayfinding Model For Pedestrian Navigation,” presented at the 11th AGILE International Conference on Geographic Information Science 2008, University of Girona, Spain, 2008.

[16] N. Victor, “évaluation des déplacements piétons quotidiens,” Université de Lyon, 2016.

[17] L. Yang and M. Worboys, “Generation of navigation graphs for indoor space,” *Int. J. Geogr. Inf. Sci.*, vol. 29, no. 10, pp. 1737–1756, Oct. 2015.

[18] M. Susi, V. Renaudin, and G. Lachapelle, “Motion Mode Recognition and Step Detection Algorithms for Mobile Phone Users,” *Sensors*, vol. 13, no. 2, pp. 1539–1562, Jan. 2013.

[19] V. Renaudin, M. Susi, and G. Lachapelle, “Step Length Estimation Using Handheld Inertial Sensors,” *Sensors*, vol. 12, no. 7, pp. 8507–8525, Jul. 2012.

[20] V. Renaudin and C. Combettes, “Magnetic, Acceleration Fields and Gyroscope Quaternion (MAGYQ)-Based Attitude Estimation with Smartphone Sensors for Indoor Pedestrian Navigation,” *Sensors*, vol. 14, no. 12, pp. 22864–22890, Dec. 2014.

[21] “ULISS.” [Online]. Available: http://www.ifsttar-geoloc.fr/index.php/en/equipment/44-uliss. [Accessed: 28-Apr-2017].

[22] “GitHub - tomojitakasu/RTKLIB_bin.” [Online]. Available: https://github.com/tomojitakasu/RTKLIB_bin. [Accessed: 01-Mar-2017].