Creating a novel approach for mobile positioning based on CDR data

Salijona Dymishi
Supervising: Amnir Hadachi, PhD
Distributed Systems Seminar, Spring 2020
University of Tartu

Abstract—User geographical positioning is important for many fields that rely on passive data analytics, like targeted marketing, urban and rural transportation planning, public health, etc. A new type of data that is commonly used for passive mobility analysis is the mobile data or the so-called Call Detail Records (CDR). The CDR events are stored by mobile operators for the primary purpose of billing. They are generated every time we use SMS, call, or Internet services. However, CDR data has two major drawbacks: the temporal and spatial uncertainties. Although the first problem is widely covered by trajectory reconstruction techniques, the second problem remains still challenging. Hence, in this project, we propose the usage of a new method based on the Sequential Monte Carlo algorithms called Particle Filtering. The particle filtering application implemented in this project, models the trajectory movement to predict the user’s position in a given area. This method uses CDR data and solely the information related to the area of the coverage from the mobile towers. Our goal is to evaluate if this nonlinear method can outperform the existent linear methods like Switching Kalman Filter. The usability of the method and future work are discussed in this report.

Index Terms—mobile data, particle filtering, location prediction, trajectory prediction

I. INTRODUCTION

Passive mobile positioning gives access to understanding the spatio-temporal behavior of the users. Lately, this type of data is used in a vast pool of fields like mining human mobility patterns, urban planning, tourism estimation, vaccination planning, public transport rescheduling, etc. The reason for the popularity in so many fields is mainly because passive mobile positioning serves as a great replacement for traditional methods that build models upon samples rather than data. The common passive mobile positioning data are the logs from Mobile Network Operators (MNO). Every time we use the mobile phone to make a phone call, send an SMS, or use 4G/5G internet packages they store what is called Call Detail Records (CDR) for the primary purpose of billing. The term CDR might be misleading because it dates from the time the only service you could use were mobile calls but nowadays they are triggered on the event of calls, sending/receiving SMS, and using Internet services. In particular cases, the configuration allows the generation of mobile logs additionally every time the user switches cells or LAC.

Typically, a CDR record has the following fields [1]:

- **Unique sequence number**: The Sequence number identifying the record
- **Calling party**: The phone number of the caller
- **Called party**: The phone number of the receiver
- **Billed number**: The billing phone number that is charged for the call
- **Call duration**: The duration of the call in minutes
- **Stat time**: The starting time of the call (date and time)
- **Call type**: The type of call that was made (VoIP (Voice over Internet Protocol), voice, or raw data)
- **Cell global identity (CGI)**: Unique identifier for each cell

The minimum required information for geo-spatial analysis are the timestamp and the CGI paired with the cell coverage area. For mobile positioning, the geographical shape of each cell plays an important role. The shape depends on factors such as antenna radiation pattern and height, network load, signal attenuation on the landscape and indoors, signal reflections, radio interference and noise, network configuration parameters etc [2].

Nowadays, there is no doubt that the mobile penetration rate is increasing even in the most rural/unreachable places. Additionally, the cost reduction of telecom services has contributed to the increase in the spatial area coverage of the CDR data, as well as an increase in its frequency. This situation is promising for an increase in importance for CDR data and probably an expansion to other fields. CDR data is easily available considering the fact that the Mobile Network Operators (MNO) always store them and can be retrieved for all users of a network. Fiadino et al. [3] have taken under study CDR data-sets describing the activity of customers from all mobile operators in Spain, from 2014 and 2016. Their comparison of the data quality showed that the number of daily data connection has increased in 2016 from 10.9 to 50.1 and overall the daily action count grew by 4 times. The data connection customer share had the largest growth from 50% to 85%. Several factors were considered like Days of Visibility, Hourly Action Rate, Average Lag Time, Total Inactive Time, and Entropy on this study. The authors revealed that all of them had marked improvements from 2014 to 2016. This study is a good indicator that the user patterns have changed and the CDR data is becoming more fine-grained for mobility studies. This feature makes CDR highly desirable for passive spatio-
temporal analysis, however, they have some disadvantages.

The first disadvantage comes from the fact that the radius of the tower coverage introduces difficulties for the exact user positioning. Every CDR record is coupled with the coverage area of the tower where the user is connected. These areas called cells, do not have unique shapes. They vary from the height of the tower location, the urban environment around the tower, population density, the obstacles etc. The radius of the cell ranges from several meters in urban areas to tens of kilometers in rural areas. Within this radius, the location of the mobile phone user is actually unknown. The second disadvantage is related to temporal uncertainties. Being that CDR data is not collected in constant predefined intervals, the gaps between two consecutive events can be really considerable. They might start from one minute to several hours. Unfortunately, the last one is a common cause. The high level of spatial and temporal uncertainties can have a negative effect on the reliability of the studies related to human mobility patterns.

In order to reduce the uncertainties, it is necessary to apply models that will firstly preprocess CDR data like eliminating the ping pong and other undesirable effects related to network functionalities. Secondly, it is necessary to reduce temporal uncertainties by building models which will fill in the gaps between consecutive events. Thirdly, estimating the user location within the cell in order to reduce spatial uncertainties. From the study of related literature, we have noticed that the majority of studies apply the first technique before performing aggregated analysis in fields like tourism, healthcare, transport planning, etc. Fewer studies deal with trajectory reconstruction for the gaps. Lastly, only a couple of studies tries to localize the users within the coverage areas of the towers.

II. BACKGROUND WORK

CDR data does not seem to be an unknown source for researchers. With the increase of its availability, the CDR is becoming complement data for many passive mobility analyses, especially in humanities studies. And lately, the application fields are expanding. However, we have found really few studies which focus on actual user positioning and path reconstruction.

We can separate the research focusing on CDR in two major groups:
- Context aware analysis
- Trajectory reconstruction and localization

The largest volume of studies uses methods from the first group. However, this is not the area we are interested. Therefore, we are going to present the second group work. In the second group, there is research done to deal with the uncertainty of mobile phone data. Firstly, the researchers have tried to deal with sparsity. The most common technique to do so is to fill the gaps with new cell IDs based on trajectory reconstruction. Trajectory reconstruction aims to infer information from aggregated user movements and then fill the missing values for unknown individual user position. [4] Hoteit et al. categorized mobile phone users in 4 categories: sedentary people, urban mobile people, peri-urban mobile people, and commuters by analyzing the cumulative distribution of the radius of gyration. They select users with more than 1000 data points during a given day and sub-sample from those trajectories to produce normal user behaviors. On the sub-sampled data, they use three classical interpolation techniques to do trajectory reconstruction. Finally, they evaluate their methods by comparing them with data before sub-sampling. In their study, the authors are not considering the cell surface but rather the BTS longitude and latitude. Another tool to complete individual CDR-based trajectories is proposed by Chen et al. in [5] based on tensor factorization. On a first iteration, the authors consider the users whose home locations are known with 80% confidence. For these users, they fill in the night gaps depending on time granularity. In a second step, the data is organized in 3D tensors which represent the daily, weekly, and instantaneous mobility of the user. They use the tensor factorization method, which has been useful in other contexts, to fill in the tensors which on average were only 0.63% complete. Tensor factorization decomposes the tensor in hyperparameters and then uses them to generate new data to fill the voids. Lastly, the positions generated are mapped back to the coordinates of the closest BTS. Their results show that they can locate users with a median displacement between 1 and 2 network cells, meaning the user location is predicted on the exact cell or in the cell next to it. In another study [6], the authors worked on trajectory reconstruction based on the path edge level. They select for every two consecutive events in the CDR trajectory a random point from the uniform distribution and map it to the closest road segment. The fastest path between two points is considered as a possible trajectory. We can consider the random points as a location estimation within the cell.

The last domain of the second group, the least explored one, is the reduction of spatial uncertainties by estimating the position of the user within the cell-plan. The only study we have found in this area is from Hadachi et Lind [7] where the authors use a version of Kalman Filter called the Switching Kalman filter with smoothing. The model tries to extract the movement patterns from the data-driven exploration phase and label each record as a Stay or Jump position. Simultaneously the authors estimated the user position within the cell and mapped it to the closest road segment or building. The results were quite promising especially for the stay model with an RMSE of 2106.8 meters. However, the Switching Kalman Filter falls into the category of linear models.

Regarding the usage of non-linear Monte Carlo Sequential models like particle filtering for positioning, the research dates as early as the 2000s but applied to other domains. For example in [8] the authors proposed to use particle filtering for car positioning. The measurement is taken from wheel speed sensors in the car and the velocity vector is considered as the input signal. The authors argued that it is possible to use mobile phone measurement data as an input vector. In another study by Hu et al. [9], the authors are using a modified version of particle filtering to localize moving nodes in a range-free wireless network. Their search space is divided into equal
rectangles. As part of their work, they analyze the effects of sample size and motion control into accuracy, but they do not compare different sizes of the coverage. The technique was tested on a simulated environment and resulted in improved accuracy of localization compared to the case where the mobility information is not taken into consideration. Later Dil et al. [10] used the same model, as the previous authors, for range-based wireless networks. Similarly, they used fixed-size areas and evaluated the effects of the sample size and speed. The reported localization improvement ranged from 12% to 49%.

III. PROPOSED METHOD: PARTICLE FILTERING FOR USER POSITIONING

The problem of the user localization has been solved until now using linear Bayesian methods like Switching Kalman Filter. Considering that the users do not move in a linear manner we adapted a nonlinear method like particle filtering to solve the same problem. This section will give an overview of our adaptation of the particle filter. The main steps of the workflow are represented on the flowchart in the Figure 1.

Preparing the data

The algorithm starts by receiving the mobile data trace as input. Every record in the trace should have at least the timestamp and the Cell ID. The Cell ID is coupled with the information about the coverage area of the specific cell. This information is expressed as a list of coordinates (latitude and longitude) that represent the corner points of the cell polygon. Any desired preprocessing on the trajectory should happen at this stage. For example it is beneficial to detect ping-pong effects, to detect long gaps between consecutive events etc. We will speak on more details about this step in Chapter 4.

As the second parameter for the input serves the geographic area where the CDR records are spread. We are using the OpenStreetMap (OSM) API to download the data in XML format. On a second step the data is read from the file and converted to a graph structure where each building is represented as a node. If two nodes are connected to each other in any way, then an edge \( e(u, v) \) is added to the graph where \( u \) marks the starting point expressed in geo-coordinates and \( v \) the end point. OpenStreetMap is an open-source project and relies on the contribution of its members for the completeness of its data. The edge element in OSM output has a tag for maximum speed allowed but that is not the case for every edge. When this tag does exist, we add the maximum speed as a property to the edges of our newly created graph. The rest of the time, when the maximum speed limit is missing we add a constant value. This value is calculated based on the average traffic characteristics of the geographical area where the movements of our users are spread.

The third parameter that the algorithm receives is the necessary number of particles \( S \), to model the distribution of the user location. After we have specified all these three inputs we are ready to start with the next phase which is initialization.

Initialization

At the initialization stage the algorithm randomly selects \( S \) edges in the complete graph from uniform distribution. Each edge endpoint is assigned a weight of \( \frac{1}{S} \) as a possible location of the user. At this step we do not have any prior knowledge on the user location therefore there are no restrictions on the graph. The user is equally probable to have started his journey at any point. The time parameter at the initialization phase is set as well to 0. In Figure 2 we are going to visually demonstrate the stages of the algorithm with an example which uses three particles. As discussed previously the hexagons represent a cell and the grid with squares represents the road network. Until now, we have presented only Step 1: Initialization. The red segments show the randomly selected edges. Notice that the line width is proportional to the probability of every edge, \( \frac{1}{S} \).

Preparing the data

The algorithm starts by receiving the mobile data trace as input. Every record in the trace should have at least the timestamp and the Cell ID. The Cell ID is coupled with the information about the coverage area of the specific cell. This information is expressed as a list of coordinates (latitude and longitude) that represent the corner points of the cell polygon. Any desired preprocessing on the trajectory should happen at this stage. For example it is beneficial to detect ping-pong effects, to detect long gaps between consecutive events etc. We will speak on more details about this step in Chapter 4.

As the second parameter for the input serves the geographic area where the CDR records are spread. We are using the OpenStreetMap (OSM) API to download the data in XML format. On a second step the data is read from the file and converted to a graph structure where each building is represented as a node. If two nodes are connected to each other in any way, then an edge \( e(u, v) \) is added to the graph where \( u \) marks the starting point expressed in geo-coordinates and \( v \) the end point. OpenStreetMap is an open-source project and relies on the contribution of its members for the completeness of its data. The edge element in OSM output has a tag for maximum speed allowed but that is not the case for every edge. When this tag does exist, we add the maximum speed as a property to the edges of our newly created graph. The rest of the time, when the maximum speed limit is missing we add a constant value. This value is calculated based on the average traffic characteristics of the geographical area where the movements of our users are spread.

The third parameter that the algorithm receives is the necessary number of particles \( S \), to model the distribution of the user location. After we have specified all these three inputs we are ready to start with the next phase which is initialization.

Initialization

At the initialization stage the algorithm randomly selects \( S \) edges in the complete graph from uniform distribution. Each edge endpoint is assigned a weight of \( \frac{1}{S} \) as a possible location of the user. At this step we do not have any prior knowledge on the user location therefore there are no restrictions on the graph. The user is equally probable to have started his journey at any point. The time parameter at the initialization phase is set as well to 0. In Figure 2 we are going to visually demonstrate the stages of the algorithm with an example which uses three particles. As discussed previously the hexagons represent a cell and the grid with squares represents the road network. Until now, we have presented only Step 1: Initialization. The red segments show the randomly selected edges. Notice that the line width is proportional to the probability of every edge, \( \frac{1}{S} \).
will present below, from update to resampling, are iterative and happen for every record in the CDR trajectory.

IV. EXPERIMENTS

When it comes to similar applications we already mentioned that there is not enough related work. However, the authors in [7], have taken under the consideration the same problem and used the Switching Kalman Filter approach to solve it. We have received access to the dataset they used in their study and the results acquired. Here we keep constant the number of particles set to 50, meanwhile the cell surface and time granularity are determined by the dataset itself. In this dataset we evaluate the accuracy of the methods by calculating the haversine distance between the predicted user position and the actual GPS point where he was located. Being that for this dataset we have the full GPS trace we add one extra step on the evaluation, path accuracy. Given that we are provided with a full GPS trajectory in the time-span between the capture of two CDR records, we can evaluate the accuracy of the proposed path from our algorithm for the user movement. Every GPS point in the trajectory is mapped to the closest edge on the graph. The result is a list of edges that serve as the ground truth. We find the intersection between this list and the list proposed by the particle filtering algorithm. Afterwards, the accuracy is calculated as a ratio between the common elements of the list and total elements in the GPS trajectory, expressed as a percentage.

For the real data, it is necessary that the trajectories are checked for any inconsistency that would affect the particle filter. We had such a case with long gaps between two records. The particle filtering bases its probability calculation in the traversal time TT and estimated time TD. In cases when the TT is considerably large compared to all TD-s, the algorithm will always define as the most probable the longest path. In order to avoid this bias, we take two measures. Firstly, if the time difference between two consecutive records is more than three hours we consider the trip to be finished. At this moment we interrupt the particle filtering algorithm and restart it from the beginning using the new trajectory that starts from the second record. The second step we perform is calculating the time that is needed to traverse the most distant points between two cells. If this time is lower than the difference between two timestamps it means the user has stayed for a long time in one of the cells without any activity. In this case we calculate TT as the time to traverse the path between two centroids of the cells to avoid the bias towards the longest path.

1) Dataset description: The dataset consists of CDR data of five mobile owners in Estonia, accompanied by the file with their GPS data locations. The period of data extraction is between April and August 2015. This dataset corresponds to a more realistic scenario from the previous dataset, and we can clearly detect that from Figure 3. Most of the data are captured between every 0 and 500 minutes and the users have moved within that period from 0 to 20 km. It is easy to notice here the lack of the information and the challenges with real data we introduced in the beginning of the experiment section. For example, a user who travelled 0 km between two events that have a gap of 2500 minutes can represent two cases. The first one would be that the user has not moved at all from his location i.e. the events are triggered in the evening and later in the morning. In this case the user has been in his home location during the whole time. However, the second case is more complicated. The user might have travelled from the initial point A to a point B and came back to the point A. During this period the two events are generated only when starting from point A and after returning to point A. However, we do not have any trace though for the intermediate event. Another challenging point are the users that travelled around 20 km or more in almost 0 minutes. This happens in the cases where there are problems with CDR data receiver and the timestamp is corrupted. The newly generated CDR record will be assigned to the last seen timestamp. This figure gives a good overview about the nature of real data. Moreover, some outliers were removed and not shown in the figure in order to have a better visualization.

In contrary to our previous example the level of uncertainty related to user positioning in this dataset is significantly increased. In addition, the cell surfaces in real scenarios do not have a perfect hexagonal shape. Their shape resembles more of a distorted polygon or hexagon. From the visualization we could notice a part of them overlap with each other in many cases. In Figure 4 we can see the distribution of the diameter of the cells in this dataset. Notice here, the diameter calculation is an approximation as the cells do not have regular hexagon or circular shape. We have calculated the surface of each cell and from their estimated the diameter of the largest circle we could create within this surface. The majority of the cells have a diameter of less than 1 000 m. This is expected as the majority of the movements of these users happen in the city. The rural cells vary from 5 000 to around 25 000 in diameter which makes the user positioning task really challenging in these cases.

In addition, for two out of five user IDs we had been provided with complete GPS trajectories of user movements. For these users we performed the step of path evaluation. You can see the results of these experiments in the next section.

Update
The algorithm selects all the edges which have both endpoints in the given cell. Assuming a uniform distribution of the user location within the cell, we select randomly $S$ edges from this group. In Figure 2, this process is represented as Step 2: Prediction. The road segments marked with red are the predictions based on the belief and the road segments are the previous state. Until this point the weights are equal. From the selected $S$ edges, the endpoints are taken into consideration. The weighted shortest paths between every edge endpoint from the previous step and the new proposed edges endpoints are calculated. The results have the form of a matrix with a size $S \times S$. The complexity of this step increases the time requirements, therefore it should be computed in parallel. As the weight for the edges, while calculating the shortest path, serves the estimated travel time which is calculated as the ratio between the endpoints distance and the maximum speed limit. For every path, we need to estimate how likely the user has traversed them given the actual traversal time. The actual traversal time, we note it with $TT$, is calculated as the difference between the current CDR record timestamp and the previous record timestamp. The probabilities of traversal for every path are estimated using the formula below:

\[
\text{probs} = \frac{\text{abs}(TT - TD)}{TT}
\]  

(1)

In this formula, $TT$ stands for the actual time difference between the CDR records, and $TD$ stands for the estimated travel time using the shortest path algorithm. The probabilities are then scaled to a range from 0 to 1. Given the $S \times S$ matrix with all the probabilities, we select the highest one. The associated path that produced this probability is assumed to be the trajectory of the user. The selected path with the highest probability is shown in Figure 2 Step 3: Update with yellow color. The line size and the size of the circles which represent the road segment end are proportional to the updated. The endpoint of this path is the estimated location of the user within the cell.

Now we need to update the weights related to the possible user locations. We consider for that purpose the edge endpoints. There are two components that decide the weights. Firstly, we take into account the transition probabilities. For every new proposed vertex, its maximum probability from the matrix of probabilities $S \times S$ is retrieved. As a result, we have a vector with $S$ probabilities, each one assigned to one edge endpoint or vertex. The second factor that defines the weights is what we call the evidence probabilities. In a GSM network, it is more probable that you will trigger an event if you are closer to the cell center due to signal strength. To take into account this effect we calculate the evidence probabilities which are the probabilities of triggering an event and depend on the distance of the proposed vertex towards the cell centroid. The evidence probabilities and the transition probabilities are then multiplied together. The weights from previous steps that are all $\frac{1}{S}$ are multiplied to these probabilities and the result is normalized.

Resampling

A resampling step happens right after where the vertices with the strongest weight are chosen to be duplicated and their weights are assigned to $1/S$. In Figure 2 this step is represented by Step 4: Resample & reweigh.

2) Results: Comparison with Switching Kalman Filter

In Figure 5 we have shown a comparison between the distribution of the results achieved by our particle filter application on user positioning and the results achieved by the proposed Switching Kalman Filter in [7]. The histogram represents bins of size 500 meters. As we can see it seems the Switching Kalman Filter is performing better in the first bin. There are 220 elements grouped there in contrary to 200 for Particle Filter. In the second bin it seems there are more elements from Particle Filter and in the third the situation is the same. In general Kalman has a mean of 988 m and standard deviation of 1631 m. The Particle Filter has a mean of 1571 m and standard deviation of 2586 m. We can see the particle filtering has more outliers with 5 present bins between 10000 m and 17000 m while Kalman has only one bin.

Cell size effect

Another points we are interested in, are the dynamics of our predictions related to the cell size. Are the elements of the first bin in the histogram coming only from the cells with small size? If this is the case it shows our method is somehow biased. Remember that the maximum error we can have is at most equal to the largest diameter of the hexagon. In this case approximately as the estimated circle diameter. Figure 6 represents the relationship between the algorithm’s accuracy and the diameter size. We can see that when the diameter
increases the maximum error increases. For cells with diameter less than 5 000 meters we have a mean error of 508 m and standard deviation of 631 meters. For the cells with diameter between 5 000 and 10 000 m we have a mean error of 2 928 m and std 1 443. For diameters between 10 000 m and 15 000m the mean is equal to 4 434 m and the standard deviation to 2 816 m. For the cells with diameter between 15 000 and 20 000 m we have a mean error of 5 790 m and standard deviation of 3 923 m.

**Path evaluation**

For almost half of the data we were provided with full GPS trajectories. This allowed us to evaluate our predicted paths using the method presented in the section "Path Evaluation". The mapping of GPS location to the edges was being really time-consuming because the GPS data were extracted every 1 second, and we had hundred thousands of edges in our graph. Hence, we decided in cases where there are more than 1000 GPS points in one trajectory we will sample every second record. The loss of information is not a concern as 2 seconds are not enough to traverse one edge. The results after we run the particle filtering algorithm are shown in Figure 7. Most of the accuracies falls within the first bin of 10%. The rest of them seems to be almost uniformly distributed. There are elements even in the last bin of 90 - 100 % accuracy.

**V. DISCUSSIONS**

There was Gustafsson et al. [8] who first said that potentially the mobile data could be used for positioning, in particular with particle filtering. The paper dates from 19 years ago and since then it seems that the researchers have been more interested in using mobile data in an aggregated form as a source for fields like urban planning, transport mode detection and planning, public health, crisis management during earthquakes and COVID-19 scenarios etc. Today, it exists only one study which considers the CDR data for user localization. However instead of particle filtering this study uses the Switching Kalman Filter. This project is the first work that considers the implementation and evaluation of a non-linear Sequential Monte Carlo Method like Particle Filtering for the task of user positioning using only the minimum required information. The main drive behind this work was to understand if it is possible to improve the positioning accuracy using a non-linear method compared to existing linear methods. Any improvements in findings will lead to a better quality data for the study fields that use CDR information for passive human mobility analytics.

One important factor that decides the accuracy of this application is the fact if we can predict correctly the travel times between two nodes. At the present we are calculating this metric using the road segment length and the maximum speed attribute attached to each edge in the OSM extracted data. Nevertheless, to say, a high percentage of these edges are missing this attribute. For these case we have to consider a single default maximum speed that is not differentiated for the city roads or highways. The selection of this default speed limit is arbitrary and tries to take into account the general knowledge about the legislation in the area of interest. This approach introduces a second approximation step in top of the particle filtering which in itself provides approximate solutions. This is an important factor that affects the path accuracy results that we received in Figure 7. It was our expectation that most of the cases the paths would not match the path given by the GPS trace. First due to the approximate nature of Particle Filtering and secondly due to not exact travel time estimations.

From the comparison we did between our method and the non-linear Switching Kalman Filter we noticed although they produce similar results, particle filtering in this data-set was not able to outperform the Switching Kalman Filter. One characteristic that might impact positively the Kalman Filter is the fact it can depict the episode of Movement or Stay. For Particle Filtering in case the user is staying in a position throughout a series of records, it has lower chance to predict the same location. First, because the newly sample particles are selected randomly and the probability that some of them are similar to the last location depends highly on cell sizes. Even if the previous stay location is duplicated in the new set of sampled particles, the probability that is selected again is high when the time gap between two events is small and the probability lowers as the time gap increases. Although we evaluated the results in only 359 data points, it might be necessary to use a larger data-set to compare both methods for the results to be more generalizing.

Particle Filtering in general it is known to be a complex algorithm with high time requirements. This is the main reason why the other methods are preferred in linear models with normal noise, even though it would be totally possible to
use particle filtering as well. In this particular application when we combine particle filtering methods with the shortest path calculations in a large graph with a hundred thousands or millions of edges the time complexity increases. Even after parallel processing performed on the shortest paths calculations, Particle Filtering continues to be slow, especially compared to the SKF.

VI. CONCLUSIONS

This work aimed to evaluate if it is possible to improve the accuracy of the user positioning with mobile CDR data by introducing the adaptation of a non-linear particle filter algorithm to the problem. Our proposed model was evaluated in synthetic and real-case data. Based on the analysis we could conclude our approach was achieving similar results but not better compared to the previous linear techniques like Switching Kalman Filter. The major factor affecting the accuracy of our method is the travel time estimation. In the future it is necessary to sophisticate the travel time estimation approach by modelling spatial and temporal fluctuations in the speed. In addition, it would be in the interest of the research to compare both methods in a larger data sample.

REFERENCES