TREKIE - Ubiquitous Indoor Localization with Trajectory REconstruction Based on Knowledge Inferred from Environment

Attila Török[1(✉)], András Nagy[2], and Imre Kálomista[1]

[1] Bay Zoltán Nonprofit Ltd. for Applied Research, Fehérvári út 130, Budapest 1116, Hungary
attila.torok@bayzoltan.hu

[2] University of Pannonia, Egyetem út 10, Veszprém 8200, Hungary

Abstract. The recent indoor localization techniques use inertial sensors for position estimations in order to obtain a certain degree of freedom from infrastructure based solutions. Unfortunately, this dependency cannot be completely eliminated due to the cumulative errors induced in the localization process. While many methods are designed to reduce the required number of reference points, completely infrastructure independent solutions are still missing. In this paper we extend the approach of DREAR, a mobile-based context-aware indoor localization framework. DREAR exploits the ability to recognize certain human motion patterns with a smartphone, representing activities related to walking, climbing stairs, taking escalators, etc. This allows the detection of corridors, staircases and escalators, knowledge which can be used to create building interior related reference points. Based on these a scenario specific context interpreter controls the localization process and provides position refinement for the elimination of the cumulated errors. However, due to the cumulated errors in the trajectory, in case of neighbouring reference points with similar characteristics an adequate distinction cannot made, based solely on the detected activities, which leads to wrong reference point associations and erroneous location refinements. Thus, we extended DREAR with a trajectory reconstruction algorithm, to cope with these errors and their effect on the outcome of reference point selection. The proposed solution is evaluated in a complex subway scenario, its performance is analysed focusing on path reconstructions and the benefits of using specific context-related information. The results are promising, the proposed algorithm presents further improvements relative to the performance of DREAR, providing an excellent localization and path reconstruction solution.

1 Introduction

To obtain a certain degree of freedom from localization infrastructures (e.g. radiofrequency - RF - reference points) novel indoor positioning solutions leverage smartphones' built-in sensors in dead-reckoning (DR) algorithms to reconstruct the movement path of the navigated person [1]. Considering conventional

M. Younas et al. (Eds.): MobiWis 2015, LNCS 9228, pp. 15–26, 2015.
DOI: 10.1007/978-3-319-23144-0_2

use cases (localization in shopping centres, office buildings) the current solutions [2] [3] can provide an adequate level of adaptability regarding the configuration changes of the RF reference points. Nevertheless, there are scenarios, certain premises (e.g. subway systems), where the placement of any kind of reference points is beyond possibility, due to legal or investment issues, exposure to vandalism, or else their usability is simply questionable because of the effect of crowd dynamics on the system's performance. It is also hard to build truly ubiquitous indoor Location Based Applications (LBA) if the used localization solution leads to vendor lock-in due to a specific technology (e.g. iBeacon). Therefore, localization infrastructure dependency must be further relaxed and completely infrastructure-free solutions have to be introduced for well-defined use cases to build future ubiquitous indoor LBAs.

In [4] a novel approach for mobile-based infrastructure-free indoor localization services was proposed (called DREAR), aimed to provide a context-aware solution for ubiquitous indoor LBAs. The goal of DREAR was to build a localization system, which does not require any kind of localization infrastructure (e.g. previously deployed RF, visual or acoustic beacons) and where the user's active involvement for finding certain reference points can be also omitted. Therefore, human movement behaviour analysis and activity recognition algorithms were employed in a context-aware middleware, to control the localization and position refinement process. A context recognition layer was designed to acquire information from user movement events (detected activities, trajectory), infer about the actual context (likeliness of events, filtering out false detections) and control the position refinement process based on the recognized environment.

Unfortunately, deterioration in DR trajectory quality can be expected during DREAR's usage, due to the cumulative nature of both the heading and stride length estimation errors. Due to the erroneous heading information a wrong reference point (e.g. an escalator/stair, called Topological Anchor Points - TAP) could be selected from a neighbouring set of TAPs with similar characteristics by the localization process (TAP association problem), or due to trajectory dilation/shrinkage a wrong turn could be proposed for the user (location refinement problem). Therefore, we enhanced DREAR to cope with heading errors and their effect on TAP's selection outcome. We propose a trajectory reconstruction algorithm, which extends the context interpreter and improves activity filtering with TAP selection and trajectory matching. We present a detailed analysis of the designed algorithm, considering different indoor scenarios from a complex subway system. As we will show the proposed algorithm presents several positive effects related to the quality of the constructed trajectory, and the proper selection of the specific indoor reference points (TAPs).

Besides the relief from infrastructure dependence the proposed solution will be especially helpful to build end-to-end navigation for multi-modal transportation services [5]. This way the new trend to provide public transit information services for subway systems (e.g. Embark) can be extended with navigation and real-time route guidance solutions.

The paper is structured as follows: the related work is presented in the next Section, followed by the main Section presenting the proposed solution. The performance evaluation of the solution is presented in Section 4. Finally, the paper is summarized in Section 5.

2 Related Work

Unsupervised localization [3] [6] [7] [8] [9] is a recent trend to create and maintain indoor localization infrastructures with minimal assumptions or maintenance costs. To reduce calibration efforts additional mechanisms are proposed for the unsupervised collection and maintenance of the wireless fingerprint space or to reduce the assumptions in modelling. Both Zee [6] and LiFS [7] leverages human mobility to build user paths and to fuse DR results with WiFi fingerprints. While in Zee the DR is matched against the map of the building by using a particle filtering algorithm, in LiFS multidimensional scaling is applied to obtain a higher dimension fingerprint space representation. The main problem with these methods is in their mapping and path reconstruction processes, since a longer trail has to be traversed to get enough information for the backward belief propagation algorithm (Zee), or only corridor and room recognition is currently possible (LiFS). This leads to their limited applicability for complex indoor spaces (e.g. subway systems) or to the requirement to have uniquely structured scenarios (e.g. corridors), because shorter or ambiguously shaped trails cannot be unequivocally recognized. UnLoc [3] introduces an unsupervised indoor localization scheme where virtual landmarks are created using the existing WiFi infrastructure. The smartphones are used to collect environment signatures (e.g. produced by an elevator) and associate these with a locally overheard WiFi AP. Later the DR algorithm can be corrected if the mobile senses the respective landmark. While this method can incorporate different type of landmarks the collection and clustering of these must be confined to a small geographical region. Therefore, all the landmarks are associated with WiFi areas, which lead to an infrastructure lock-in problem, just like in the case of the previous solutions.

FootPath [10] presents an infrastructure-free solution where only the accelerometer and compass are used to provide turn-by-turn instructions. The authors propose matching techniques to compare and project the detected steps onto the expected route. Unfortunately, the sequence alignment algorithm can work only for long and specially shaped corridors. Obviously, the drawback of FootPath is similar to Zee's: if the planned path and the traversed scenario is not specific enough the user's path cannot be reconstructed properly. A different solution is proposed in FTrack [11], where an infrastructure-free floor localization system is presented. The accelerometer data is used to detect walking, climbing stairs and taking the elevator. Then by capturing user encounters (using Bluetooth) and analysing user trails FTrack finds the mapping from the traveling time or the step counts. While FTrack achieves high accuracy in a huge, 10-floor office building, its methods are inappropriate to be directly applied in the context of subway systems, where the topology is much more complex and where the exact paths must be also reconstructed with a finer granularity. In [12] the authors propose a solution which uses the mobile's cameras to recognize store logos for providing indoor localization. While this solution provides an infrastructure-free localization it cannot be used in subway systems where there are no such logos or the texts (commercials) are changing frequently.

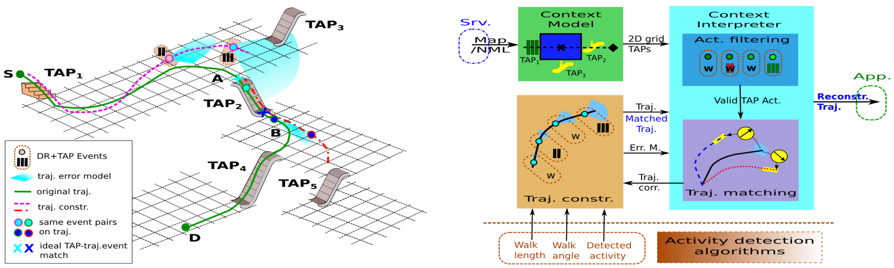
Body-attached IMU-based DR uses heuristic drift elimination methods [13] [14] to minimize the drifting errors. These solutions assume that the majority of buildings have dominant directions defined by the orientation of their corridors; consequently a person walks along straight-line paths parallel to these. However, if no dominant directions are available (larger premises, halls) or a crowded subway station has to be crossed (sidestep people in your way) similar methods cannot be applied confidently. The foot-mounted approach of these solutions is also not applicable for our case.

3 TREKIE: Trajectory REconstruction Based on Knowledge Inferred from Environment

3.1 Problem Statement

Our goal is to provide a truly infrastructure-free indoor localization solution precise enough to navigate pedestrians in complex scenarios, such as subway systems, using only conventional smartphones. Unfortunately, deterioration in DR trajectory quality can be expected due to the cumulative nature of both the heading and stride length estimation errors. Since in the targeted scenarios usually there is no localization infrastructure present (e.g., RF, visual or acoustic beacons) applicable for position refinements, we extended the sensing capabilities of smartphones to detect and recognize human motion patterns specific of traversing particular structures of the indoor places. These structures (e.g. stairs, elevators, escalators) characteristic of building interiors are used to provide the required reference points (called *Topological Anchor Points -TAPs*) for localization error cancellation. While previous efforts [4] managed to provide an infrastructure-free localization solution through the introduction of context-related filtering, there are still issues to be solved, in order to provide a fully functional ubiquitous localization service.

Considering, for example a subway system (Figure 1(a)), along its route the user have to pass through several TAPs (2 and 4) and make proper decisions, turns in intersections (Zone A, B) to reach its destination (D). However, due to the accumulated DR trajectory errors (bluish cones on figure), by the time the user reaches the respective decision points wrong navigation instructions could be provided for him/her. In the constructed trajectories (dashed lines on figure) one of these errors can be attributed to the difficulties to provide proper heading error compensations, since currently there are no steady solutions, especially for such extreme indoor scenarios, with unusual topologies and frequent magnetic perturbations. Due to the erroneous heading information a wrong association could happen (TAP 3 in Zone A), respectively due to trajectory dilation/shrinkage a wrong turn could be made (late turn in Zone B).



(a) Floor plan of a subway system

(b) The architecture of the context-aware localization middleware

Fig. 1. Problem definition and building blocks for context-aware trajectory reconstruction

Even in cases when the users selects the proper escalator (and consequently the respective TAP events are detected) there will be no enough information for the localization middleware to decide about the outcome of the user's TAP selection (TAP 2 or 3), because the heading information of the DR is unreliable. In such cases a wrong turn could be interpreted as a correct route selection, and vice versa, following the planned path and choosing the proper TAP could be considered as taking the wrong direction.

Therefore, we shall employ a mechanism which helps the context interpreter to make correct associations (the factual TAP selection) in case of multiple TAPs, to confirm internally the outcome of the user's decision, and to act accordingly with localization error cancellation, respectively route guidance/re-planning. The trajectory dilation/shrinkage, attributed to stride length estimation errors for normal walking and walking on stairs/escalators, can be also compensated if the TAP event detections are properly fitted to the respective TAPs (by reducing the gap between the ideal and constructed trajectories' TAP events).

3.2 Concept of Trajectory Reconstruction Based on Context-Aware Middleware

The Context-Aware Localization Middleware. To enable the detection of TAPs DREAR [4] introduces feature detection algorithms used for human motion recognition, giving the ability to derive information for the error cancellation process. The activity recognition tasks were interpreted as a classification problem, aiming to recognize unique motion patterns in real-time with a fuzzy associative classifier. The activity and special TAP recognition (e.g. escalator) algorithms have been implemented as a separate layer in the middleware (see Figure 1(b)), producing the respective class labels and the DR information.

The location refinement is controlled by the detected TAP events provided by the activity detection algorithms. Unfortunately, due to the variance in user movement patterns or phone carrying habits the activity detections can lead to wrong predictions (on Figure 1(a) the false TAP events on traj. const.), providing false results (e.g. stair down instead of walking event). These false events will affect the proper refinement of the user's position. Therefore, a filtering mechanism was employed to estimate and remove unlikely or false events, to properly control the localization process. Since there can be long stairs and escalators it is also important to precisely detect their start/end to provide accurate zone traversal detections.

Algorithm 1. Trajectory Reconstruction Algorithm

```
Input: User position, heading (U_{xy,h}), Buffered trajectory (Traj), Error Model of Traj
       (Err.M_{dist,head}), Valid TAP Activity (TAP_{act}), Set of neighbouring TAPs
       (TAP_{1:n})
Output: Reconstucted trajectory Reconstr.Traj
foreach TAP_i ∈ TAP_{1:n} do
    if (distance(U_{xy}, TAP_i) <= Err.M_{dist}) && (heading(TAP_i) ⊆ Err.M_{head}) then
        Create temporary Traj_i from Traj ;
        Calculate Traj.corr = heading(TAP_i) − U_h;
        Calculate heading_diff = Traj_i[U_h[last]] − Traj_i[U_h[first]] ;
        foreach U_{xy,h} ∈ Traj_i do
            Calculate Matched.Traj_i = Correction(Traj.corr, heading_diff) applied on
            Traj_i[U_{xy,h}]
        Estimate reconstruction quality of Matched.Traj_i {
          Select TAP_{act}[1:k] from Matched.Traj_i[TAP[last : last − k]];
          Apply TAP Association(TAP Model(TAP_i), TAP_{act}[1 : k]);
          Calculate Recon.Quality(Matched.Traj_i) from TAP Association;
        }
        Rank Recon.Quality_{1:i};
Reconstr.Traj = Traj_{Recon.Quality_1};
Apply Viterbi Decoding on Reconstr.Traj;
```

The role of the context-aware localization middleware is to provide the required functionality by making assumptions about the user's actual situation, namely to acquire, match and understand the perceived sensory information relative to the current belief about the ambient environment. Then the localization refinement can be influenced based on the recognized context and proper actions can be triggered during the navigation.

The context interpreter of DREAR is used to fuse and process the different levels of information, in order to understand the active context in its entirety. Its core is formed from a Hidden Markov Model (HMM), which is built, managed by the Context Model, and it will directly influence the constructed trajectory, respectively the evolution and control of the DR error. Viterbi decoding is effectuated on the model to find the maximum likely sequence of traversed zones which led to the detected TAP, if there is any TAP event.

Extending the Middleware Architecture. In order to provide solution for the problems mentioned in Section 3.1 we enhanced DREAR to cope with heading errors and their effect on TAP's selection outcome. One solution would be to increase the number of hidden states of the planar 2D grid zones and TAP models to represent the transition probabilities related to the evolution of the trajectory's heading. However, this decision would introduce a substantial increase in the system state dimensions, hindering the middleware's deployment on mobiles. The advantage of such a detailed representation is also questionable due to the aforementioned difficulties in heading error compensations, which possibly would result in infrequent heading corrections.

Therefore, we decided to employ a separation of concerns in the Context Interpreter (Figure 1(b)), treating separately the tasks related to recognition and understanding the detected TAP events considering the actual context. The first stage is provided by the *Activity Filtering* process, with the goal of finding and affirming valid TAP events from a noisy stream of detected activities and a distorted trajectory. When a valid TAP activity is detected the *Trajectory Matching* process can be triggered, where the reconstruction of the real trajectory can be effectuated, based on the characteristics of the recognized TAP and the knowledge derived from the actual context.

From the neighbouring TAPs' information delivered by the Context Model the TAP orientations can be used to calculate the hypothetical trajectory headings, which would occur if the respective, "virtual" TAP, was crossed by the user. Then a correction factor is applied on the heading and position elements of the original, buffered trajectory for each of these hypothetical TAP orientations. The matching process will search for an optimally fitted event sequence (the best match) between the detected TAP events along the traversed path and the structure of the neighbouring, physical TAPs. Its outcome is the TAP selection with the best match (the associated TAP), for which the trajectory reconstruction provides the most viable explanation considering the actual context.

Thus, in the trajectory reconstruction process we combine past and present sensory and context related information. The process is triggered by the occurrence of certain events, such as a confirmed TAP detection or acknowledged heading information. These will provide additional knowledge regarding the evolution of the trajectory by that time the respective event occurred, this information can be exploited to provide insights about the errors acting on sensor measurements and DR calculations (the trajectory).

Obviously, besides the utilization of these "virtual" TAP headings, the concept of trajectory reconstruction can also cope with the occasionally acknowledged heading information, coming from a fingerprint database or an orientation filtering algorithm.

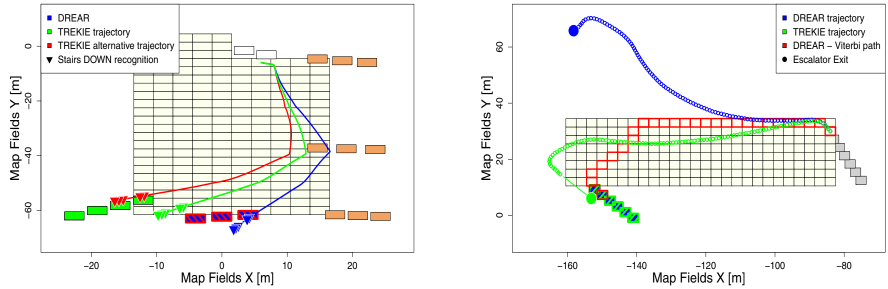
3.3 The Trajectory Reconstruction Algorithm

As we presented above the algorithm starts when a valid TAP activity is detected by the Activity Filtering process, which is the positive outcome of the Viterbi decoding algorithm (a certain number of TAP events - parameter k - are detected and the model's state converges to a TAP zone). Then from the set of neighbouring TAPs (provided by the Context Model) the trajectory reconstruction process is triggered for those, where the distance between the user's position and the inspected TAP is inside, respectively the TAP orientation is within the bounds of the error model ($Err.M$). For such TAPs we calculate the correction factor ($Traj.corr$), the difference between the "virtual" TAP and user's heading, and the heading difference (*heading_diff*) accumulated along the user's path (trajectory construction phase). Using these values a *Correction* algorithm can be applied on a copy of the original trajectory, by using different heading/position alteration strategies on atomic/aggregated trajectory sections. Currently we use a linear and atomic correction algorithm. After this, for each trajectory reconstruction's outcome we have to estimate its quality. This will be effectuated by applying the *TAP Model* of the respective "virtual" TAP on the TAP events (1:k) which triggered the reconstruction process. A "virtual" TAP is more preferable than the others if its position, orientation and other physical characteristics (e.g. length, shape) provide better matching, a more viable explanation (the *TAP Association*), for the detected TAP events. Finally, based on the quality ranking (*Recon.Quality*) the best match will be selected (the reconstructed trajectory: *Reconstr.Traj*), which will be used in the Context Interpreter for Viterbi decoding, together with the newly arriving DR and TAP events; thus, continuing the localization and position refinement process.

4 Evaluation

We evaluated the performance of TREKIE focusing on certain aspects of the localization middleware by designating different walking paths (trails) in the most complex subway station of Budapest. To validate our solution four users performed several measurements for each of these trails, by using the escalators in different ways (e.g. walking and/or standing on it). The localization middleware was implemented on the Android mobile OS platform; the measurements were made with Samsung Galaxy S3 and S4 phones, mounted on belt. Two DR algorithms were implemented, based on step counting and different, personalized step lengths calculations [15], both tested with and without heading error compensations [16]. Due to space limits we present the evaluation only for a couple of trails; however, we should mention that for over 80% of the measurements TREKIE provided proper localization performance (enough precision to find the proper upcoming turning points). The scenarios are presented through a graphic representation of the world model using R, specifically designed for analysis and visualization purposes. In this model the planar 2D grid zones are constructed from grid maps, while the TAPs are represented by a series of consecutive zone elements, with length and direction conform to their actual dimensions.

Figure 2(a) presents the evaluation of a single scenario, where the user walked from the upper stairs (middle white TAP), crossed through the hall (2D grid) and reached the first escalator (TAP at bottom left). We can observe that when the TAP events are recognized (stairs down activities), due to the heading drift of the trajectory construction, the plain DREAR algorithm (blue line) selects the wrong TAP (blue striped stairs at the bottom centre). This would lead to an erroneous position cancellation and a confused navigation process, which in this case would be the conclusion that the user missed the correct TAP; thus, a replanning process is triggered. Such TAP association problems can happen if the neighbouring TAPs have similar types, or the activity detection algorithms



(a) Trajectory reconstruction in case of multiple TAPs (green: correct TAP, red/blue: wrong TAP)

(b) The effect of trajectory reconstruction on path and TAP event alignments (escalator exit detection)

Fig. 2. The effect of trajectory reconstruction considering different TAP scenarios

cannot distinguish TAP directions (e.g. up or down stair/escalator was taken). Fortunately, by applying TREKIE two derived, hypothetical trajectories (red and green lines) are built for these neighbouring TAPs, from which the best fit (green line) will help to select the TAP used in reality. While on the figure the other reconstruction (red line) could seem as a better fit, we must mention that its reconstruction quality will be much worse, due to its distance to the corresponding (striped, bottom centre) TAP.

Other aspects of the trajectory reconstruction algorithm are presented on Figure 2(b), where a plain DREAR trajectory (blue line) and its reconstructed TREKIE trajectory (green line) are represented. Here the user traversed a long corridor, which connects two escalators. The outcome of the Viterbi decoding (Viterbi path) is also represented on the figure. Since TREKIE approximates better the real path we will get an optimally fitted event sequence between the detected TAP events (escalator exit event) and the structure of the physical TAP. This helps to artificially effectuate a dilation on the final, reconstructed trajectory (green line between trajectory and TAP event), helping to cope with the different stride length estimations, necessary to properly consider walking on a longer escalator. Therefore, TREKIE's error cancellation is more efficient than DREAR's.

The effects of the TAP association on the final trajectory selection can be observed on Figure 3, where trajectory reconstruction in effectuated for three neighbouring TAPs (red dashed TAPs), resulting in wrong path reconstruction qualities (red dashed lines), while the original trajectory represents a better result (blue line). Fortunately, due to the TAP association step TREKIE selects the original trajectory instead of the other options, which besides of a more accurate alignment (TAP events get closer to the actual TAP) it will also provide a more concise, smaller DR error.

Figure 4 presents a scenario where the walking path starts outdoors and crosses through two escalators to reach the destination metro platform. There are two decision points along this path, one at the end of the first hall

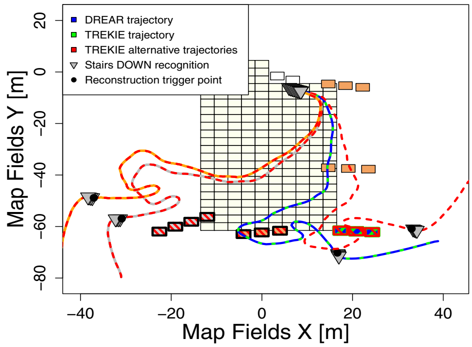


Fig. 3. The effect of TAP association on trajectory selection (blue: original, green: selected, red: alternatives)

(to find the first escalator), and one after the first escalator, to make a proper turn, in an intersection, to reach the second escalator, which leads to the destination platform (left panel on Figure 4). Without position refinements the DR error would grow unbounded, leading to inaccurate localization; thus, wrong navigation instructions would be provided. The question is if the localization is accurate enough to guide the user to the respective platform and how the different algorithms are performing in this context. During the measurements we recorded at specific points a predestined set of positions as ground truth; all positions on the graphs are related to these points. Thus, we are able to present the effect of TAPs on the evolution of the localization error. From these measurements we selected three different cases: the users stand still on the first, and acted differently (walking, standing, mixed movement patterns) on the second escalator.

On the left figure it can be observed how the trajectory reconstructions of TREKIE (green lines) and the plain DREAR trajectories (red lines) are related to each other. In this case the DREAR algorithm also finds the destination platform, since there is no possibility of TAP confusions, thanks to their type differences (escalators: grey, stairs: orange). The effect of trajectory reconstructions on localization error's evolution can be observer on the right figure. As we can see, the localization error for the reconstructed trajectories (red lines) is far below the ones presented by the DREAR algorithm. While at the end of the first escalator the localization error is reduced consistently to 2 metres, in case of the second escalator the error cancellation will vary between 1 and 6 metres.

Despite the eventuality to produce huge errors in position refinements (up to 40 metres for the second escalator) all the cases ended with a controlled and suppressed positioning error. Without this the raw DR (without DREAR+TR) would mess up the trajectory, leading to wrong navigation instructions. While in this case we did not profit directly from the TAP association and event alignment

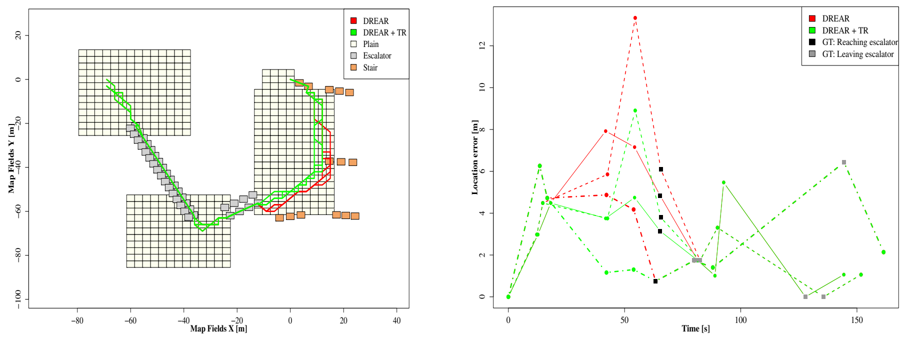


Fig. 4. The effect of trajectory reconstruction on trajectory matching

characteristics of TREKIE, since the first escalator was precisely recognized, and the distance to the second escalator was too short to produce significant DR errors, it became obvious that trajectory reconstruction can be also helpful for cases when tracking or map building functionalities must be provided. For such applications we can guarantee a more precise and realistic user path construction with less error than in case of a simple DR or even the DREAR algorithm.

5 Conclusions

In this paper we enhanced the approach of DREAR, a mobile-based localization middleware, which allows to exclude the infrastructural reference points (RF or visual), hence to provide a completely infrastructure-free localization system. Our goal was to further improve the context interpretation process, the detection of user movement activities (e.g. walking, taking stairs, using escalators) and the selection, association of position refinement points (TAPs) along the users' trajectories. The proposed solution was thoroughly evaluated in a complex subway system, focusing on different aspects of the trajectory reconstruction and localization process. TREKIE presents clear improvements related to the performance of DREAR, with promising results and a great potential for real-world scenarios (e.g. subway stations). As future work we want to consider also the inclusion of acknowledged heading information in the trajectory reconstruction process.

References

1. Jin, Y., Motani, M., Soh, W.-S., Zhang, J.: Sparsetrack: enhancing indoor pedestrian tracking with sparse infrastructure support. In: IEEE Infocom 2010, San Diego, California, USA (2010)
2. Youssef, M., Agrawala, A.: The horus wlan location determination system. In: ACM MobiSys 2005, Seattle, Washington (2005). doi:10.1145/1067170.1067193

3. Wang, H., Sen, S., Elgohary, A., Farid, M., Youssef, M., Choudhury, R.R.: No need to war-drive: unsupervised indoor localization. In: ACM MobiSys 2012, Low Wood Bay, Lake District, UK (2012)
4. Török, A., Nagy, A., Kováts, L., Pach, P.: Drear - towards infrastructure-free indoor localization via dead-reckoning enhanced with activity recognition. In: Conference on Next Generation Mobile Apps, Services and Technologies, NGMAST 2014, Oxford, UK (2014)
5. Rehrl, K., Bruntsch, S., Mentz, H.-J.: Assisting multimodal travelers: Design and prototypical implementation of a personal travel companion. Trans. Intell. Transport. Sys. **8**(1), 31–42 (2007). doi:10.1109/TITS.2006.890077. http://dx.doi.org/10.1109/TITS.2006.890077
6. Rai, A., Chintalapudi, K.K., Padmanabhan, V.N., Sen, R.: Zee: zero-effort crowdsourcing for indoor localization. In: ACM Mobicom 2012, Istanbul, Turkey, (2012). doi:10.1145/2348543.2348580
7. Yang, Z., Wu, C., Liu, Y.: Locating in fingerprint space: wireless indoor localization with little human intervention. In: ACM Mobicom 2012, Istanbul, Turkey, (2012). doi10.1145/2348543.2348578
8. Liu, J., Chen, R., Pei, L., Guinness, R., Kuusniemi, H.: A Hybrid Smartphone Indoor Positioning Solution for Mobile LBS. Sensors **2012**(12), 17208–17233 (2012). doi:10.3390/s121217208
9. Liu, K., Liu, X., Xiaolin, L.: Guoguo: Enabling Fine-grained Indoor Localization via Smartphone. In: ACM MobiSys 2013, Taipei, Taiwan, June 25–28 (2013)
10. Link, J.A., Smith, P., Wehrle, K.: Footpath: accurate map-based indoor navigation using smartphones. In: Indoor Positioning and Indoor Navigation (IPIN) Conference, Guimaraes, Portugal (2011)
11. Ye, H., Gu, T., Zhu, X., Xu, J., Tao, X., Lu, J., Jin, N.: Ftrack: Infrastructure-free floor localization via mobile phone sensing, pp. 2–10. IEEE PerCom, Lugano, Switzerland (2012)
12. Tian, Y., et al.: Towards ubiquitous indoor localization service leveraging environmental physical features. In: IEEE Infocom 2014
13. Jimenez, A., Seco, F., Zampella, F., Prieto, J., Guevara, J.: Improved heuristic drift elimination (ihde) for pedestrian navigation in complex buildings. In: Indoor Positioning and Indoor Navigation (IPIN) Conference, Guimaraes, Portugal (2011). doi:10.1109/IPIN.2011.6071923
14. Angermann, M., Robertson, P.: Footslam: pedestrian simultaneous localization and mapping without exteroceptive sensors - hitchhiking on human perception and cognition, In: Proceedings of the IEEE, Vol. 100, May 13, 2012
15. Weinberg, H.: Using the adxl202 in pedometer and personal navigation applications. Analog Devices AN-602 application note (2002)
16. Madgwick, S.O.H.: An efficient orientation filter for inertial and inertial/magnetic sensor arrays. Technical report, Univ. of Bristol, UK (2010)