Correction of Telecom Localization Errors by Context Knowledge

Minmin Zhu[1], Buyang Cao[1(✉)], Mingxuan Yuan[2], and Jia Zeng[2]

[1] School of Software Engineering, Tongji University, Shanghai, China
buyang60@hotmail.com
[2] Huawei Noah's Ark Lab, Shatin, Hong Kong

Abstract. Telecom localization that had aroused widespread attentions of major telecommunication operators has become vital in recent years. However, current available technologies suffer from high localization errors, typically with mean errors more than 100 m. In order to tackle this problem, in this paper we leverage context knowledge to reduce the localization error. To this end, we propose a framework adopting several modified filter methods in terms of context to eliminate localization errors that cannot be easily detected by the existing localization algorithms. We apply the optimized filter methods combining with the context knowledge to verify the effectiveness of our methodologies according to the experiments based on the telecom localization utilizing the GPS-associated MR data in the downtown area of Shanghai, China.

1 Introduction

LBS requires relatively accurate locations of mobile phone users. Nevertheless, the traditional telecom positioning approaches suffer from either low precision (e.g., the range-based methods have the typical mean errors by hundreds of meters) or high costs (the fingerprinting methods have to maintain a fingerprint database). The recent mea- surement report (MR)-based positioning systems instead have many advantages includ- ing availability in most mobile phones and being active whenever users make phone calls and use mobile broadband services. It has been considered as a very useful com- plement to GPS. However, MR positioning systems still cannot achieve high precision.

In the view of the current domestic and international research status, localization schemes on Wireless Sensor Network (WSN) data can be classified as three categories, (1) range-based methods, (2) fingerprinting methods, and (3) model-based methods. The range-based methods use range measurements as physical models [6], which record the TOA (Time of Arrival) and AOA (Angle of Arrival), TDOA (Time Difference of Arrival) and RSSI (Radio Signal Strength Indicator) of transmitting wireless signal by the unknown node hardware receiving from external symbol node, then transform these distance metric values to the distance upon which the related algorithms such as trilateration and triangulation method, maximum likelihood estimation method can be employed.

S. Song et al. (Eds.): APWeb-WAIM 2017 Workshops, LNCS 10612, pp. 107–117, 2017.
https://doi.org/10.1007/978-3-319-69781-9_11

Fingerprint positioning algorithm is a feature matching algorithm, which uses a plurality of signal strength values of wireless routers in positioning environment, and establish the off-line fingerprint database [9] by collecting and training, then to match the real-time collected fingerprints in the positioning process and fingerprint database in order to estimate the best matching positions. The last method is a model-based localization, which uses related machining learning algorithms to learn our excited position estimation, such as Random Forest (RF) algorithm and Artificial Neural Network (ANN) algorithm. This method serves the real GPS position as our training label to build our localization model by training the Measurement Report(MR) provided by mobile phone service providers.

Despite the localization methods we studied had achieved good position estimation results, we had found that lots of localization errors having large deviations from the real values, which could mainly reflect in either the oscillations of position in the same road section or deviating far away from the original real road segments. The direct use of the localization algorithm to calculate the location of a user may produce big error due to the change in user position. Furthermore, the user movement is not very smooth and it might be affecting the real-time positioning system performance and stability seriously. Inevitably, there will be a lot of noise during the process of the signal transmission in addition to the noise produced by the localization algorithm itself, which is a vital reason causes the error of localization.

Furthermore, we found that in addition to the estimated positions deviating seriously from the true positions, there are serious velocity variations in the predicted locations. We believe the speeds between two adjacent positions of a user trajectory do not appear reasonable according our daily life experience. Therefore, solving these localization error problems become very important to improve localization accuracy as the basis for LBS.

In this paper, we introduce several filtering algorithms to remove the abnormal location coordinates created by the positioning algorithm and to further improve the positioning accuracy of the positioning system built on model-based localization.

Our study indicates that it is necessary to propose a practical and effective data postprocessing method to resolve a variety of positioning errors. In our view, every GPS point is not isolated position but contextual related because these GPS points represent real physical locations of user trajectory and normally it could predict the next position according the current or last position, and/or infer the last position according the current position. As we discussed above, the work we attempt to accomplish in this paper is to design a sliding window filter based on map-matching algorithm, which mainly combines with two items of the big rich context knowledge from the entire user trajectory and road network data, in order to eliminate the influence of errors in our localization models.

To summarize, we make the following contributions in this paper:

- Novel application of several filters in trajectory data processing and separated from data preprocessing work

- The first to introduce the algorithm of using context knowledge based on map-matching
- The combination of mathematical filter algorithm model and spatial-temporal data.

The rest of this paper is organized as follows, Sect. 2 describes the problem of localization error to be solved. Section 3 introduces the theory for proposed model and error correction algorithm. Section 4 presents the related experiment results. Finally, Sect. 5 concludes with the efficiency of the proposed method and the possible future work.

2 Background

When mobile users make phone calls using mobile phones or use mobile broadband services, their phones connect to telecom networks, e.g., GSM. The network next generates measurement report (MR) data. The MR data records the received signal strength indicator (RSSI) of nearby base stations to support communication services, etc. On the other hand, the widely-used location based services (LBSs) have accumulated lots of over-the-top (OTT) global positioning system (GPS) data in telco networks. We then use the GPS data as the training labels to learn accurate MR-based positioning systems. Figure 1 shows the data flow of an MR-based positioning system. LBSs generate low sampling OTT GPS locations (green dots). With the OTT GPS locations as label data, the MR-based positioning system can train the high sampling MRs by using machine learning models. Since the GPS locations are numeric data, we can adopt the classic machine learning algorithm named Random Forests to solve a regression problem. When the training model is ready, given the MR records without labels, we predict the GPS locations (yellow dots) with respect to such MR records. In this way, with the predicted GPS points, we can fully recover the entire trajectory.

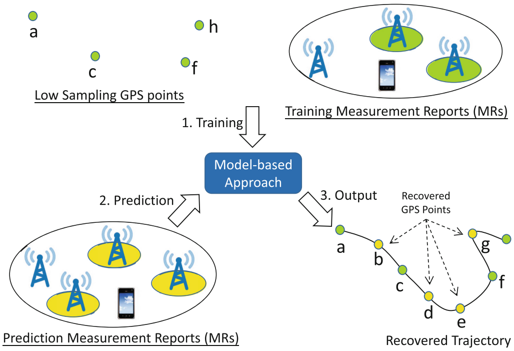


Fig. 1. Model-based MR positioning systems (Color figure online)

Suppose that the above MR records have ground truth (i.e., the GPS locations), we can measure the positioning precision by comparing the predicted location to the ground truth GPS location. The previous work [10] can achieve a mean error of around 80 m. Though the positioning precision obtained by applying the RF algorithm is much better than the traditional telecom positioning approaches, it cannot compete with GPS. The main purpose of this paper is to present the methodology that can be applied to the estimated GPS locations by applying RF to achieve more accurate estimated locations.

2.1 Positioning Errors

With help of road network maps, we are able to observe the predicted points on the maps and two types of positioning errors as follows:

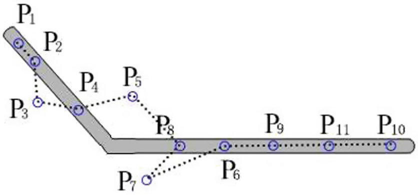


Fig. 2. Noise error points in a trajectory

As the Fig. 2 shows a trajectory consists of eleven points with noise.

- Horizontal error
  Horizontal error is not a simple error in latitude direction, and it mainly represents the predicted locations originally close to the road but now are far away from the true locations. In Fig. 2, there are three such examples: p_3, p_5, p_7. What we need to do is pull these errant points back to the road network.
- Vertical error
  Vertical error mainly represents the predicted location points in wrong sequences, although they are distributed on the correct road network. From the common sense, a human walking/driving trajectory will usually not appear in repeated crossing. The Fig. 2 depicts there are eleven points in the whole walking trajectory, and it should be expressed in the sequential order: $p_1 \rightarrow p_{11}$ according to the experience knowledge, but point p_8 appears between p_5 and p_7, and so point p_{11} does the same. We define this errant sequence as the vertical error.

To solve the above vertical and horizontal errors, we are going to leverage the road network maps and multiple consecutive GPS points as the context information to improve the positioning precision.

3 Telco Localization Solution Introduce

In this section, we first give an overview of several solutions that can correct errors in predicted locations [3].

3.1 Kalman Filter

Review of Kalman Filter. Before using Kalman Filter (KF) [1] as a tool to improve the positioning accuracy in our problem, we would like to first give a quick review of KF. More specifically, KF [7] mainly consists of two main parts: one is the state Eq. 1 and the other is the observation Eq. 2. The KF model assumes the true state at time k is evolved from the state at $(k-1)$ as Eq. (1) states.

$$\widehat{x}_k = Ax_{k-1} + Bu_k + w \tag{1}$$

where

- A is the system state parameter, which is the transition model applied to the previous state x_{k-1};
- B is the control-input model that is applied to the control vector u_k and can be ignored in this paper;
- w is the processing noise that is assumed to be zero mean Gaussian white noise, with covariance Q;
- At time k an observation (or measurement) z_k of the true state x_k is obtained according to the following observation equation.

$$z_k = Hx_k + v \tag{2}$$

where

- z_k is the observation result;
- H is the observation matrix;
- x_k is the true state value in its system;
- v is the observation noise that is assumed to be zero mean Gaussian white noise with covariance R (In this paper, we assumed that this covariance as well as Q won't be altered with the system state dynamically).

The updating equations from time $k-1$ to k are as follow.

$$\widehat{X}_k = AX_{k-1} + BU_k \tag{3}$$

$$\widehat{P}_k = AP_{k-1}A^T + Q \tag{4}$$

$$K_k = \widehat{P}_k H^T (H\widehat{P}_k H^T + R)^{-1} \tag{5}$$

$$\widehat{X}_k = \widehat{X}_k + K_k(Z_k - H\widehat{X}_k) \tag{6}$$

$$P_k = (1 - K_k H)P_k \tag{7}$$

where

- K_k is the Kalman gain at time k;
- P_k is the error covariance at time k.

Algorithm 1. Kalman Filter Algorithm

```
Input: Input GPS points ⟨ longitude, latitude, timestamp ⟩
Output: Corrected GPS points ⟨ longitude, lattitude, timestamp ⟩
1: set initial point and speed;
2: set noise $Q$ and observation noise $R$;
3: for each input GPS point do
4:    compute the predicted position by Eq. 1;
5:    update the current position by Eq. 2;
6: end for
7: return points = sum($point_k$) $k$ = 1,2,3.....;
```

Kalman Filter-Based Correction Algorithm. Algorithm 1 shows the overall Kalman filter procedure referred to two core equations as introduced before (1 and 2) to process the obtained training data from telco big data platform. First, we need to set a necessary initial data point and its speed according Kalman equations (line 1) as well as two noise sets Q and R noise (line 2). Second, for each sequential point (line 3) of the whole trajectory we should apply Kalman Filter to evaluate their true GPS values (lines 4 and 5, where we apply the series of equations (3 to 7) to compute the real time estimated values and update two kinds of noise mentioned above). Finally, the algorithm aggregates all evaluated points to form the original sequence (line 6). Equations 3 and 4 are to project the state and error covariance ahead, then compute kalman gain in Eqs. 5 and 6 and 7 update the estimation with measurements and error covariance.

3.2 Mean Filter-Based Correction

Recall that the KF-based correction algorithm does not fully leverage context information. In order to resolve the issue where the existing model-based algorithms are unable effectively to deal with the abnormal of predicted positions and increase the positioning accuracy, we borrow the idea of the mean filter [5] that has been applied in the imagery data processing and design a context-aware correction algorithm based on the GPS points inside a sliding window. The methodology will be applied to the post-processing of the predicted trajectory. Specifically, for a measured point position x_i, the estimate value of this point is the mean of its $n/2$ successive GPS points and $n/2$ proceeding GPS points, where n is the size of a given sliding window.

$$\hat{x}_i = \frac{1}{n}\left(\sum_{k=i-n/2}^{i+n/2} x_k\right) \tag{8}$$

In the above equation, $\hat{x}_i$ is the estimate of x_i. To ensure that the mean filter-based correction algorithm work, we first need to preprocess the input GPS points with the equal interval interpolation. For the given input GPS points, we first find the minimum time interval between any two continuous points. Next, based on the minimal time interval, we will obtain these consecutive points with the time interval greater the minimal one, and fill the missed GPS points by the median interpolation.

In addition, we note that the mean filter is sensitive to the outliers contained in the input GPS points. To resolve this issue, we would like to find those outliers and remove them from the input GPS points. To find such outliers, we use a classical median filter.

Algorithm 2. Mean Filter-based Correction

```
Input: Input GPS points ⟨ longitude, latitude, timestamp ⟩, and window size $n$
Output: Corrected GPS points ⟨ longitude, lattitude, timestamp ⟩
1: load the $S$ input GPS points;
2: preprocess the GPS points by equal interval interpolation with help of a minimum time interval.
   eliminate outliers by using median filter;
3: for each $i$ in size of window do
4:   for each of the $n$ input GPS point do
5:     moving equalize point with the Eq. 3;
6:   end for
7: end for
8: points = sum($point_k$) $k$ = 1,2,3.....;
```

The body of this Algorithm 2 mainly depends on the Eq. 8 we introduced at the beginning of this section.

3.3 Map Matching-Based Correction

Map-matching [2,8] is to match the recorded geographic coordinates (such as collected GPS points) to a logical model of the real world. It has been developed as a very mature technique combining digital map with locating information, for example, to obtain the real position of vehicles in a road network.

We plug the map-matching technology to Algorithm 2 to filter out those GPS outliers during the preprocessing and postprocessing phases. First, in the preprocessing step in line 2 of Algorithm 2 it can plug in the map matching technique to make sure every input GPS point is on the correct road. In this way, we can remove the outliers in the input points. Second, even after Algorithm 2 is performed, it is still possible that some corrected GPS points (i.e., the output of Algorithm 2) might not be on the roads. Thus, we can again apply the map-matching technique to these corresponding GPS points to acquire the final corrected points or locations. In this case, Algorithm 2 can work together with the map-matching technique to improve the positioning accuracy significantly.

4 Evaluation

In this section, we compare the performances of three models: KF, mean-filter (MF) and map-matching (MF+MM) via the computational experiments. In order to measure the positioning precision, we first apply the model-based (random tree) positioning approach to derive the recovered or estimated GPS points from MR data. After that, we employ these models to correct the recovered GPS points. Based on the corrected GPS points by these three approaches, we then measure the positioning precision of each model. To perform the computational experiments, we use a real dataset of user mobility trajectories in one day (containing around 600,000 MR records) collected from a telecom service provider in the city of Shanghai, China.

4.1 Performance Comparison

In addition to the measurements of the positioning precisions of individual models mentioned above, we also include the positioning precision for the recovered GPS points as the baseline. We use two metrics, namely, the mean error and median error as shown in Table 1 to present the positioning precision.

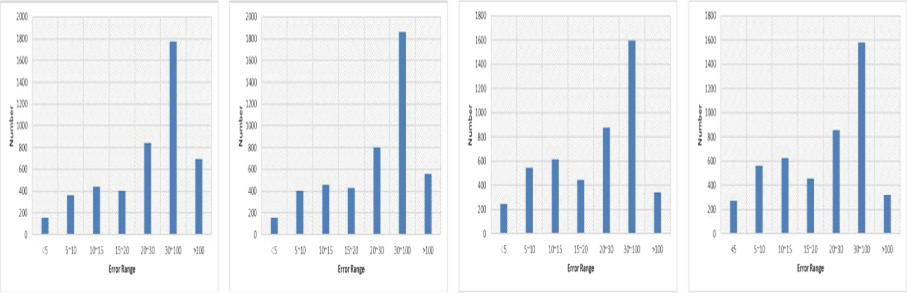
Table 1. Comparison of recovered and corrected GPS points

Metric	Median (meters)	Mean (meters)
Recovered points	31.8075	56.6148
Corrected points by KF	31.2072	48.7996
Corrected points by MF	25.4192	39.3617
Corrected points by MF+MM	23.1236	31.2681

In Table 1, KF model slightly improves the recovered points' positioning precision. For example, the mean error is reduced from 56.6148 m to 48.7996 m and yet the median stays almost the same. From the table it can be seen that the median has a very little reduction from 31.8075 m to 31.2072 m. Albeit MF improves the positioning precision, MF+MM greatly increases the positioning precision with around 20.08% and 30.47% reductions in the mean and median errors respectively. The numbers in Table 1 clearly verify the superiority of MF+MM in terms of providing the positioning precision.

Next, we plot the error distribution of the recovered and corrected GPS points in Fig. 3. The x-axis represents the error range and the y-axis represents the error distributions in the different error ranges. Figure 3(a) shows the original error distribution without any error correction process. Figure 3(b) depicts the error distribution of each recorded point that is simply corrected by the KF model. Nevertheless, the resultant error distribution is very similar to the original one. The prerequisite of classic Kalman filtering is to establish an accurate dynamic model and observation model, and it needs more clearly understanding of the moving object [4]. But in this paper, we assume that the observation equation is linear and with the stable noise.

Figure 3(c) represents the error distribution for the results obtained by MF mode. This algorithm has a relatively good effect for the wide range of error



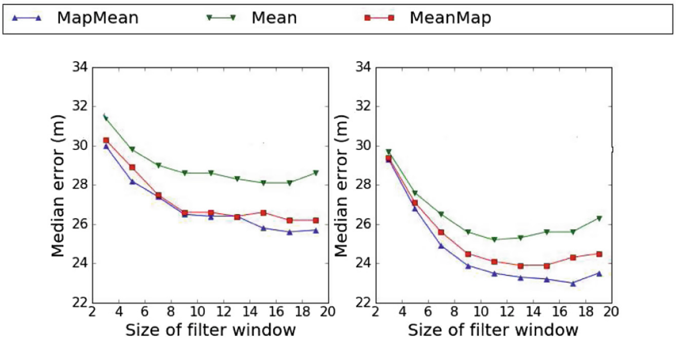
a) initial recovered points (b) KF model (c) MF model (d)MF+MM model

Fig. 3. Whole error distribution

correction because it is more inclined to smoothing trajectory based on the empirical knowledge of human motion behavior and maintains a sustained and stable state for a short time of period. It can be seen from Fig. 3(c) the number of points with the biggest errors has reduced dramatically compared to the original one. Similarly, this approach also helps correcting the errors for these locations around the point that has relatively big error, and these locations usually can be easily affected by this outlier. Figure 3(d) reflects the combined effect about MF and Map matching. From the picture we are able to recognize that this approach is more effective in reducing errors after introducing map-matching method.

4.2 Sensitive study of KF+MM model

We can use the map-matching in both pre-processing and post-processing phases. To study the sensitivity of the KF+MM model, we vary the window size and measure the median error of the KF+MM model. In addition, we are interested



(a) without median filter (b) with median filter

Fig. 4. Error curve changing window size

in the effect of the median filter to filter out outliers in the preprocessing phase (i.e., line 2 of Algorithm 2). For the comparison purpose, we also evaluate the median errors of (1) MF model alone, (2) MF model enhanced by the MM-based preprocessing, and (3) MF model enhanced by the MM-based postprocessing.

In Fig. 4, three different lines represent three algorithms according whether adopt map-mating and the order adopted map-matching. Figure 4(a) and (b) provide the median errors of above three models without and with applying the median filter respectively. As it is shown in the figure, the models adopting the median filter can usually achieve much lower median errors than the one without the median filter.

When comparing these three models no matter in the preprocessing phase or the postprocessing one, two MF models enhanced by MM have much lower median errors than the MF model alone. Moreover, we find that the MF model with the MM-based preprocessing out performs the one with the MM-based postprocessing. It is because the MM-based preprocessing can clean the outliers without affecting the MF model. In MF model with MM-based postprocessing, the outliers still appear in the input of the MF model and could impact the overall precision of the KM model negatively.

4.3 Visualization of Positioning Models

Figure 5 illustrates the recovered GPS points and corrected points or locations on a real road network. First in Fig. 5(a), there exist many points that are not on roads. It is mainly because the two-layer random forests (RFs) use the center points of those leaf nodes in the RFs as the predicted GPS points, no matter the center points are on the roads or not. Second, Fig. 5(b) demonstrates that MF model obviously is able to smooth the trajectory. Nevertheless, in our dataset, many GPS points appear on the overpass and underpass that occur on the same street segments. It is hard for the MF model to put them onto the correct roads (because the roads on the overpass and underpass share the same longitude and latitude coordinates). However, Fig. 5(c) demonstrate that this problem can be

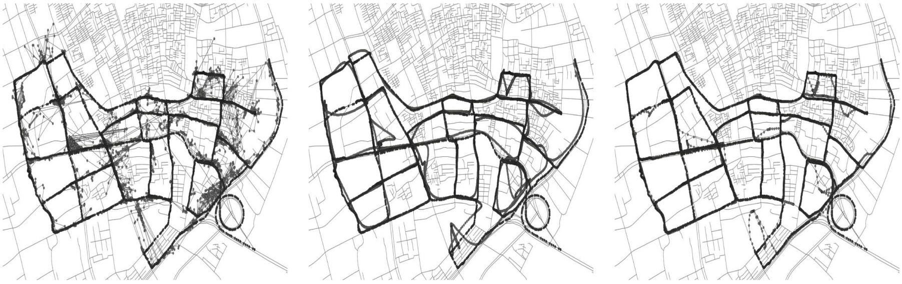

(a) original trajectory
(b) corrected trajectory by MF
(c) corrected trajectory by MF+MM

Fig. 5. Trajectory error comparison

overcome by applying the MM technique and as the result it produces the best positioning precision.

5 Conclusion

In this paper, we propose the methodology to leverage context information in order to correct the estimated or recovered city-scale localization errors by applying model-based localization methods. By adopting the powerful techniques including Kalman Filter, Mean filter, and Map-matching, the proposed approaches can greatly improve the positioning precision.

As the future work, we will continue work on improving the positioning precision. For example, we are planning to explore the regularity patterns from the recovered trajectories for further postprocessing. In addition, beyond the regression model-based prediction algorithm, we are interested in other advanced machine learning techniques to replace the currently used random forest models.

Acknowledgements. The authors would like to appreciate Professor Weixiong Rao for his critical and useful suggestions. The authors also want to thank two anonymous reviewers for their critics and suggestions that help improving the quality of our paper.

References

1. https://en.wikipedia.org/wiki/kalman_filter
2. https://en.wikipedia.org/wiki/map_matching
3. Anderson, J.L.: Exploring the need for localization in ensemble data assimilation using a hierarchical ensemble filter. Phys. D: Nonlinear Phenom. **230**(1), 99–111 (2007). Data Assimilation
4. Anderson, J.L.: Localization and sampling error correction in ensemble kalman filter data assimilation. Mon. Weather Rev. **140**(7), 2359–2371 (2012)
5. Dawoud, N.N., Samir, B.B., Janier, J.: N-mean kernel filter and normalized correlation for face localization, pp. 416–419 (2011)
6. Dil, B., Dulman, S., Havinga, P.: Range-based localization in mobile sensor networks. In: Römer, K., Karl, H., Mattern, F. (eds.) EWSN 2006. LNCS, vol. 3868, pp. 164–179. Springer, Heidelberg (2006). doi:10.1007/11669463_14
7. Evensen, G.: The ensemble kalman filter: theoretical formulation and practical implementation. Ocean Dyn. **53**(4), 343–367 (2003)
8. Lou, Y., Zhang, C., Zheng, Y., Xie, X., Wang, W., Huang, Y.: Map-matching for low-sampling-rate GPS trajectories, pp. 352–361 (2009)
9. Yuan, M., Deng, K., Zeng, J., Li, Y., Ni, B., He, X., Wang, F., Dai, W., Yang, Q.: OceanST: a distributed analytic system for large-scale spatiotemporal mobile broadband data. PVLDB **7**(13), 1561–1564 (2014)
10. Zhu, F., Luo, C., Yuan, M., Zhu, Y., Zhang, Z., Gu, T., Deng, K., Rao, W., Zeng, J.: City-scale localization with telco big data. In: Proceedings of the 25th ACM Conference on Information and Knowledge Management, CIKM 2016, 24–28 October 2016, Indianpolis, USA (2016)