Interpolating and Using Most Likely Trajectories in Moving-Objects Databases

Byunggu Yu[1] and Seon Ho Kim[2]

[1]Computer Science Department, University of Wyoming,
Laramie, WY 82071, USA
yu@uwyo.edu
[2]Computer Science Department, University of Denver,
Denver, CO 80208, USA
seonkim@cs.du.edu

Abstract. In recent years, many emerging database applications deal with large sets of continuously moving data objects. Since no computer system can commit continuously occurring infinitesimal changes to the database, related data management techniques view a moving object's trajectory as a sequence of discretely reported spatiotemporal points. For each pair of consecutive committed trajectory points, a spatiotemporal uncertainty region representing all possible in-between trajectory points is defined. To support trajectory queries with a non-uniform probability distribution model, the query system needs to compute (interpolate) the "most likely" trajectories in the uncertainty regions to determine the peak points of the probability distributions. This paper proposes a generalized trajectory interpolation model using parametric trajectory representations. In addition, the paper expands and investigates three practical specializations of our proposed model using a moving object with momentum, i.e., a vehicle, as the exemplar.

1 Introduction

With advances in GPS (Global Positioning System), RFID (Radio Frequency Identification) technology, and sensor technology, emerging database applications begin to deal with large sets of objects each of which can continuously move in a geographic space and frequently report its current spatiotemporal attribute values, such as position, velocity, and acceleration, to the database server. These applications include mobile communication systems, location-based services, digital battlefields, transportations, air- or ground-traffic control systems, and sensor networks, to name a few. In the future, more complex and larger applications that deal with higher dimensional attributes (e.g., moving sensors capturing multiple stimuli) will become commonplace – increasingly complex sensor devices will proliferate alongside potential applications associated therewith.

To support such *moving-objects database* (*MOD*) applications, one requires an on-line database server that can store, update, and retrieve large sets of moving objects. Each moving object has both spatiotemporal properties representing the trajectory and non-spatiotemporal properties such as identification, associated phone number, and owner's name. Conventional database technology can efficiently manage the non-spatiotemporal properties of moving objects and efficiently process queries referring to only non-spatiotemporal properties of moving objects. To support the new applications, it is important to design and implement a MOD server that can also efficiently process queries referring to the spatiotemporal trajectories.

A MOD server must be able to keep track of the trajectories of individual moving objects to process queries referring to the trajectories. Existing techniques view a trajectory as a sequence of connected segments in a 2-, 3-, or 4-dimensional space. This is due to the fact that, although objects can continuously move or change, database management systems cannot deal with continuously occurring infinitesimal changes – this would effectively require infinite computational speed and sensor resolution. Thus, each object's combined attribute values (states) spanning multiple dimensions – location as the lowest order derivative in the spatiotemporal context, and velocity and acceleration as higher order derivatives – can only be discretely updated. In turn, each segment is associated with a certain degree of uncertainty that encloses all possible unknown locations of the object for that segment.

This paper presents our study of trajectory representation models, specifically most likely trajectory representation of moving objects with momentum. Representing the trajectory of a moving object more accurately with a fewer number of reported points is an important issue in designing MOD servers because the frequency of trajectory updates is a significant factor in determining the performance of a real-time MOD server.

This paper also proposes a solution framework for the following issue: Since queries referring to moving object trajectories are processed over the uncertainty regions, each resulting object should be associated with the probability (or likelihood) that the trajectory really satisfies the query predicate with respect to the reference object. To support this, the probability distribution of all possible states must be defined for each uncertainty region. Many practical applications require a non-uniform probability model such as the skew-normal distribution. However, there is a marked lack of investigation on this problem – relevant MOD techniques define only the uncertainty boundaries. To support probabilistic trajectory query processing with a non-uniform distribution model, the query system needs to accurately estimate the "most likely" states of the objects, given a time point, in order to determine the peak points of the corresponding probability density surfaces of all possible states.

The rest of this paper is organized as follows. Section 2 presents related work and proposes a framework for processing trajectory queries. Section 3 describes conventional line-based trajectory models, and proposes a generalized parametric trajectory model. In Section 4, we quantify our discussion by comparing real trajectory gathered from a GPS device with analytical results from our trajectory models and from conventional line-based models. Conclusions and future research directions are discussed in Section 5.

2 Related Work and Proposed Application

A moving object's trajectory stored in a database is a sequence of connected segments in space-time, and each segment has two endpoints that are consecutively reported (factual) states. Only reported states are stored in the database (due to the fact that a database cannot be continuously updated) [13]. Given the theoretical possibility of an infinite number of states between two reported states, a mathematical model and computational approach is required to manage the in-between and future states. For these reasons, a number of uncertainty models have been proposed.

One of the proposed models is as follows: at any point in time, the spatial state of each object must be within a certain distance *d*, of its last reported state; if the object moves further than *d*, it reports its new state and, if necessary, changes *d* for future updates [12]. Another model, known as the network movement model [12], is a one-dimensional model assuming that, at any point in time, the object is moving along one of a set of predetermined straight lines. Another model in [5] assumes that an update occurs whenever the object's velocity (speed or direction) changes. Other models assume that the object travels with known velocity along a straight line, but can deviate from this path by a certain distance [8, 9].

A spatial model of uncertainty in the recorded trajectories is found in [4]. Assuming the maximum velocity of an object (one of the properties of the object's *dynamics*) is known, all possible states of the object during the time interval between two consecutive observations lie on a certain ellipse (called the error ellipse). With this model, any update policy can be used to optimize the database system: Several update policies (also known as the dead-reckoning policies), such as the fixed time-interval update, plain dead-reckoning, and adaptive dead-reckoning, have been separately investigated [11]. Although we can generalize this ellipse model to 3- or higher dimensional moving objects using the notion of hyper-ellipse, this model is inefficient for spatiotemporal range queries: a 3-dimensional spatiotemporal query window whose extent is 0.1 along every dimension occupies 0.1^3 in the original space-time but 0.1^2 (a much larger portion) in the projected space, resulting in an enlarged search space. A spatiotemporal uncertainty model that produces 3-dimensional cylindrical uncertainty regions representing the past uncertainties of trajectories is found in [10].

More recently developed uncertainty model reported in [14] formally defines both past and future "spatiotemporal" uncertainties of trajectories of any dimensionality. Figure 1 shows two examples of this trajectory uncertainty model, given the maximum velocity M_v. The *uncertainty region* of the object during t_i-t_j is defined to be the overlap between the two cones whose tops are, respectively, P_1 and P_2. The *snapshot* of the object at any time point t_k that is between t_i and t_j is the uncertainty region's cross section produced by the cutting plane *time* = t_k.

One can further improve this model (i.e., reduce the size of the spatiotemporal uncertainty region) by taking into account more dynamics and derivatives related to velocity, acceleration, and even higher derivatives (please contact the authors for more details).

Because of this uncertainty, each result object of a query referring to the trajectories must be associated with the probability (or likelihood) that the item really satisfies the query predicate. This is more pronounced when the uncertainty regions are very large. As an extreme case, let us suppose that the uncertainty regions are bounded only by the boundaries of the data space (i.e., $M_v = \infty$). In this case, given any query point, or region, at a point t in time, every object has a non-zero probability that it intersects the query point or region at t, except for the ones that have an exact state at t. Therefore, in order to properly adopt existing trajectory query processing algorithms (e.g., [2, 3]), one needs a probability distribution model that can represent the probability distribution of all possible states of each snapshot.

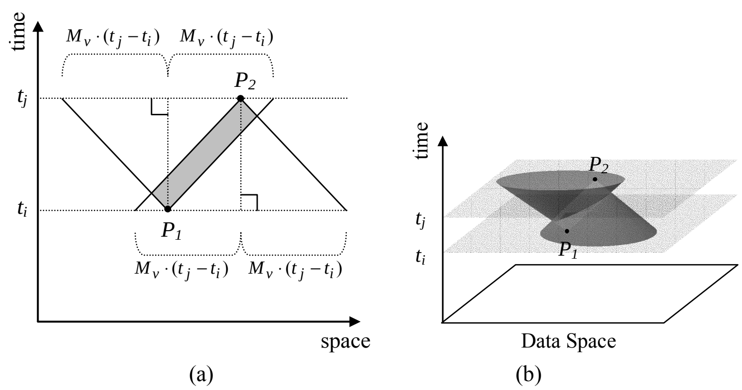


Fig. 1. Spatiotemporal uncertainty region representing a trajectory segment: (a) in a 2-dimensional space-time; (b) in a 3-dimensional space-time

As shown in Figures 2a and 2b, given a time point, the linear trajectory interpolation model can be used to determine the peak of the skew-normal probability density surface [1, 7] of all possible states. In this case, because the query range, q, reaches the peak point, the linear model reports 50 % of the probability that the actual state is covered by q (Figure 2b). However, considering a more skewed distribution of possible states (Figures 2a and 2c), the linear model reports too optimistic results because the probability becomes much lower than 50%. Our proposed "most likely" trajectory model will more accurately estimate the actual trajectory, resulting in a more accurate estimation of the probability.

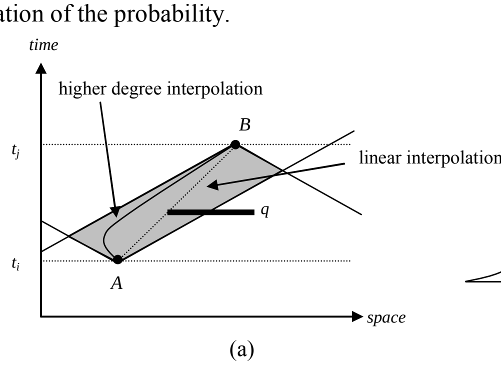


Fig. 2. (a) shows a trajectory segment that can possibly intersect q, where q is a query region; (b) and (c) show skew-normal probability distributions of a snapshot that overlaps q

3 Most Likely Trajectory: A Parametric Approach

We propose a generalized, parametric trajectory interpolation model (Definition 1) connecting any two consecutive reported states.

Definition 1. Given a pair $<P_i\ P_j>$ of consecutive reported states, the most likely trajectory segment of degree $2n+1$ is defined as follows:

$P^{(0)}(u) = \sum_{k=0}^{2n+1} a_k u^k$, where $0 \le u \le 1$ is a free-variable parameter, $n \ge 0$ is the number of derivatives written in each reported state P and the coefficients are derived by solving the following constraints for a_0, a_1, …, a_{2n+1}: for (l=0; $l \le n$; l++) $\{P^{(l)}(u=0)=P_i^{(l)}$ and $P^{(l)}(u=1)=P_j^{(l)}\}$, where $P^{(l)}$ is the l^{th} derivative of a state P.

Given any pair $<P_i\ P_j>$ of consecutive reported states, most MOD techniques, as surveyed in [10, 13], uses the linear interpolation (i.e., a special case of Definition 1 with n=0) assuming that the velocity of the object is fixed during the time period of the segment. The first non-linear model investigated in [13] is a special case of Definition 1 with n=1 assuming that the acceleration changes linearly in only one direction during the period. In contrast, the 5^{th} degree trajectory (n=2), which is possible in our model, can accommodate smoothly changing accelerations. Considering fast changing objects that are affected by momentum, not only the locations but also some higher order derivatives change without angle. Unlike fast changing objects, some slowly changing objects (e.g., animals and humans) can change velocity more abruptly. Importantly, the generalized trajectory interpolation model in Definition 1 provides a basis for investigating optimization solutions that, given a proper description of a moving-objects set, can adaptively choose the most efficient equation (i.e., $n \ge 1$).

3.1 Specialization 1

Considering a discrete sequence of reported states each of which is a location-time <X, Y, Z, T>, where X, Y, and Z are spatial coordinates and T is a time value, Definition 1 can be specialized to obtain a connected sequence of spatiotemporal linear trajectory segments that passes through the joints (reported states) $<P_0^{(0)}\ P_1^{(0)}\ P_2^{(0)}\ \ldots\ P_n^{(0)}>$ where, for all k = 0,..,n, $P_k^{(0)}$ is a location-time $<X_k, Y_k, Z_k, T_k>$ in the data space-time.

To spatiotemporally connect consecutive reported states $P_i^{(0)}$ and $P_j^{(0)}$, where i = 0,..,n-1 and $j = i$+1, we use the following parametric linear function (a special case of Definition 1 with n = 0): $P^{(0)}(u) = a_0+a_1u$. To derive the coefficients, solve the following constraints for a_0 and a_1: $P^{(0)}(u=0)= P_i^{(0)}$ and $P^{(0)}(u=1)= P_j^{(0)}$. Substituting the derived coefficients into the parametric linear function, we have the following function:

$$P^{(0)}(u) = \begin{bmatrix} 1 & u \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} P_i^{(0)} \\ P_j^{(0)} \end{bmatrix}, \quad \textbf{(1)}$$

where $0 \le u \le 1$.

3.2 Specialization 2

Considering a sequence of reported trajectory states each of which consists of a location-time <X, Y, Z, T> and a velocity <X'=ΔX/ΔT, Y'=ΔY/ΔT, Z'=ΔZ/ΔT>, one can use a parametric cubic function $P^{(0)}(u) = a_0+a_1u+a_2u^2+a_3u^3$ (a special case of Definition 1) to connect each pair of two consecutive joint-velocities <$P_i^{(0)}$ $P_i^{(1)}$> and <$P_j^{(0)}$ $P_j^{(1)}$>, where $P_i^{(0)}$ = <X_i, Y_i, Z_i, T_i>, $P_j^{(0)}$ = <X_j, Y_j, Z_j, T_j>, $P_i^{(1)}$ = <$X_i'\cdot(T_j-T_i)$, $Y_i'\cdot(T_j-T_i)$, $Z_i'\cdot(T_j-T_i)$, T_j-T_i>, and $P_j^{(1)}$ = <$X_j'\cdot(T_j-T_i)$, $Y_j'\cdot(T_j-T_i)$, $Z_j'\cdot(T_j-T_i)$, T_j-T_i>.

One can derive the coefficients of $P^{(0)}(u) = a_0+a_1u+a_2u^2+a_3u^3$ by solving the following constraints for a_0, a_1, a_2, and a_3: $P^{(0)}(u{=}0) = P_i^{(0)}$; $P^{(0)}(u{=}1) = P_j^{(0)}$; $P^{(1)}(u{=}0) = P_i^{(1)}$; $P^{(1)}(u{=}1) = P_j^{(1)}$. Substituting these coefficients into the polynomial equation, we have the following function:

$$P^{(0)}(u) = [1 \quad u \quad u^2 \quad u^3]\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3 & 3 & -2 & -1 \\ 2 & -2 & 1 & 1 \end{bmatrix}\begin{bmatrix} P_i^{(0)} \\ P_j^{(0)} \\ P_i^{(1)} \\ P_j^{(1)} \end{bmatrix}, \qquad \textbf{(2)}$$

where $0 \le u \le 1$.

3.3 Specialization 3

Considering a sequence of reported states each of which consists of a location-time <X, Y, Z, T>, a velocity <X'=ΔX/ΔT, Y'=ΔY/ΔT, Z'=ΔZ/ΔT>, and an acceleration <X''=ΔX'/ΔT, Y''=ΔY'/ΔT, Z''=ΔZ'/ΔT>, we can use a parametric function of degree 5, $P^{(0)}(u) = a_0+a_1u+a_2u^2+a_3u^3+a_4u^4+a_5u^5$ (a special case of Definition 1), to connect each pair of two consecutive joint-velocity-accelerations <$P_i^{(0)}$ $P_i^{(1)}$ $P_i^{(2)}$> and <$P_j^{(0)}$ $P_j^{(1)}$ $P_j^{(2)}$>, where $P_i^{(0)}$ = <X_i, Y_i, Z_i, T_i>; $P_j^{(0)}$ = <X_j, Y_j, Z_j, T_j>; $P_i^{(1)}$ = <$X_i'\cdot(T_j-T_i)$, $Y_i'\cdot(T_j-T_i)$, $Z_i'\cdot(T_j-T_i)$, T_j-T_i>; $P_j^{(1)}$ = <$X_j'\cdot(T_j-T_i)$, $Y_j'\cdot(T_j-T_i)$, $Z_j'\cdot(T_j-T_i)$, T_j-T_i>; $P_i^{(2)}$ = <$X_i''\cdot(T_j-T_i)^2$, $Y_i''\cdot(T_j-T_i)^2$, $Z_i''\cdot(T_j-T_i)^2$, 0>; $P_j^{(2)}$ = <$X_j''\cdot(T_j-T_i)^2$, $Y_j''\cdot(T_j-T_i)^2$, $Z_j''\cdot(T_j-T_i)^2$, 0>.

One can derive the coefficients of $P^{(0)}(u) = a_0+a_1u+a_2u^2+a_3u^3+a_4u^4+a_5u^5$ by solving the following constraints for a_0, a_1, a_2, a_3, a_4, and a_5: $P^{(0)}(u{=}0) = P_i^{(0)}$; $P^{(0)}(u{=}1) = P_j^{(0)}$; $P^{(1)}(u{=}0) = P_i^{(1)}$; $P^{(1)}(u{=}1) = P_j^{(1)}$; $P^{(2)}(u{=}0) = P_i^{(2)}$; $P^{(2)}(u{=}1) = P_j^{(2)}$. Substituting these coefficients into the parametric function, we have the following function:

$$P^{(0)}(u) = [1 \quad u \quad u^2 \quad u^3 \quad u^4 \quad u^5]\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0 \\ -10 & 10 & -6 & -4 & -1.5 & 0.5 \\ 15 & -15 & 8 & 7 & 1.5 & -1 \\ -6 & 6 & -3 & -3 & -0.5 & 0.5 \end{bmatrix}\begin{bmatrix} P_i^{(0)} \\ P_j^{(0)} \\ P_i^{(1)} \\ P_j^{(1)} \\ P_i^{(2)} \\ P_j^{(2)} \end{bmatrix}, \qquad \textbf{(3)}$$

where $0 \le u \le 1$.

4. Experiment

To compare the three specialized models of the most likely trajectory, we placed a GPS device (Trimgle Navigation's ProXRS Receiver with GPS logger) in a car and drove from a location near the north boundary of Denver, Colorado, to Loveland, Colorado, USA along Interstate highway 25. Every second, we logged a spatiotemporal data from the GPS device. The acceleration vector of each logged state was calculated using the second degree approximation on the recorded velocities (for the first and the last trajectory points, the first degree approximation was used). Then, we divided the recorded trajectory states, each of which consists of a location-time <longitude, latitude, altitude, time>, a velocity <longitude', latitude', altitude'>, and a derived acceleration <longitude'', latitude'', altitude''>, into two subsets: Set1 consisted of 482 recorded states logged every second; Set2 consisted of 742 recorded states logged every second. Note that Set1 represents driving on a relatively straight road and Set2 represents some winding trajectory.

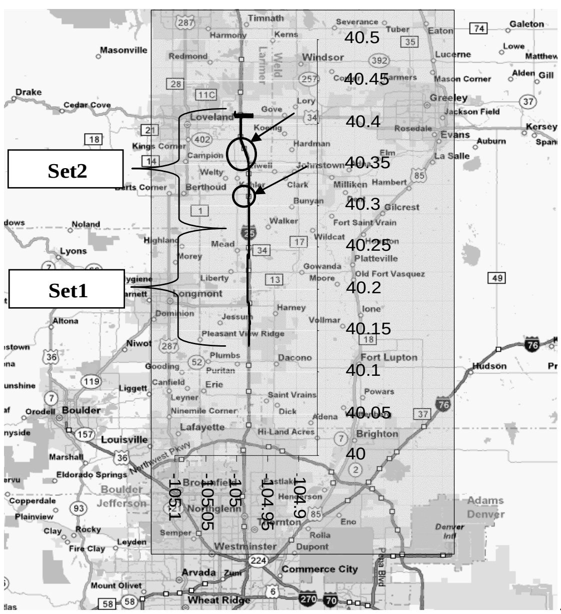


Fig. 3. A real trajectory states collected

For each of Set1 and Set2, we randomly selected logged spatiotemporal records with various sampling ratios. The three specialized models (i.e., Equations 1, 2, and 3) were used to connect the selected samples (4-dimensional spatiotemporal trajectories were produced). Figure 4 gives a magnified view of the circled parts in Figure 3 (the sampling ratio was about 5%; for illustration sake, we projected the 4-dimensional spatiotemporal trajectories onto the XY-plane).

For each of the three specialized models, we quantified the actual deviations between the non-sampled real location-times and the corresponding estimates. In all tested cases, the higher degree models, Equations 2 and 3, produced significantly smaller average deviations (up to more than 3 times smaller, Figure 5) and standard error deviations (Figure 6) than the linear model Equation 1. For example, with 37 sampled out of 742 states in Set2, the average distance deviation of Equations 1, 2, and 3 were 19, 21, and 62 meters, respectively. Their maximum deviations in this section were 162, 231, and 683 meters, respectively. As shown in Figure 6, the standard deviations in this section were 35, 41, and 134 meters.

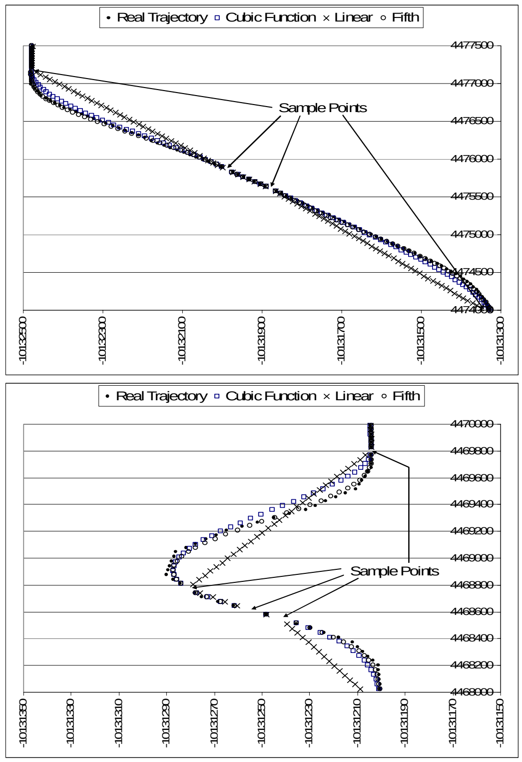
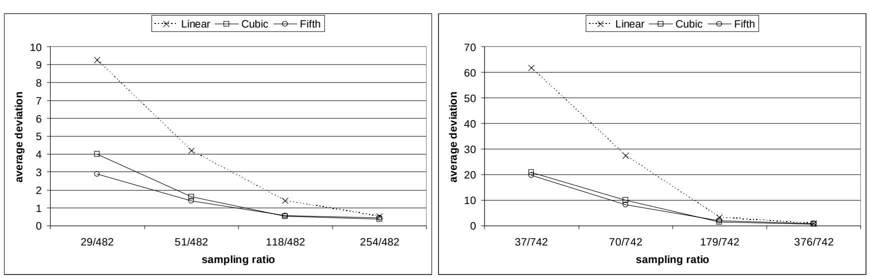
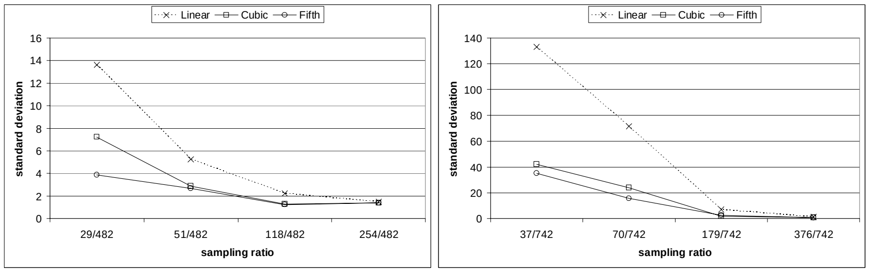


Fig. 4. Two parts of the trajectory projected onto XY-plane and elongated along X-axis for better visual comparison: the X-axis is longitude in meters; the Y-axis is latitude in meters; the sampling ratio was 37/742 (≈ 5%)



(a) (b)

Fig. 5. Average spatial deviations (in meters) with various sampling ratios on (a) Set1 and (b) Set2



(a) (b)

Fig. 6. Standard error deviations (in meters): (a) Set1 and (b) Set2

For all cases, we observed that the difference between the linear model and the higher degree models was significant and became greater when the actual trajectory is winding (Set2) (Figures 5 and 6). In most cases, the 5th degree model excelled the 3rd degree model.

In our experiments using a Linux machine equipped with an Intel Pentium III 800MHz and 256MB main memory space, the linear model took 0.7 – 0.8 microseconds of CPU time to interpolate a point in-between two consecutive joint states. The cubic model and the 5th degree model required 4.3 – 4.6 microseconds and 10.8 – 11.1 microseconds, respectively.

5 Discussion

For the tested cases, both velocity and acceleration varied smoothly because of the momentum gained while moving. However, the linear model assumes that the velocity of the object is fixed during the period of each segment and the 3rd degree model assumes that the acceleration changes linearly in only one direction during the period. The 5th degree model considers both varying velocity and acceleration so it can present the most accurate trajectories with the same set of recorded states, which can be used for a better estimation of the probabilistic trajectory queries.

Moreover, the performance of MOD can be significantly enhanced as follows: 1) given a maximum allowed deviation between a point of a database trajectory and the corresponding point of the real trajectory, a smaller number of trajectory update transactions is required; 2) a reduced amount of secondary storage space is occupied by trajectories, 3) the trajectory index structure size is reduced; 4) a smaller number of disk I/Os are performed in processing trajectory update transactions and trajectory queries. For example, given a maximum deviation threshold (e.g., 62 meters) for update, the higher degree models reduce the number required updates by a factor of up to 5. In typical situations, this has more significant impacts on the system performance and scalability than the CPU overhead. However, investigating the relevant issues in developing an adaptive system that can automatically balance the CPU-I/O trade-offs using various trajectory models will be practically viable for some application systems that have a severely limited CPU power or extremely fast secondary storage.

By taking into account for how the environment may be variably constraining movement and thus variably affecting the set of possible positions of the object, one can contextualize (modify) the probability distribution as well as the most likely trajectory state of each individual snapshot. A related preliminary study, contextualizing the probability distribution of vehicle whereabouts with geographic road data sets, can be found in [6]. If the contextualization of uncertainty regions can be properly performed, the spatiotemporal regions requiring indexing can also be commensurately limited and the query results will be associated with more probable likelihoods. We reserve this as our future work.

References

1. Azzalini and A. Capitanio, "Statistical applications of the multivariate skew-normal distribution. Journal of the Royal Statistical Society," B(61): 579—602, 1999.
2. R. Cheng, D.V. Kalashnikov, and S. Prabhakar, "Evaluating Probabilistic Queries over Imprecise Data," ACM SIGMOD, 551—562, 2003.
3. R. Cheng, D.V. Kalashnikov, and S. Prabhakar, "Querying Imprecise Data in Moving Object Environments," IEEE TKDE, 16:9, 1112—1126, 2004.
4. D. Pfoser and C.S. Jensen, "Capturing the Uncertainty of Moving-Objects Representations," SSDBM 123—132, 1999.
5. D. Pfoser and C.S. Jensen, "Querying the Trajectories of On-Line Mobile Objects," ACM MobiDE, 66—73, 2001.
6. S.D. Prager, "Environmental Contextualization of Uncertainty for Moving Objects," GeoComputation, Ann Arbor, Michigan, 2005.
7. Development Core Team R: A language and environment for statistical computing, R Foundation for Statistical Computing, Vienna, Austria, 2004.
8. A.P. Sistla, O. Wolfson, S. Chamberlain, and S. Dao, "Querying the Uncertain Position of Moving Objects. Temporal Databases," Research and Practice, no. 1399, 1998.
9. G. Trajcevski, O. Wolfson, F. Zhang, and S. Chamberlain, "The Geometry of Uncertainty in Moving Object Databases," Int'l Conf. on Extending Database Technology, 2002.
10. G. Trajcevski, O. Wolfson, K. Hinrichs, and S. Chamberlain, "Managing Uncertainty in Moving Objects Databases", ACM Transactions on Database Systems, Vol. 29, No. 3, 463-507, 2004.
11. O. Wolfson, L. Jiang, A.P. Sistla, S. Chamberlain, N. Rishe, and M. Deng, "Databases for Tracking Mobile Units in Real Time," ICDT International Conference on Database Theory, Lecture Notes in Computer Science, Springer-Verlag, Berlin Heidelberg New York, 169—186, 1998.
12. O. Wolfson, A.P. Sistla, S. Chamberlain, and Y. Yesha, "Updating and Querying Databases that Track Mobile Units. Distributed and Parallel Databases," 7:3, 257—387, 1999.
13. B. Yu, S.H. Kim, T. Bailey, and R. Gamboa, "Curve-Based Representation of Moving Object Trajectories," IEEE International Database Engineering and Applications Symposium, 419—425, 2004.
14. B. Yu, S.D. Prager, and T. Bailey, "The Isosceles-Triangle Uncertainty Model: A Spatiotemporal Uncertainty Model for Continuously Changing Data," In: C. Gold (Eds.), Workshop on Dynamic & Multi-dimensional GIS, International Society for Photogrammetry and Remote Sensing, XXXVI:2/W29, ISPRS ICWG II/IV, Wales, UK, 179—183, 2005.