# Accurate and Efficient Map Matching for Challenging Environments

Reham Mohamed
Dept. of Computer and Systems Engineering
Alexandria University, Egypt
reham.mohmd@alexu.edu.eg

Heba Aly
Dept. of Computer and Systems Engineering
Alexandria University, Egypt
heba.aly@alexu.edu.eg

Moustafa Youssef
Wireless Research Center
Alexandria University and E-JUST, Egypt
moustafa.youssef@ejust.edu.eg

## ABSTRACT

We present the *SnapNet*, a system that provides accurate real-time map matching for cellular-based trajectories. Such coarse-grained trajectories introduce new challenges to map matching including (1) input locations that are far from the actual road segment (errors in the orders of kilometers), (2) back-and-forth transitions, and (3) highly sparse input data. *SnapNet* addresses these challenges by applying extensive preprocessing steps to remove the noisy locations and to handle the data sparseness. At the core of *SnapNet* is a novel incremental HMM algorithm that combines digital map hints and a number of heuristics to reduce the noise and provide real-time estimation. Evaluation of *SnapNet* in different cities covering more than 100km distance shows that it can achieve more than 90% accuracy under noisy coarse-grained input location estimates. This maps to over 97% and 34% enhancement in precision and recall respectively when compared to traditional HMM map matching algorithms. Moreover, *SnapNet* has a low latency of 1.2ms per location estimate.

## Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

## Keywords

Map-matching, Cellular-based trajectories, HMM, Challenging environments

## 1. INTRODUCTION

Map matching, the problem of mapping a set of coordinates with errors to the corresponding points on the road network, has been widely used in location based applications including car navigation, directions finding, car heading estimation, traffic analysis, among others. A large number of map matching techniques [9, 10] have been proposed in literature, all based on GPS as the ubiquitous localization

*SIGSPATIAL'14*, November 04 - 07 2014, Dallas/Fort Worth, TX, USA
Copyright 2014 ACM 978-1-4503-3131-9/14/11$15.00
http://dx.doi.org/10.1145/2666310.2666429.

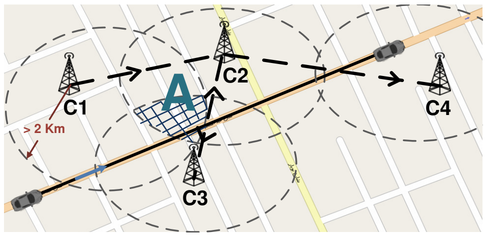

Figure 1: Example of the different challenges of cellular-based locations map matching: Reported location is typically far from the correct road segment, location update rate is low, and there are back-and-forth jumps (The user is moving straight but her location estimates sequence is C1-C2-C3-C2-C4 due to the overlapping coverage between the cell towers in Area *A*.

technology for outdoors. However, since GPS is an energy hungry device and may not be available everywhere, a number of map matching techniques have been proposed that can handle a lower sampling rate and lower localization accuracy. These techniques usually depend on using other phone sensors such as WiFi [11] or inertial sensors [13]; or do extensive war driving effort assuming the availability of neighbouring cell tower information [12][1].

In this paper, we argue that there is a growing number of map matching applications that cannot assume the existence of GPS or other phone sensors. In particular, traffic estimation from the cellular provider-side, low-energy GPS-less localization [2, 3, 6–8], applications that involve low-end phones with no GPS or other sensors (e.g. in developing countries), and crowd-sensing applications in which GPS is usually turned off are examples of such applications [1]. In such scenarios, the typical assumptions of traditional map matching algorithms do not hold. Specifically, as shown in Figure 1, the phone location is no longer near the actual road segment, the location error is in the order of kilometers, there are back-and-forth transitions due to changing the cell tower association, and the input location samples are highly sparse. These harsh constraints lead to a much harder map matching problem, in terms of quality of input points and number of candidate road segments, affecting both the accuracy and computational complexity.

We therefore present *SnapNet*, a system for accurate and efficient map matching of challenging environments. At the core of *SnapNet* is a *novel* incremental Hidden Markov Model (HMM)-based algorithm that takes into account the noise of the input data as well as digital map hints to enhance

[1]The majority of cell phones in the market only give access to the associated cell tower information with no access to the neighboring cell tower information.

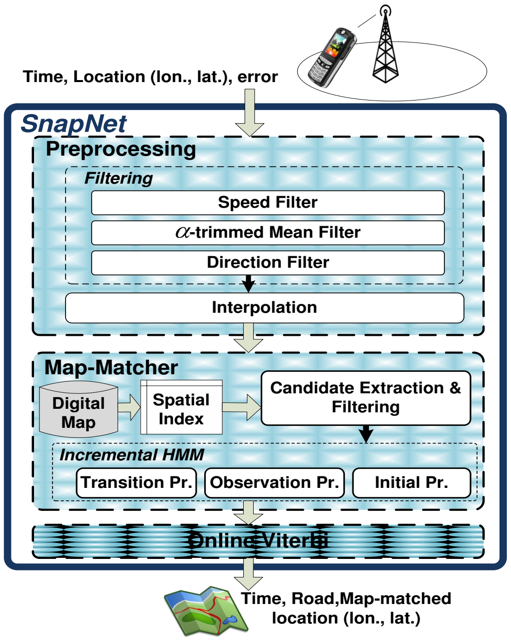

Figure 2: *SnapNet* system architecture.

the accuracy of the estimated road segments and efficiently handles the increased number of candidate road segments. This HMM is also combined with a number of preprocessing modules that reduce the noise and spareness in the input data.

We evaluated *SnapNet* on driving traces collected in multiple cities. Our results show that *SnapNet* can achieve 90% accuracy in identifying the correct road segment. This comes with a low latency in computation of 1.2ms per location.

## 2. THE SnapNet SYSTEM

Figure 2 shows an overview of the *SnapNet* architecture. The input to *SnapNet* is a time-stamped coarse-grained location. Each location is represented by its latitude and longitude as well as an estimate of the localization error.

*SnapNet* map-matches the user trajectory incrementally in real-time. It starts by two preprocessing steps: filtering and interpolation. A series of consecutive filters are applied to the raw data to eliminate the noise. However, this makes the traces much more sparse. Therefore, we apply interpolation on the filtered data to reduce this sparseness.

The filtered and interpolated points are then passed to the map matcher that employs a novel HMM algorithm to handle the noisy location data efficiently. Specifically, our map matching algorithm contains two sub-modules: Candidate Extraction and Filtering, and Incremental Map Matching. The *Candidate Extraction and Filtering* module determines the candidate road segments from OpenStreetMaps. The *Incremental Map Matching* algorithm integrates a number of modifications to the standard HMM algorithm to take the quality of the input data into account as well as digital map hints to enhance the accuracy of the estimated road segments. Finally, the *Online Viterbi Algorithm* efficiently calculates the probabilities of the different road segments and determines the most probable road segment. The Map Matcher outputs the matched road segment along with the estimated location on it.

### 2.1 Filtering Modules

We apply three consecutive filters to detect noisy transitions: the *Speed filter*, the *$\alpha$-trimmed filter*, and the *Direction filter*.

**The Speed Filter** assumes that the user does not exceed a certain speed threshold as her speed is limited by many factors such as the vehicle maximum speed.Therefore, if the user's estimated current speed exceeds this threshold, her current location estimate is detected as an outlier.

The speed between any two points can be calculated by dividing the geodesic distance between the two points by the difference in their time-stamps. However, due to the high error in the input locations, this speed is noisy to use. Instead, we estimate the user's current speed by averaging the speed between this location and a window of the preceding unfiltered locations.

**The $\alpha$-trimmed Mean Filter** [4] aims to reduce the back-and-forth transitions in the input data. An $\alpha$-trimmed filter has the advantage of handling both impulse and Gaussian noise, as compared to mean and median filters that can handle only one of them.

The basic idea is to look at the neighbors of each point, remove $2\alpha$ of the extreme neighbors, i.e. outliers, then replace the point by calculating the mean of the unfiltered neighbors.

**The Direction Filter** further reduces the back-and-forth transitions which lead to a change in the user direction, i.e. as if the user made a u-turn. It ensures that the change in the user's direction is only allowed when we are sure that it is originating from an actual change in direction, not due to noise in the location data. To do that, if a location point indicates a direction change, we cache the point temporarily till we get a new location point. If the new location confirms the direction change, we add the two points to the pre-processed trace. Otherwise, we drop the cached point.

### 2.2 Interpolation

After applying the filters on the raw input location points, the sparseness of the data points increases. To overcome this, we apply linear interpolation on the unfiltered location points when needed. We add interpolated points at equally-spaced distances between the actual location points when the distance between them exceeds a threshold. We found that, in most cases, the interpolated points will match with the shortest path between the true points on the road network when using a step of $50m$ without performing expensive shortest path computations.

## 3. THE SnapNet MAP MATCHER

The input to the *Map Matcher* module are the filtered and interpolated location data as well as the digital map. We model the map matching problem as a Hidden-Markov Model (HMM). Our extended incremental HMM can effectively fuse the noisy input location data and the provided road network constraints in a sound way to provide accurate map matching. The *Map Matcher* module has two submodules: the *Candidate Extraction and Filtering* module and the *Incremental Map Matching* module.

### 3.1 Candidate Extraction and Filtering

For each input location ($\text{lat}_t$, $\text{lon}_t$, $\text{err}_t$) at time $t$, we extract all candidate road segments that intersect with the circle centered at ($\text{lat}_t$, $\text{lon}_t$) with radius $\text{err}_t$ using the digital map. To speed up the candidate extraction, we build an R-tree spatial index on all possible road segments in the road network. We further filter any candidate segment that is not connected to at least one of the candidate segments from the previous estimation step. The unfiltered road segments represent the candidates for the current observation. An observation point with an empty candidates set is considered an outlier.

## 3.2 Incremental Map Matching

*SnapNet* employs an extended HMM to address the noisy input data. In particular, we provide (a) dynamic parameter estimation based on the noise in the input data, and (b) apply a heuristic for road transitions to handle the false transitions.

Traditional HMM-based map matching approaches use a window of input locations to estimate the corresponding map matched locations. This technique leads to a good accuracy, however, it increases the latency of the estimated location as the system has to wait for a full window of samples before it can produce the output. This is even worse with the coarse-grained and sparse cellular location information as the number of candidate road segments is high, leading to more calculations per iteration. To address this latency issue, *SnapNet* uses an incremental map matching approach, where a sliding window is used for estimation: at each time instant, a new location sample is added and the oldest sample is removed from the window.

We start this section by providing the mathematical model and notations followed by the details of how *SnapNet* estimates the different model parameters.

### 3.2.1 Mathematical Model and Notations

The input to the *Map Matching* module is a set of $T$ cellular locations $Z = (z_1, ..., z_T)$, where $z_t = (lat_t, lon_t, err_t), 1 < t < T$, represents the location information at time $t$. Let $S_t = \{s_{1,t}, s_{2,t}, ..., s_{N_t,t}\}$ be the set of possible states, i.e. road segments, at time $t$ obtained by the *Candidate Extraction and Filtering* module, where $N_t = |S_t|$.

For each road segment $i$, the *Map Matching* module uses two probability distributions:

1. The state transition probability distribution between road segments $i$ and $j$, $A = \{a_{ij}\}$, where $a_{ij} = P[s_{j,t}|s_{i,t-1}]$
2. The observation probability distribution in state $i$, $B = P[z_t|s_{i,t}]$, i.e the probability of observing an input location given the user is actually on road segment $i$.

In addition, the module calculates the initial state distribution $\pi = \{\pi_i\}$, where $\pi_i = P[s_{i,1}]$. Therefore, the problem becomes, given a sequence of location observations ($Z$), we want to find the most probable sequence of road segments (states) $Q = (q_1, ..., q_T)$, where each $q_t \in S_t, 1 < t < T$.

### 3.2.2 Observation Probability

The observation probability $p(z_t|s_{i,t})$ is the probability that state (i.e. road segment) $s_{i,t}$ emits the observation (i.e. input location) $z_t$. To estimate the observation probability, we take it as a function of the distance between the observed location and the projected location on the corresponding road segment. The intuition is that the closer the observed location to the road segment, it is more probable that this road is the user's actual segment. It has been shown that this distance can be modeled accurately using a Gaussian distribution for GPS trajectories [5]. Therefore, we extend the Gaussian model in literature to fit our problem. Specifically, the observation probability in *SnapNet* is modeled as:

$$p(z_t|s_{i,t}) = \frac{1}{\sqrt{2\pi}\sigma_t} e^{-0.5\left(\frac{\text{dist}(z_t, s_{i,t})}{\sigma_t}\right)^2} \quad (1)$$

where $\text{dist}(z_t, s_{i,t})$ is the geodesic distance between $z_t$ and $s_{i,t}$, and $\sigma_t$ is the standard deviation of a Gaussian random variable that corresponds to the error in the sensor measurements. *SnapNet* has two novel contributions to this observation probability model compared to literature: First, instead of using a fixed value for $\sigma_t$, and since the error in the input location in our case varies with time, e.g. based on the range of the associated cell-tower, we estimate the value of sigma dynamically based on the average error of the input locations over a time window ($w = 10$). Second, to further handle the back-and-forth effect and errors in the digital map, we also consider the relation between the direction of the candidate road segment and the input trajectory: roads in the opposite direction of the input trajectory are assigned a small observation probability.

### 3.2.3 Transition Probability

The transition probability is the probability of moving to the next road segment (state $s_{j,t}$) given that the current state is road segment $s_{i,t-1}$. The transition probability can be estimated as a function based on the difference between the distance on the input trajectory and the distance between the projections on the road segments. The smaller this distance is, the larger the transition probability should be. We assume that this distance difference ($d_t$) has an exponential distribution [10]. However, due to the noisy location input and using an online incremental algorithm, it gives a large number of false transitions. To reduce this abrupt transitions, we bias the transition probability to favor staying on the current road. This is achieved by weighting the exponential transition probability according to whether the transition is on the same road segment or not. Therefore, the transition probability used in *SnapNet* is:

$$p(s_{j,t}|s_{i,t-1}) = \begin{cases} p_{\text{same}}.\frac{1}{\beta}e^{-\frac{d_t}{\beta}} & \text{when } s_{j,t} = s_{i,t-1} \\ (1 - p_{\text{same}}).\frac{1}{\beta}e^{-\frac{d_t}{\beta}} & \text{otherwise} \end{cases} \quad (2)$$

Where $p_{\text{same}} \geq 0.5$ is the probability (weight) of staying on the same road between transitions.

## 4. EVALUATION

We tested *SnapNet* on real data traces collected from two cities in Egypt. Our data consists of 100 km of car driving, collected using different Android phones including Samsung Galaxy S3, S4, and S5 phones as well as an LG Nexus 4.

To get low-accuracy network-based location estimates, we use the Google Android Location API with *WiFi turned off*. This way, the returned location is based on the associated cell tower information only as the used phones do not report the neighboring cell towers. This provides a coarse-grained location accuracy. For ground truth, we also collected accurate location data at high sampling rate using a GPS/GLONASS receiver. We further apply map matching to this high accuracy data to obtain the actual driving path. It has been shown in literature that map matched GPS traces with a one second sampling rate almost yields

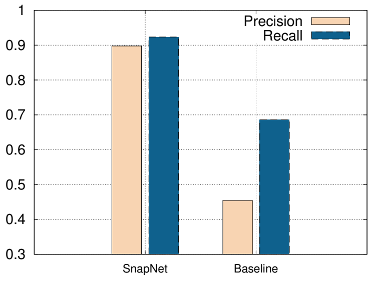

Figure 3: Accuracy comparison between *SnapNet* and the baseline HMM map matching technique.

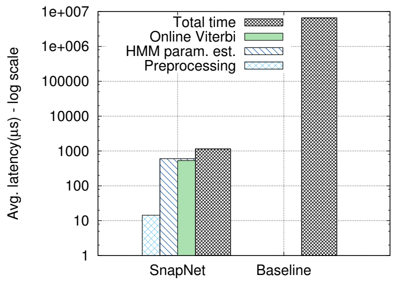

Figure 4: Running time of the *SnapNet* components.

the ground truth path [10].

To evaluate *SnapNet*, first we find the common matching sequence of roads $X$ between the map matched output trajectory $Y$ and the ground-truth trajectory $G$. Then, we compute two evaluation metrics: precision and recall [10]. Precision is defined as the ratio between the distance traversed on the matching sequence $X$ and the total distance of the map matched trajectory $Y$. Recall is defined as the ratio between the distance traversed on the matching sequence $X$ and the total distance of the ground truth trajectory $G$. In the following subsection, we show the effect of the overall system performance of *SnapNet* and compare it with a baseline HMM map matching technique.

### 4.1 System Performance

We compare the *SnapNet* accuracy with a baseline technique that uses a plain HMM map matching without any of the additional filters or our HMM modifications. Figure 3 shows that *SnapNet* achieves over 97.4% enhancement in precision and over 34.6% enhancement in recall, highlighting the combined advantage of the different modules of the system. Figure 4 shows the average latency per input location for each module (log scale). The figure shows that the overall latency of *SnapNet* is about 1.2 ms. Most of the running time is consumed by the HMM parameters estimation and the online Viterbi algorithm. This is due to the large error of the input network-based locations, which results in a large number of candidates. Compared to the window-based baseline technique, the online Viterbi algorithm and the incremental HMM module reduce the latency by more than three orders of magnitude.

## 5. CONCLUSION

We presented *SnapNet*, a real-time map matcher for noisy cellular-based trajectory traces. *SnapNet* is unique in targeting mainstream cell phones which can only provide its associated cell-info for localization systems; opening up new possibilities for location-based services. We provided the *SnapNet*'s system architecture and the different modules that help it reduce the noise and sparseness in the input data, and leverage the digital map information to enhance the estimation accuracy while maintaining real-time operation. An evaluation of *SnapNet* using real-traces, shows that *SnapNet* can achieve a map matching precision and recall of more than 90% with an average running time of 1.2ms.

## Acknowledgment

This work was supported in part by a grant from the Egyptian Information Technology Industry Development Agency (ITIDA).

## 6. REFERENCES

[1] H. Aly, A. Basalamah, and M. Youssef. Map++: A crowd-sensing system for automatic map semantics identification. In *SECON*. IEEE, 2014.
[2] H. Aly and M. Youssef. Dejavu: an accurate energy-efficient outdoor localization system. In *SIGSPATIAL*. ACM, 2013.
[3] M. Alzantot and M. Youssef. Uptime: Ubiquitous pedestrian tracking using mobile phones. In *Wireless Communications and Networking Conference (WCNC), 2012 IEEE*, pages 3204–3209, 2012.
[4] J. Bednar and T. Watt. Alpha-trimmed means and their relationship to median filters. *IEEE Trans Acoust Speech Signal Process*, 1984.
[5] C. Y. Goh, J. Dauwels, N. Mitrovic, M. Asif, A. Oran, and P. Jaillet. Online map-matching based on hidden Markov model for real-time traffic sensing applications. In *ITSC*. IEEE, 2012.
[6] M. Ibrahim and M. Youssef. Cellsense: a probabilistic RSSI-based GSM positioning system. In *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pages 1–5. IEEE, 2010.
[7] M. Ibrahim and M. Youssef. A hidden Markov model for localization using low-end GSM cell phones. In *Communications (ICC), 2011 IEEE International Conference on*, pages 1–5. IEEE, 2011.
[8] M. Ibrahim and M. Youssef. Cellsense: An accurate energy-efficient GSM positioning system. *Vehicular Technology, IEEE Transactions on*, 61(1):286–296, 2012.
[9] J. Krumm, J. Letchner, and E. Horvitz. Map matching with travel time constraints. In *SAE World Congress*, 2007.
[10] P. Newson and J. Krumm. Hidden Markov map matching through noise and sparseness. In *SIGSPATIAL*. ACM, 2009.
[11] A. Thiagarajan, L. Ravindranath, and et al. VTrack: accurate, energy-aware road traffic delay estimation using mobile phones. In *SenSys*. ACM, 2009.
[12] A. Thiagarajan, L. Ravindranath, and et al. Accurate, low-energy trajectory mapping for mobile devices. In *NSDI*. USENIX, 2011.
[13] H. Wang, Z. Wang, G. Shen, F. Li, S. Han, and F. Zhao. Wheelloc: Enabling continuous location service on mobile phone for outdoor scenarios. In