An HMM-based Map-Matching Method with Cumulative Proximity-Weight Formulations

Ali Oran
Future Urban Mobility IRG
Singapore-MIT Alliance for Research & Technology (SMART)
Singapore, 138602
Email: aoran@smart.mit.edu

Patrick Jaillet
Department of Electrical Eng. & Comp. Science (EECS)
Massachusetts Institute of Technology
Cambridge, MA 02139 USA
Email: jaillet@mit.edu

Abstract—**In this study, we introduce a hidden Markov model map matching method that is based on a cumulative proximity-weight formulation. This new formula is based on the line integral of point-wise segment weights rather than the almost standard shortest distance based weights. The proposed method was tested using vehicle and map data from Seattle area. Several simulations were conducted so as to have a clear comparison of the new weight to the traditional one; and particular emphasis were given to matching of GPS data with long sampling periods and high level noise. Overall, possible improvements to MM accuracies by the new weight were identified. It was seen that the new weight could be a better option than the shortest distance based weight in the presence of low-frequency sampled and/or noisy GPS data.**

I. Introduction

Usage of satellite-based navigation devices have been increasing steadily since the beginning of 2000s. Therefore, the analysis of vehicle position measurements collected by these devices with an underlying road network to identify vehicles' true locations has become a major research problem under the generic name of Map Matching (MM). As a result of the interdisciplinary nature of MM problem, in the last two decades, numerous methods with very different approaches have been proposed. Early MM methods have been developed with the assumption that frequently sampled GPS points[1], e.g. sampled at 1 second intervals, would be available. In the presence of such data, most methods from the last decade have been able to yield almost perfect matching results. However, these methods have not fared well in general when the sampling period between data points were long, especially when it exceeded 30 seconds. For this reason, in the last few years there has been an increasing interest for methods that can address this non-traditional MM problem of low-frequency GPS data. The major reason for the drop in accuracy comes from the increasing uncertainty regarding the travel of a vehicle from one point to the next one when measurement frequency is low [1]. To reduce the effect of increased uncertainty, various heuristics have been proposed. While these heuristics have improved MM accuracies to some degree, more research is still needed to have highly accurate methods for low-sampled GPS data. A second thing to note is that the accuracy of an MM method can also vary depending on the chosen network's road sparsity, and the GPS device's measurement quality. While it is possible to get almost 100% accuracy with a very accurate GPS receiver, while traveling on a sparse network free of urban canyons, such high accuracy cannot be expected in every problem. For this reason, there has been an additional recent interest in developing methods that would be robust to noisy GPS measurements. Any proposal that can address these two problems, while being not particularly dependent on a specific underlying MM algorithm, and not relying on extra data (using only positioning data), can benefit MM methods in general.

For MM methods, some concepts always remain essential; and the above goal could be achieved by improving the measures related to these. The proximity concept of spatial closeness between observed GPS points and road segments in the network is one of these. In deterministic methods, it has been quantified as *Proximity-Weight* e.g. [2], [3], and in probabilistic methods, as *Observation (Emission) Probability* e.g. [4], [5]. While formulations for segment proximity have been unique in each method, all formulations still have followed the fact that for a given GPS point, segments closer to this point are more likely to be the true segment compared to segments further away [6]. Therefore, in deterministic methods, monotonically decreasing functions of distance, and in probabilistic methods, likelihood functions of distance defined through the Gaussian distribution are most common. Meanwhile, it is interesting to note that in all these functions, that quantify the proximity of a segment with respect to a given GPS point, the shortest distance from the point to the road segment have been used as the sole argument [2], [3], [4], [5], [7], [8], [9], [10], [11]. Although this is a valid approach for quantifying the closeness of a segment and a point, notice that it only relies on the proximity of the closest point of the road segment. Drawbacks of this approach have been discussed in detail recently in [12], where the authors have shown that it could yield questionable weights (probabilities), e.g. road segments of very different geometry and orientation getting the same

This research was supported by the National Research Foundation Singapore through the Singapore MIT Alliance for Research and Technology's FM IRG research program.

The authors would like to thank Paul Newson and John Krumm of Microsoft Research for making their vehicle and map data available for other researchers.

The authors would like to thank Kakali Basak, Liang Yu, and Joseph K. Lee of SMART for their helpful discussions about the QGIS software.

[1] In this paper, we use the generic term, 'GPS point', for each time-stamped vehicle position data collected from a satellite-based navigation device.

weights only as a result of having the same shortest distance to the GPS point. As an alternative, point-wise proximity-weights were defined for points defining a segment, and the cumulative value of these weights, defined through a line integral, yielded the proximity-weight of that segment. A closed form of the proposed weight was also developed for probabilistic methods under the assumption that road segments were straight, and that the GPS noise was Gaussian. In some of the recent studies, similar formulations that avoid the shortest-distance based proximity-weights have also been proposed e.g. [13], yet without a compact formulation. For this reason, the proposed cumulative proximity-weight of [12] could be an easily implementable, yet effective improvement to probabilistic methods in general.

However, [12] has stayed short of using the proposed weight as part of an actual MM algorithm. For this reason, at this point, a complete MM analysis is still needed to see whether the proposed weight would be expected to improve the overall accuracy of probabilistic methods or not. If so, one also needs to identify under what conditions that could be the case. In this study, we tried to answer these questions by developing a Hidden Markov Model (HMM) based MM algorithm with the proposed cumulative proximity-weight. To compare the MM results with the the proposed weight against results with a traditional shortest distance based weight, a parallel algorithm using the latter weight were also developed. HMM was chosen as the underlying methodology because of its simplicity in formulating the MM problem, and the reliable estimates it yields by finding the most likely sequence of road segments through the Viterbi algorithm. First used by Hummel [14], and later by Newson and Krumm [4], HMM has recently become essential in probabilistic MM studies.

Since proximity-weights are defined only through position data, in order to get a clear comparison of the two approaches, in this study we will assume the availability of only this type of data. This problem of identifying the true location of a vehicle by only using its position data, and the underlying road network is considered as the most basic MM problem. Other MM variants could be defined on this problem with the availability of extra data, such as velocity or heading direction data; and without a doubt their availability can improve matching results. Since we will be performing simulations on the most basic problem, improvements reflected in our simulations could be also expected to be seen with the availability of extra data.

For our simulations, we have used the same GPS and road network data set that was originally used by Newson and Krumm in [4]. The primary reason to use this data set was the availability of the ground truth data, which helped us identify the number of mismatching segments under our algorithms correctly. In addition, the usage of this data set also gave us an opportunity to safe check our shortest distance based algorithm by comparing our results with theirs. While we will start our simulations with comparing the performance of the two algorithms for the standard MM problem (GPS data sampled at each second), our primary focus will be about their performances under low-sampled and also noisy-data.

The rest of the paper is organized as follows. First, in II, we will briefly reintroduce the proximity-weights, both from the shortest distance and the cumulative weight perspectives. Later in III, we will go over the details of our HMM-based MM method, that will be used for both weight formulations. Finally, in IV, we will describe our simulations, and present our results. A comparison of the results and discussion will follow to evaluate the new proximity-weight in general.

II. Defining Proximity-Weights (Probabilities)

A. Segment Proximity-Weight based on Shortest Distance vs. Segment Proximity-Weight based on Line Integral

Let $d(p,s)$ define the distance between two points p and s; $d_m(p,S)$ the shortest distance between point p and a road segment $S \in \mathbb{S}$. Also, let $f : \mathbb{R}^+ \to \mathbb{R}^+$, be the generic weight function that defines the proximity-weight of segments in an MM method, and $\mathcal{W}_m(p,S)$ the proximity-weight of segment S with respect to p defined through the shortest distance measure. Then, in its general form, the shortest distance based proximity-weight can be formulated as,

$$\mathcal{W}_m(p,S) \doteq f(d_m(p,S)), \text{ where } \quad d_m(p,S) \doteq \min_{s\in S}(d(p,s)). \tag{1}$$

For defining a more precise proximity-weight of S, the proximity-weight of each point on S could be defined, and summed along S. Let $f(d(p,s))$ be the proximity-weight of a point s on S with respect to p. Then, the cumulative proximity-weight of road segment S with respect to p can be formulated with the line integral along S,

$$\mathcal{W}(p,S) \doteq \int_S f(d(p,s))\, dl. \tag{2}$$

Once the weights of all segments are calculated, normalization could be done to obtain relative weights,

$$w(p,S) \doteq \mathcal{W}(p,S) / \sum_{\mathcal{S}\in\mathbb{S}} \mathcal{W}(p,\mathcal{S}). \tag{3}$$

B. Proximity-Weights for Probabilistic MM Methods:

Considering the uncertain nature of the MM problem, in [12] (2) was progressed under a probabilistic approach, where a likelihood function of distance was used as the weight function, f. Let $pr(p\,|\,d(p,s))$ denote the likelihood of vehicle's true location being a distance of $d(p,s)$ away from point p. Then, following (2), the overall likelihood of vehicle traveling on segment S while being observed at p, $Pr(p\,|\,S)$, could be defined with an integral over S, which also yields the cumulative proximity-weight for probabilistic methods.

$$\mathcal{W}(p,S) = Pr(p\,|\,S) = \int_S pr(p\,|\,d(p,s))\, dl. \tag{4}$$

Since the year 2000, after the removal of Selective Availability, GPS data has shown more clear pattern of a Gaussian distribution [6]. Consequently, a Gaussian distribution centered around the GPS point with zero mean was chosen for $pr(p\,|\,d(p,s))$, similar to works [4], [5], [7], [15], [16], [10], [11].

$$pr(p\,|\,d(p,s)) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(d(p,s))^2/2\sigma^2} \tag{5}$$

where σ is the assumed standard deviation of the GPS measurements. By substituting (5) into (4), and parametrizing the road segment S, a closed form was obtained as,

$$\mathcal{W}(p,S) = e^{(b^2/4a-c)/2\sigma^2}\left[\Phi\left(\frac{2a+b}{2\sigma\sqrt{a}}\right) - \Phi\left(\frac{b}{2\sigma\sqrt{a}}\right)\right] \tag{6}$$

where Φ is the standard cumulative distribution function for the Gaussian distribution, and the constants are,

$$\begin{aligned} a &= (x_B - x_A)^2 + (y_B - y_A)^2, \\ b &= 2\left[(x_A - x_1)(x_B - x_A) + (y_A - y_1)(y_B - y_A)\right], \\ c &= (x_A - x_1)^2 + (y_A - y_1)^2, \end{aligned} \tag{7}$$

where (x_A, y_A) and (x_B, y_B) are the Cartesian coordinates of nodes defining the straight road segment S. On the other hand, if one had used the shortest distance based proximity-weight function of (1), along with the same Gaussian likelihood function of (5), the following weight would have been obtained.

$$\mathcal{W}_m(p,S) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(d_m(p,S))^2/2\sigma^2}. \tag{8}$$

As said earlier, in probabilistic MM literature these proximity-weights are commonly known as the observation probability, and we will interchangeably use both terms.

III. HMM- Based Map Matching Algorithm

The MM problem can be approached as a discrete-time state estimation problem, where the state of the system at some time k, X_k, is the road segment that the vehicle traverses at k, and the measurement at k, Y_k, is the position of the vehicle. When the dynamics of the vehicle (speed, acceleration, heading etc.) is not known, approaching this estimation problem from a probabilistic viewpoint would be reasonable. Considering the fact that the states are not observable, and also the fact that GPS measurements are independent, this problem could be solved under an HMM framework. An HMM is a bivariate discrete time process $\{X_k, Y_k\}_{k\geq 0}$, where $\{X_k\}$ is a Markov chain, and conditional on $\{X_k\}$, $\{Y_k\}$ is a sequence of independent random variables such that the conditional distribution of Y_k only depends on X_k [17]. It is characterized by its 5-tuple, state space, observation space, state transition probability, observation probability, and the initial probability.

For the MM problem, HMM state space would consist of the road segments of the road network, and the observation space would be the continuous space defined by latitude and longitude coordinates on the globe. Since we do not have any additional information about the state of the system initially, the initial probabilities can be considered as uniformly distributed. In reality, the observation probability conditioned on the state, $Pr(Y_k|X_k)$, would depend on several factors, such as road segment's latitude/longitude/elevation, and satellite health. Like in other probabilistic MM methods, in this method, we also consider a simplified model where observation probability will be defined only by the proximity of the GPS point to the segment. In fact, following this simplifying assumption, the probabilistic proximity-weights of (6) and (8) had become to be known under the term observation probability. Markovian transition probabilities can be defined by considering both network topology and the GPS measurements. Details of observation and transition probability formulations are discussed in the next part.

A. Candidate Segments/Links and Observation Probabilities:

We start our analysis by identifying the candidate segments of given GPS points, that is identifying the road segments that are possibly the original segment on which the vehicle was traveling. For practicality, we have used a circular error region rather than an ellipse. For deciding the size of the error region, there is no consensus among MM methods. Most methods use a large enough circular radius, e.g 100m. in [10], and 200 m. in [4], to make sure that none of the true segments are omitted. However, this might also turn the candidate identification into a computational burden. Since the road segments remaining out of this region are excluded as unlikely segments, from a probabilistic perspective, one only needs to make sure that the probability of an excluded segment being a true segment should almost be equal to zero. For this reason, the size of the error region can be decided by a function of the standard deviation of GPS measurement noise, σ_o, rather than using a preset value. When a bivariate, uncorrelated Gaussian distribution is assumed for the GPS data, by having a circular error region of radius, $R_0 = 5 * \sigma_o$, one would expect having only 1 out of 100,000 GPS measurements to be outside this region. Since this number is just the expectation, we've doubled this factor, and fixed the radius of our error circle as, $R_0 = 10 * \sigma_0$, in our algorithms. Overall for all simulations including noise added GPS data, it was sufficient to catch all true candidate segments.

Identifying the candidate road segments of a given set of GPS points is a range query problem. Thus, data structures suitable for this query, e.g. cell structure, KD-tree, or quad-tree have commonly been used in MM algorithms. Notice that in (7) (x_A, y_A) and (x_B, y_B) are defined in Cartesian coordinates. For this reason, in this study we have transformed the given data in latitude/longitude coordinates to rectangular coordinates in UTM coordinate system. Consequently, we have chosen to do range querry with the cell structure which is also based on a rectangular grid. For the cell size, our initial choice was $2 * R_0$. However, when identifying candidate segments, this common approach of identifying them only through identifying the shape points falling inside the error region would be insufficient. This approach would miss a candidate segment whose nodes remain outside the error region, but which still passes through the region. In order not to miss these, we needed to increase the cell size to a larger value, so that all necessary shape points could be identified. At the same time, given some GPS points, we can not know the length of candidate segments before finding them; yet to detect these segments completely we would need to know their length. For this reason, the maximum segment length on the network, L_M, was used for defining the size of the cell. Yet, this results in a huge number of candidate segments, most of which might not

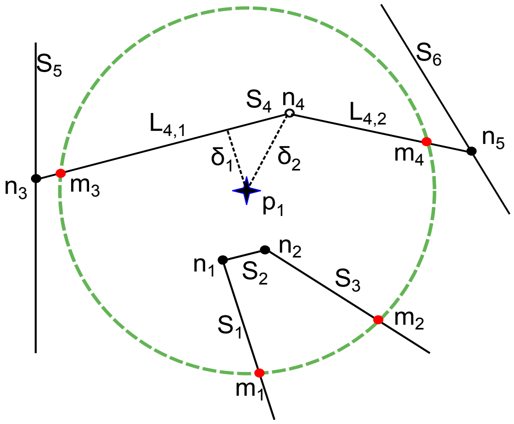


Fig. 1. Circular error region defined around GPS point, p_1, to identify its candidate road segments S_1, S_2, S_3, and S_4. S_4 is a polyline road consisting of two straight links $L_{4,1}$ and $L_{4,2}$. Black dots, $\{n_i\}$, represent the nodes/shape points of a network, red points, $\{m_i\}$, represent the intersection points of the candidate segments with the error region.

ever pass through the error circle. Consequently, a secondary, finer, cell structure was also used, with a size of 100 meters. The range query was implemented as a combination of these two cell structures, and considering the lengths of segments in the network.

In [12], (6) was developed under the assumption that the road segments were straight lines. In spatial vector maps this assumption may not hold. Yet again, in these maps road segments are represented by polylines which are a union of straight links defined by their shape points, e.g. segment S_4 is a union of $L_{4,1}$ and $L_{4,2}$ in Fig. 1. Consequently, when working with any vector map, (6) would be still valid for these straight links, and could be used to find the weights of links making up a segment. Since these links are disjoint (except intersection points), the proximity-weight of a segment could be found by the sum of the weights of these links. As an alternative, each link could be analyzed separately, and the whole MM analysis could be based on links rather than segments.

Notice that the need for analyzing the links of a segment is also needed in a shortest distance based weight formulation as well. In order to find the shortest distance from a particular segment to a point, the distance from all the links of that segment to the point would need to be found, e.g distances δ_1 and δ_2 need to be found for S_4 in Fig. 1. For this reason the computational complexity of calculating both weights remain similar. The only difference is that, when finding the cumulative-proximity-weights, only the portion of links remaining inside the error region should be taken into account. For instance, in Fig. 1, coordinates of n_1 and n_2 will be used to find weights of S_2, and coordinates of m_3 and n_4 will be used for $L_{4,1}$. Finding the intersection points of road links with error regions will be the only extra calculation that would be required by the new weight, but this effort is minimized by using simple algebraic line-circle intersection formulas.

B. Transition Probabilities and The Solution of HMM

As mentioned earlier, for the MM problem, the state space of HMM consists of the segments in the road network. When formulating the transition probabilities between two states Hummel [14] had only considered the connectivity of segments. Later, Newson and Krumm [4] had considered the shortest distance (SD) between the segments and compared it to the great circle distance (GCD) between the two GPS points, corresponding to those segments. The difference between these two distances is used in an exponential distribution that resulted in the transition probability. In our work, we also favor using a formulation similar to [4]. However, we will not keep the exponential scale parameter, β, a constant, but will make it a variable depending on the period between GPS points, Δ.

$$Pr(X_k|X_{k-1}) \doteq \frac{1}{\beta_\Delta} e^{-d_{(k,k-1)}/\beta_\Delta} \tag{9}$$

where,

$$d_{(k,k-1)} \doteq |\|Y_k - Y_{k-1}\|_{GCD} - \|X_k - X_{k-1}\|_{SD}| \tag{10}$$

This change is introduced particularly to improve MM results of low sampled GPS data. When the period between GPS points is low, the difference between $\|Y_k - Y_{k-1}\|_{GCD}$ and $\|X_k - X_{k-1}\|_{SD}$ would be much smaller, as compared to the difference when that period is large. Thus, using the same β value for larger Δ values could lead to having some transition probabilities become almost zero, and in our simulations we have noticed subsequent Markov chain breakup as a result of this. This chain breakup with constant β was also observed by Song et. al. in [18], and the authors had calculated β estimates for each different sampling period to avoid that. In this study, we have defined β_Δ with an affine formulation,

$$\beta_\Delta = \beta_0 + \Delta/10 \tag{11}$$

where β_0 is the base β value. Under this formulation when periods are large, a big difference between GCD and SD values will not be assigned a very small probability, as compared to when delta is small. For solving the HMM problem, we will use the well known Viterbi algorithm, which finds the most likely sequence of states.

IV. Simulations and Results:

A. Preprocessing

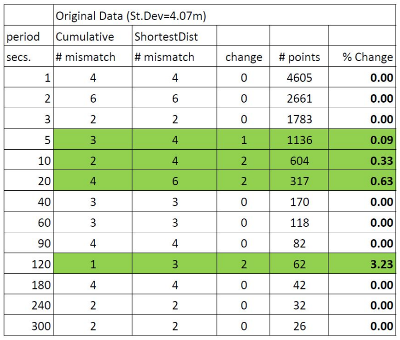
The original GPS and network node data were given in latitude and longitude coordinates. The formulation of (6), on the other hand, was developed under Cartesian coordinates. To have coordinate compatibility, in the preprocessing step we have converted all GPS and node data to the UTM system, a metric-based Cartesian grid coordinate system. This conversion also let us avoid the Great Circle, or the Vincenty formulas needed for calculating the distance between two points on globe. Instead we will be able to use practical Euclidean distance formulations. Another advantage will be for the range query of nodes around GPS points with the cell data structure, which will be very easily implemented under the UTM system.

Since we also planned to make a comparison of our results with [4], we followed their preprocessing of GPS points, that is we removed points that were within $2*\sigma_o$ distance away from their predecessors, to make sure that the vehicle has moved. We have also identified only 2 GPS outliers that were observed more than 50 meters away from their original segments, and removed them. They were right next to a tunnel, where the GPS signals must be weak. The value of σ_o was found to be 4.07 meters using the median absolute deviation estimator. Following this, we were also left with 4605 points out of the original 7531. During our early simulations we also have noticed several break points in the road network where two nodes defined at the same point were not connected to each other, and manually connected those points. After patching the right connections to the network, and also using β as a variable of sampling period (11), in our simulations we did not observe any HMM breaks, as opposed to 100 of them being reported in [4].

B. Standard Map Matching Comparison

After preprocessing, we ran both of our algorithms, with the cumulative, and the shortest distance based observation probabilities. In our simulations we have set the base value, β_0, to 1. For the standard problem, that is with 1 second sampling period, both methods only mismatched 4 GPS points out of 4605. In [4], the authors have reported no mismatch. However, their result does not include 100 excluded points related to chain break up. Considering this omitted points, we believe our results were as competitive as theirs.

We have repeated our simulations for a total of 13 different sampling periods, ranging from 1 second to 300 seconds. Fig. 2 lists our findings where the number of mismatched GPS points are listed in the first two columns for cumulative and shortest distance based probabilities, respectively From these simulations it was observed that the cumulative probability had a slight edge to the traditional one for sampling periods of $5, 10, 20$ and 120 seconds; and for the other sampling periods the results were same.



	Original Data (St.Dev=4.07m)				
period	Cumulative	ShortestDist			
secs.	# mismatch	# mismatch	change	# points	% Change
1	4	4	0	4605	**0.00**
2	6	6	0	2661	**0.00**
3	2	2	0	1783	**0.00**
5	3	4	1	1136	**0.09**
10	2	4	2	604	**0.33**
20	4	6	2	317	**0.63**
40	3	3	0	170	**0.00**
60	3	3	0	118	**0.00**
90	4	4	0	82	**0.00**
120	1	3	2	62	**3.23**
180	4	4	0	42	**0.00**
240	2	2	0	32	**0.00**
300	2	2	0	26	**0.00**

Fig. 2. Comparison of MM results, with original data, for different periods

C. Sparse and Noisy Map Matching Comparison:

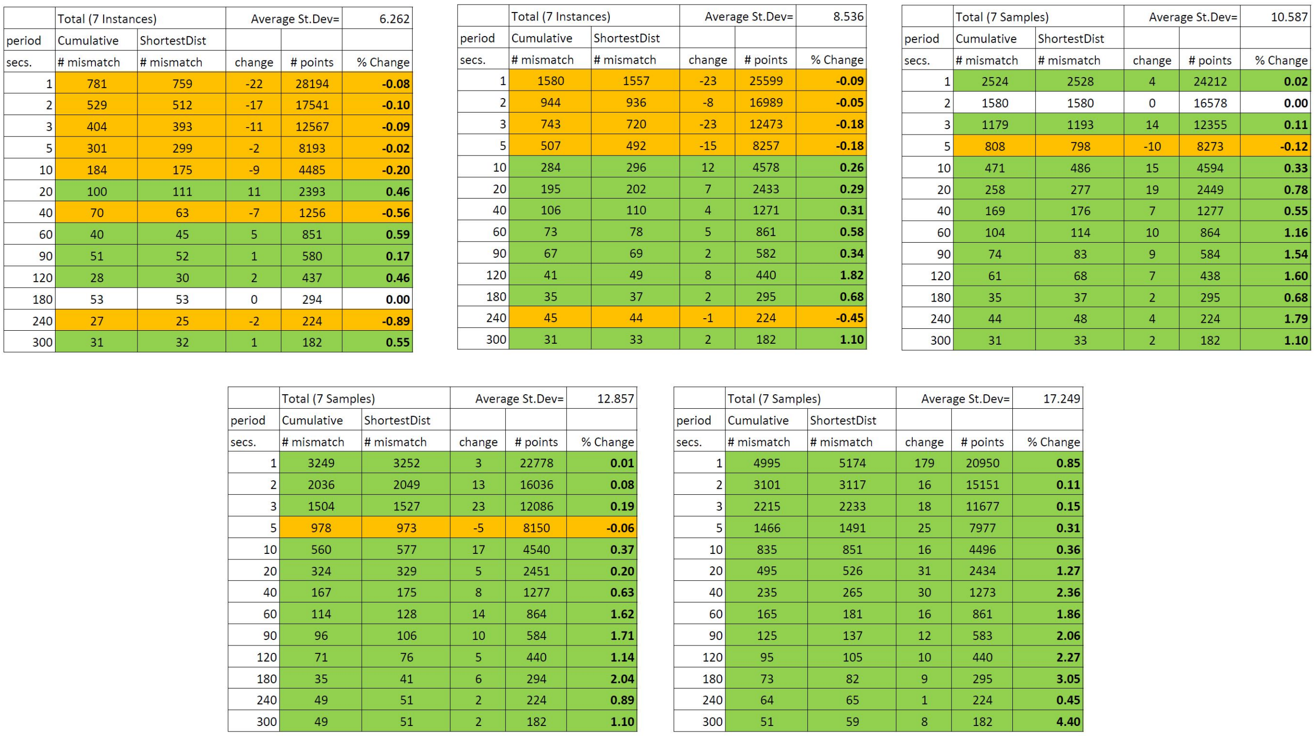
Recall that the original GPS data had a standard deviation of 4.07 meters. This value is below the expected standard deviation values defined in the GPS guideline [19]. For this reason, there is a possibility that the previous simulations were done in the availability of highly accurate GPS measurements, and thus we were interested in seeing our algorithms' expected performances when GPS data might not be so accurate. For this purpose, we intentionally corrupted the original GPS data. In order to make sure that the corrupted data will also be Gaussian, we will be adding again Gaussian random variables with zero mean and a chosen σ_x standard deviation value to the original data. Since the original GPS data, and the added noise are independent, the final corrupted GPS data will be Gaussian with zero mean and standard deviation, σ_f, where $\sigma_f^2 = \sigma_o^2 + \sigma_x^2$. We wanted to analyze, the MM performance of both algorithms when the noise levels exceeded their original value about 50%, 100%, 150%, 200%, 300%, which would result about $\sigma_f = 6$, $\sigma_f = 8$, $\sigma_f = 10$, $\sigma_f = 12$, $\sigma_f = 16$ meters, respectively. The corrupted data was formed according to these σ_f values. One thing to note is that the corrupted data is itself a random variable, and the GPS values will be different at each sampling of the added noise. For this reason, for each σ_f, we formed 7 different corrupted GPS data, and ran the simulations for all of them. Each of these simulations were also run for 13 different sampling periods, same as the simulation in the previous part.

Fig. 3 summarizes our findings, where each subfigure shows the total MM results of 7 simulations for each set of corrupted data. In these simulations, the actual standard deviation of each corrupted GPS data set was found again by median absolute deviation estimator. The average standard deviation of each data group is given on the top right corner. Orange color is used to highlight results where MM accuracy has dropped with cumulative probability, green color is used to highlight results where MM accuracy has improved, and white color results are the ones that MM accuracy stayed the same.

From these simulations we can see that the proposed probability has not always yielded better results. Especially when noise levels are low, such as the case where σ_f is around 6 meters, and sampling frequency is also high, its results remain slightly inferior to the shortest distance based weight. As sampling period increases, it starts yielding better results but the gain is not much when GPS data noise is not high. On the other hand, we notice that cumulative probability starts performing better with increasing noise levels. Improvements also become more substantial with $2\% - 3\%$ accuracy gains possible.

D. Discussion

Following our simulations to gain more insight, we analyzed individual mismatches. We have noticed that the loss of accuracy for the less noisy GPS data stems from the fact that the proposed probability, being defined through the line integral, favors long and nearby segments to a GPS point more than the shortest distance based probability. For instance, when noise levels, σ_f, was very low, like in the original data, both weights did well, since the true segments were most of the time the nearest segments to the GPS points. Around mid-noise



	Total (7 Instances)		Average St.Dev=		6.262
period	Cumulative	ShortestDist			
secs.	# mismatch	# mismatch	change	# points	% Change
1	781	759	-22	28194	**-0.08**
2	529	512	-17	17541	**-0.10**
3	404	393	-11	12567	**-0.09**
5	301	299	-2	8193	**-0.02**
10	184	175	-9	4485	**-0.20**
20	100	111	11	2393	**0.46**
40	70	63	-7	1256	**-0.56**
60	40	45	5	851	**0.59**
90	51	52	1	580	**0.17**
120	28	30	2	437	**0.46**
180	53	53	0	294	**0.00**
240	27	25	-2	224	**-0.89**
300	31	32	1	182	**0.55**

	Total (7 Instances)		Average St.Dev=		8.536
period	Cumulative	ShortestDist			
secs.	# mismatch	# mismatch	change	# points	% Change
1	1580	1557	-23	25599	**-0.09**
2	944	936	-8	16989	**-0.05**
3	743	720	-23	12473	**-0.18**
5	507	492	-15	8257	**-0.18**
10	284	296	12	4578	**0.26**
20	195	202	7	2433	**0.29**
40	106	110	4	1271	**0.31**
60	73	78	5	861	**0.58**
90	67	69	2	582	**0.34**
120	41	49	8	440	**1.82**
180	35	37	2	295	**0.68**
240	45	44	-1	224	**-0.45**
300	31	33	2	182	**1.10**

	Total (7 Samples)		Average St.Dev=		10.587
period	Cumulative	ShortestDist			
secs.	# mismatch	# mismatch	change	# points	% Change
1	2524	2528	4	24212	**0.02**
2	1580	1580	0	16578	**0.00**
3	1179	1193	14	12355	**0.11**
5	808	798	-10	8273	**-0.12**
10	471	486	15	4594	**0.33**
20	258	277	19	2449	**0.78**
40	169	176	7	1277	**0.55**
60	104	114	10	864	**1.16**
90	74	83	9	584	**1.54**
120	61	68	7	438	**1.60**
180	35	37	2	295	**0.68**
240	44	48	4	224	**1.79**
300	31	33	2	182	**1.10**

	Total (7 Samples)		Average St.Dev=		12.857
period	Cumulative	ShortestDist			
secs.	# mismatch	# mismatch	change	# points	% Change
1	3249	3252	3	22778	**0.01**
2	2036	2049	13	16036	**0.08**
3	1504	1527	23	12086	**0.19**
5	978	973	-5	8150	**-0.06**
10	560	577	17	4540	**0.37**
20	324	329	5	2451	**0.20**
40	167	175	8	1277	**0.63**
60	114	128	14	864	**1.62**
90	96	106	10	584	**1.71**
120	71	76	5	440	**1.14**
180	35	41	6	294	**2.04**
240	49	51	2	224	**0.89**
300	49	51	2	182	**1.10**

	Total (7 Samples)		Average St.Dev=		17.249
period	Cumulative	ShortestDist			
secs.	# mismatch	# mismatch	change	# points	% Change
1	4995	5174	179	20950	**0.85**
2	3101	3117	16	15151	**0.11**
3	2215	2233	18	11677	**0.15**
5	1466	1491	25	7977	**0.31**
10	835	851	16	4496	**0.36**
20	495	526	31	2434	**1.27**
40	235	265	30	1273	**2.36**
60	165	181	16	861	**1.86**
90	125	137	12	583	**2.06**
120	95	105	10	440	**2.27**
180	73	82	9	295	**3.05**
240	64	65	1	224	**0.45**
300	51	59	8	182	**4.40**

Fig. 3. Comparison of results, for different set of noisy data, for different periods

levels, when σ_f is around 6 or 8 meters, the true segments were not necessarily the nearest segments to the GPS points. However the new formulation still favored nearest segments, particularly if they were long. This was the reason for the slight loss in accuracy. This situation diminished when noise levels, σ_f, further increased, and (6) distributed more weights to farther away segments, something the shortest distance weight couldn't do.

V. Conclusion

In this study we have developed a hidden markov model based map matching method, which used the cumulative proximity weights of [12]. These weights were proposed to improve some shortfalls of the classical shortest distance based weights, but have never been used in a real MM method; and thus their possible improvements to MM methods were yet to be seen. In order to asses possible accuracy improvements, simulations were done using the GPS data set of [4] from Seattle area. Particular emphasis was given to improving the MM accuracy of low-frequency sampled, and noisy GPS data. The simulations resulted in interesting results that showed the proposed weights may not be always be superior to the traditional weight. However, the proposed weights were indeed able to improve MM accuracy when GPS sampling periods were long and/or the noise level in GPS data was also large. From our simulations, we can conclude that, the cumulative proximity-weight could be a better choice than the shortest distance based weight when MM will be done with GPS data that is highly noisy or when the sampling periods are long.

References

[1] D. Pfoser and C. S. Jensen, "Capturing the uncertainty of moving-object representations," in *SSD '99: Proceedings of the 6th International Symposium on Advances in Spatial Databases*, pp. 111–132, 1999.

[2] M. Quddus, W. Ochieng, L. Zhao, and R. Noland, "A general map matching algorithm for transport telematics applications," *GPS Solutions*, vol. 7, pp. 157–167, 2003.

[3] N. R. Velaga, M. A. Quddus, and A. L. Bristow, "Developing an enhanced weight-based topological map-matching algorithm for intelligent transport systems," *Transportation Research Part C: Emerging Technologies*, vol. 17, no. 6, pp. 672 – 683, 2009.

[4] P. Newson and J. Krumm, "Hidden markov map matching through noise and sparseness," in *17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 336–343, 2009.

[5] Y. Lou, C. Zhang, Y. Zheng, X. Xie, W. Wang, and Y. Huang, "Map-matching for low-sampling-rate gps trajectories," in *17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 352–361, ACM, 2009.

[6] F. Diggelen, "System design & test-gnss accuracy-lies, damn lies, and statistics," *GPS World*, vol. 18, no. 1, pp. 26–33, 2007.

[7] S. Brakatsoulas, D. Pfoser, R. Salas, and C. Wenk, "On map-matching vehicle tracking data," in *31st international conference on Very large data*, VLDB '05, pp. 853–864, VLDB Endowment, 2005.

[8] W. Ochieng, M. Quddus, and R. Noland, "Map-matching in complex urban road networks," *Brazilian Journal of Cartography (Revista Brasileira de Cartografia)*, vol. 55, no. 2, pp. 1–18, 2003.

[9] F. Marchal, J. Hackney, and K. Axhausen, "Efficient map matching of large global positioning system data sets: Tests on speed-monitoring experiment in zürich," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1935, no. -1, pp. 93–100, 2005.

[10] J. Yuan, Y. Zheng, C. Zhang, X. Xie, and G.-Z. Sun, "An interactive-voting based map matching algorithm," in *11th International Conference on Mobile Data Management*, pp. 43–52, IEEE Computer Society, 2010.

[11] C. Goh, J. Dauwels, N. Mitrovic, M. T. Asif, A. Oran, and P. Jaillet, "Online map-matching based on hidden markov model for real-time traffic sensing applications," in *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, pp. 776 –781, sept. 2012.

[12] A. Oran and P. Jaillet, “A precise proximity-weight formulation for map matching algorithms,” in *Positioning Navigation and Communication (WPNC), 2013 10th Workshop on*, pp. 1–6, 2013.

[13] M. Bierlaire, J. Chen, and J. Newman, “A probabilistic map matching method for smartphone gps data,” *Transportation Research Part C: Emerging Technologies*, vol. 26, no. 0, pp. 78 – 98, 2013.

[14] B. Hummel, “Map matching for vehicle guidance,” in *Dynamic and Mobile GIS: Investigating Changes in Space and Time*, Innovations in GIS, Taylor & Francis, 2006.

[15] M. E. E. Najjar and P. Bonnifait, “A road-matching method for precise vehicle localization using belief theory and kalman filtering,” *Autonomous Robots*, vol. 19, pp. 173–191, 2005.

[16] C. Smaili, M. E. El Najjar, and F. Charpillet, “A road matching method for precise vehicle localization using hybrid bayesian network,” *Journal of Intelligent Transportation Systems*, vol. 12, no. 4, pp. 176–188, 2008.

[17] O. Cappé, E. Moulines, and T. Rydén, *Inference in hidden Markov models*. Springer, 2005.

[18] R. Song, W. Lu, W. Sun, Y. Huang, and C. Chen, “Quick map matching using multi-core cpus,” in *20th International Conference on Advances in Geographic Information Systems*, pp. 605–608, ACM, 2012.

[19] *NAVSTAR Global Positioning System Surveying*, ch. 4, pp. 4–1 – 4–18. No. 1110-1-1003, US Army Corps of Engineers, February 2011.