A Map Matching Algorithm for Car Navigation Systems with GPS Input

Konstantoulas Ioannis, ikonst@math.uoc.gr
Prastacos Poulicos, poulicos@iacm.forth.gr

Institute for Applied and Computational Mathematics, FORTH and University of Crete

INTRODUCTION

This paper describes a new map matching algorithm suitable for online car navigation systems using a GPS receiver for location estimation. A brief report is also presented of the results of an experimental implementation and field test in Heraklion City in Crete. The algorithm was designed with the scope in mind of striking a balance between the simplicity required of an easily implementable, practical algorithm and the sophistication required of a strong algorithm with good error handling in problematic situations. In particular, the algorithm combines a straightforward implementation and usability in complex or dense road networks.

Although artificial error sources like SA have been removed from civilian GPS devices, a number of naturally occurring error sources remain that undermine the quality of GPS measurements to an extent that for practical car navigation systems, techniques like map matching are still required to identify the road a vehicle is moving on with a high degree of confidence. Moreover, not all GPS devices exploit the full range of the capabilities of the GPS system in order to be more cost-effective; the type of information received and processed by the receiver varies with manufacturer and model.

These considerations drove the authors to look for an algorithm that does not require anything but the minimal input offered by GPS devices. The main requirement of this algorithm is a GPS source that provides location information at a relatively high frequency, about one measurement per three to seven seconds. A relatively error-free map is also highly desirable; here error-free means that it does not miss roads and the piecewise-linear approximations to curved roads have small deviation in curvature. A goal of the design has been to keep the hardware resources required for the implementation to a minimum, so that an average PDA can run the algorithm amidst several other operations.

DESCRIPTION OF THE ALGORITHM

The algorithm is an iterative scheme working with periodic inputs in a time interval of less than seven seconds (assuming a car is traveling with an average speed of less than 70 km/h, a reasonable requirement in a dense network). For a general description of templates for map matching algorithms see the standard reference (Zhao,1997). Initially, the algorithm uses a crude estimation from the GPS to isolate a tight area outside which the vehicle has zero probability of being located. An initial probability distribution is then assigned on the set of roads within that area (termed a local neighborhood of roads) that describes for each road how likely it is for the vehicle to lie on that particular road. An appropriate function of the inverse of the distance between the road segments and the GPS estimation is used for this purpose.

In subsequent measurement cycles the algorithm uses the new measurement to isolate a next neighborhood and create a transition probability distribution between the previous and the current neighborhoods. This new probability assignment describes for pairs of roads the likelihood the vehicle has transited from the first road to the second. This is conceptually the hardest part of the algorithm (and its core). Currently this transition distribution is created exploiting local road geometry and optionally speed information provided by the GPS device; the formulae used for the calculation are the main source of the requirement of high frequency of measurements. However, the authors are currently studying options concerning possible refinements of the method to relax the high frequency of measurements requirement.

Below lies a sketch of how the algorithm creates the transition probabilities. To facilitate the presentation, some useful mathematical notations are introduced. Since the roads are represented by linear segments on a Euclidean frame, for each point and road segment there corresponds a projection on the segment. The projected point is denoted by $pr(x,S)$. If the line through the point *x* orthogonal to *S* does not meet the road segment, $pr(x,S)$ is taken to be the endpoint on *S* nearest to *x*. The road database is denoted by *R* and the road segments by *S*. The elements *S* contain all the information provided by the road except curvature. For points on road segments a distance function $d(a,b)$ is defined as the length of the shortest path from a to b. Thus the *r*-isoroutes are the sets $\{S \in R \mid d(a,b_S) < r\}$, $b_S \in S$ being the farthest (in the Euclidean sense) point to a. When a or b do not belong on a road, $d(a,b) = d(a',b')$, the last variables denoting the projections of the first to the nearest roads. If *M* is the maximum error margin for the GPS (in this work usually taken to be approximately 100m), the *M*-isoroutes are the abovementioned local road neighborhoods, or LR for short. Let TS be the set of all road segments that touch *S*. Given two consecutive measurements, the transition probability $p_{(t-1,t)}(S \mid S')$ for $S' \in TS \cap LR_t$ must depend only on the vectors $dm = \overline{m}_i - pr(\overline{m}_i)$ and $dm_j = \overline{m}_i' - pr(\overline{m}_i)$. This means that the transition probability depends on how close, in an appropriate sense, the vectors dm and dm_j lie, the closer they are the greater the probability. This is justified as follows: the error corrupting m_i is not as great in order as the error introduced by attaching the final measurement on a wrong road (always assuming the initial road is the correct one). Furthermore, this exactly is the method empirically used to estimate the route of the vehicle (when there is no future information).

Now the method functions as follows: for each $S \in LR_{t-1}$ and $S_j \in LR_t$ set $m_j^1 = m^1(S_j) = w_1(l_u)$, with *u* being the angle between dm and dm_j, l_u the arc length of the arc dm_j traces until it coincides with dm and w_1 a strictly decreasing smoothening function. Denote the coincidence point of dm and the rotated dm_j by $r(dm_j)$. Then a second estimation is given by $m_j^2 = m^2(S_j) = w_2(\left\| pr_j(\overline{m}_i) - r(dm_j) \right\|)$ with w_2 another strictly decreasing smoothening function.

Afterwards the non normalized estimation of the transition likelihood is defined as $l_j = l_j(m_j^1, m_j^2) = Q(S,S_j)(m_j^1 + m_j^2)$. The *Q* function plays the following important role: it examines whether the vehicle can end up on *S* moving from S_j in the time interval between the two measurements with the speed estimated by the GPS. Without this function the algorithm would fail in many cases, especially in long parallel roads connected by a small road segment orthogonal to both. The function is also responsible for checking non topological information. The database may give further information for roads like pedestrian walkway or one-way status. In general, the purely geometric term of isoroute is not a satisfactory measure for accessibility from a point to a road. Such characteristics are taken into consideration and play an important role in the function of the algorithm. The function Q returns 0 if some of the checks it performs fails, otherwise it returns 1.

Finally we normalize setting

$$p_{(t-1,t)}(S \mid S') = l_j \left(\sum_{S_k \in LR_{t-1}} l_k \right)^{-1}$$

To complete the cycle the transition probability distribution acts on the prior distribution by matrix-to-vector multiplication to produce the next probability distribution on the new local neighborhood of roads.

The algorithm, up to this point, does not single out a road as the correct road on which the vehicle is moving. This is a process better suited to a separate decision making module. This does not mean that there are many ways, after the distribution assignment, to choose the correct road. The natural choice (and the one used in the test implementation) is to choose the road with the highest probability. However, by separating the procedure of assigning probabilities from that of selection, greater flexibility is allowed concerning how the distributions are handled and interpreted. This enables one to be able to answer questions other than that of the exact location of the vehicle.

IMPLEMENTATION AND CONCLUSIONS

The present work focused on a map matching algorithm for use in online car navigation systems with limited processing power and real-time demands that is easy to implement and does not require much information from the GPS besides the essential.

Using this data, the algorithm is called to estimate, in an appropriate sense, the likelihood, at each measurement of the GPS and for each road in the network, the vehicle is on that particular road at the time instant the measurement was taken. In this work, this appropriate sense is the following: a probability assignment is attached to each road “near” the GPS measurement. The totality of these assignments forms a probability distribution that achieves the required estimation.

There are many reasons why the system does not simply output the road with the highest probability. First and foremost, the system could be wrong. Therefore, since infallibility cannot be guaranteed, it is senseless to output a single road estimate. This is especially the case when missing the correct road can potentially result to system failure, such as in an online route planning and guidance application. Another reason is that an external module might be needed to further process the probability distribution and use it as a guideline for other tasks (fuzzy-set based travel guides and traffic control servers are notable examples). Finally, the technical reason for the format of the output is that we use it recursively as feedback to the system so we need to compute it anyway; it is not an additional, non-operational process.

All the numerical formulae used in this work are very easily implemented. This was one of the design goals, since many map matching systems are theoretically optimal but their implementation is very hard or demanding on the device capabilities; for example they often demand many statistical data on the GPS signal to be known or they may demand computations including many time-consuming matrix operations where round-off errors may be difficult to control.

However, care is required for the present work as well. Since the scheme presented here is iterative of depth one, it is theoretically possible to accumulate round-off errors that render the procedure useless after a sufficient number of steps.

In order to test the efficiency of the algorithm a test implementation was programmed. Data was collected using ambulances that roamed the city of Heraklion in Crete for a period of several days. The road network of Heraklion City is well known for its density and irregularity, so it provided a good measure against which the algorithm was to be tested. Using the data, an online environment was then emulated in which the implementation had to correctly identify roads where the vehicle was moving on using limited memory and CPU resources. The implementation followed closely the above description of the algorithm and all design choices (like the smoothening functions used in the crucial step of determining the transition probability distribution) were set to be as simple as possible. In order to cover the rare event that the algorithm becomes destabilized, a simple module was programmed that detected instability by examining the probability distribution assignment and then deciding whether it met some empirically defined criteria for instability or not. Despite initial concerns about instability, there was no place where the algorithm was persistently destabilized and the success rate for the correct road estimation was almost 100% in each trial route. Even after introducing a further (artificial) error with Gauss distribution the success rate remained over 96%. This showed that the algorithm can cope with the empirical tests and has been proven resistant to natural and artificial shocks in most of the cases encountered.

BIBLIOGRAPHY

Zhao, Y.,1997, Vehicle Location and Navigation Systems, London, Artech House Publishers.