Journal of Korean
Data & Information Science Society
2008, Vol. 19, No. 3, pp. 979 ~ 994

Continuous Location Tracking Algorithm for Moving Position Data

Yoon-Ae Ahn[1)]

Abstract

Moving objects are spatio-temporal data that change their location or shape continuously over time. Generally, if continuously moving objects are managed by a conventional database management system, the system cannot properly process the past and future location which is not stored in the database. Up to now, for the purpose of location tracking which is not stored, the linear interpolation to estimate the past location has been usually used. It is suitable for the moving objects on linear route, not curved route. In this paper, we propose a past location tracking algorithm for a moving object on curved routes, and also suggest a future location tracking algorithm using some past location information. We found that the proposed location tracking algorithm has higher accuracy than the linear interpolation function.

Keywords: Location Tracking; Moving Position Data; Uncertainty.

1. Introduction

Moving objects are spatio-temporal data, which continuously change their location or shape over time. When these continuously moving objects are managed by a conventional database, it is difficult to store all the location information changed over time in the database. Therefore, a time period for sampling data is determined and the location information of moving objects is discretely stored in the system for every time period. However, if the continuously moving objects are managed as discrete model, we may have a problem that the system cannot properly answer to the query about the uncertain past and future location information. This problem is caused by the uncertain location information of the

1) Professor, Department of Computer Multimedia, Chungju National University, Chungju, 380-702, Korea
E-mail : yeahn@cjnu.ac.kr

moving objects which is not stored in database. For the purpose of solving this uncertainty problem, a new model is required to answer properly to the location queries and to reduce the location uncertainty of the moving objects resulting from storing and managing continuously moving data in discrete model.

Pfoser and Tryfona(2000) and Güting et al(2000) proposed the linear interpolation method for the uncertain location of moving point objects, which was not sampled. However, there is no mention about the future location estimation method. Furthermore, a method to integrate spatio-temporal data and error information into relational database schema was proposed. Forlizzi et al(2000) proposed a data model and data structure for moving objects databases. In defining the location of the moving point objects, the method to measure the non-sampled uncertain location using linear interpolation was proposed. However, it is a suggestion of the methodology, which lacks the accuracy verification through a concrete experiment. Nothing about future location estimation method has been mentioned.

Cruz et al(2002) proposed a method to predict future location of moving objects based on the present location, speed, and direction of the object, but did not present a moving location estimation method of the uncertain past moving objects. The current research refers to the location update policy, and uncertainty of the location related to database specifying. It also suggests ways to deal with queries of tracking current and future locations.

Lema et al(2003) and Ilarri et al(2006) studied problems which the design and embodiment of spatio-temporal database system has, suggests the structure of spatio-temporal database system, and embodies it partly. There are some other studies on representation of periodic moving objects in databases, uncertainty management for network constrained moving objects, managing moving Objects on dynamic transportation networks, and managing uncertainty in moving objects databases. (Ding and Güting(2004 a), Ding and Güting(2004 b), Trajcevski et al(2004), Behr et al(2006))

Almeida and Güting(2005) studied about indexing the trajectories of moving objects in networks and Güting(2007) proposed moving object languages. There is a research on moving objects in on-line and network environment. (Cao et al(2005), Trajcevski et al(2006)) However, neither does the proposed model store history information of the uncertain past movement location of moving objects, nor does it indicate the method to find out the past location.

Up to the present, works have been focused on either past or future uncertain location estimation. Previous works cannot estimate both past and future moving location simultaneously in the same database.

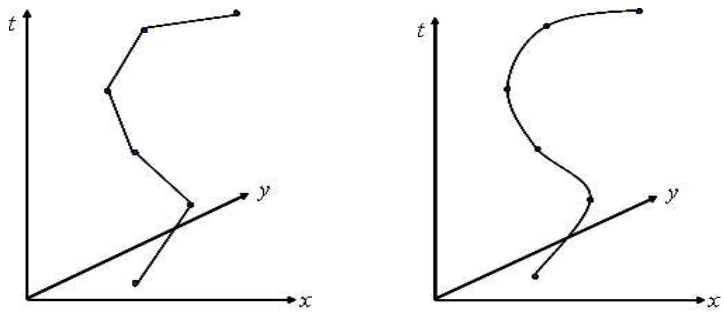
Therefore, we propose a method that predicts both past and future locations of the moving object using the same movement information. The past movement information is stored at a regular sampling period in a history database. For estimating locations of the moving object at past random period which is not sampled, the piecewise cubic spline interpolation is used, not the traditional linear interpolation. And the location estimation at future random period uses the average values of the past moving information that was stored in the history database.

The description of this paper is as follows. In Section 2, the concept of the moving object and the modeling types of the moving position data. And the past and future location estimation functions of moving objects are described. In Section 3, we present the continuous location tracking algorithms of moving position data. Section 4 analyzes the errors of predicted location through the experiments of the proposed location estimation method. Finally, Section 5 concludes this paper and mentions some directions for further research.

2. Data Model and Location Estimation Function

2.1 Definition of Moving Position Data

The positions of moving objects are continuously changing over time. For modeling these moving objects, we can consider both continuous and discrete models. The continuous model allows us to represent the moving object in terms of infinite sets of points, and to view the moving point as a continuous curve in the 3D space. This model can accurately describe the motion information, but it is inadequate to be implemented since we cannot store or manipulate infinite points in computer. On the other hand, discrete model allows us to describe the moving object in terms of finite sets of points and to view the moving point as a polyline in the 3D space. This can be implemented by showing the motion information of the moving object in terms of approximate values.



(a) Discrete Representation (b) Continuous Representation

Figure 1. Moving Position Data Model

In Figure 1, the location change scenarios of the moving objects are used as a basic tool for evaluating spatio-temporal uncertainty. The location change of the moving objects can occur discretely or continuously, and can be recorded as time points or time periods. Thus, it is composed of three-dimensions. T axis can express the time value of past, present, and future. However, the values stored in the database are actually those of the past and present, and the future is processed by a specific operation.

In this paper, moving objects are assumed to be represented as moving points. Moving locations of the objects are supposed to be sampled in discrete points and the changes are expressed on two-dimensional space. Moving objects are specified through location estimation function and estimation of uncertain information using a relational database. Database is composed of two relations to store sampled location of moving objects and to have the following schema:

OBJECT(Oid, Code, Name, Type, Etc.)
HISTORY(Code, Time, X, Y)

In OBJECT relation, Oid is the key value as an identifier of the moving object. The properties of each object including non-spatial information are Code, Name, Type, and Etc in the OBJECT relation of the above schema. Movement information relation, HISTORY, stores historical information regarding sampled time and location of the object. The attributes in the HISTORY relation are Code, Time, X, and Y. The Code describes key value and the X and Y specify sampled time point and location coordinate values. This paper proposes past location estimation function using the piecewise cubic spline interpolation.

2.2 Location Estimation Function

To find out the uncertain past location of the gradually curved movement, the piecewise cubic spline interpolation is used as follows. The function is used for estimation of the past location and based on the definitions 1 and 2.

[Definition 1] Past location estimation function

Past location estimation function models the continuous location changes of the moving objects and it returns all locations of the objects within a random time period of $T_i = [t_i, t_{i+1}]$. Past location estimation function is described as PLEF: $t_k(x_{t_k}, y_{t_k})$, when random time point t_k is inserted and location coordinates (x_{t_k}, y_{t_k}) are produced. In this case, the range of t_k is $t_0 \le t_k \le now$. The t_0 is the starting time point stored in the OBJECT relation. The now is the latest time point stored in the HISTORY relation.

To define the past location estimation function PLEF of a random moving object O_A using cubic spline interpolation (Jeong et al(2008)), the history set of moving objects is defined as follows. The data of the moving objects are stored in three-dimensional set, which is formulated with two-dimensional spatial information (x, y) and one-dimensional time information. However, the location estimation function does not deal with three-dimensional sets but only with two-dimensional set of x- and y-axis over time.

[Definition 2] History set of moving objects

All P values are the history sets of the random moving object O_A. Then a three-dimensional history set is produced. Here, $P=\{p_0(t_0,x_0,y_0),p_1(t_1,x_1,y_1),\cdots p_n(t_n,x_n,y_n)\}$. In P, $p_0(t_0,x_0,y_0)$ is the location coordinates (x_0,y_0) of time point t_0 that is stored first in the history relation. And $p_n(t_n,x_n,y_n)$ is the location coordinates (x_n,y_n) of time point t_n that is stored the latest. The three-dimensional history set P is divided into two-dimensional sets of x and y over time t. x-axis history set is $P_x=\left\{p_{x_0}(t_0,x_0),p_{x_1}(t_1,x_1),\cdots p_{x_n}(t_n,x_n)\right\}$, and y-axis history set is $P_y=\left\{p_{y_0}(t_0,y_0),p_{y_1}(t_1,y_1),\cdots p_{y_n}(t_n,y_n)\right\}$.

Based on the definitions 1 and 2, the piecewise cubic spline polynomial to find x and y of past random time point t_k can produce two cubic spline functions $x=S_i(t_k)$ and $y=S_i(t_k)$ using two-dimensional history set P_x and P_y. During the random time period of $T_i=[t_i,t_{i+1}]$, $S_i(t_k)$ is the function to find x and y coordinate of random past time point t_k, which meets the condition of $t_i<t_k<t_{i+1}$.

Most of the applications using the moving object are focused on the moving track control of a car and the estimation of moving route. A moving track management can provide the query results using the history information stored in a database, but the estimation of the future moving track should consider various situations suitable for a special application. We propose a simple location estimation method using the past moving information stored in a history relation, not in a database about information of the speed and direction. The future moving object location is processed by a future location estimation function, and we use the following definition.

[Definition 3] Future location estimation function

A future location estimation function is described as FLEF: $t_f(x_{t_f},y_{t_f})$. We obtain t_f of future random time point as input value and consequently return location coordinates of (x_{t_f},y_{t_f}). Then, the range of time point t_f is $now<t_f$. The now is the latest time point stored in a moving information relation, and t_f always describes the future random special time point.

The location estimation function to obtain x, y coordinates of future random time point t_f uses denotations as definition 3 on a target of the x-axis history coordination P_x and the y-axis history coordination P_y. We obtain the future location estimation of the moving object O_A using x=FLEF(t_f) and y=FLEF(t_f) function.

3. Location Tracking Algorithm

3.1 Past Location Estimation Algorithm

The past location estimation algorithm for random time point t_k, which is not stored in the database, is as follows. The algorithm uses object and history relation mentioned in section 2.1, and the piecewise cubic spline interpolation polynomial using 4 points is used for the function.

Algorithm 1. Past Location Tracking of Moving Position

```
Algorithm PLEF(oid_O, t_k)
Input: oid_O(the identifier of object OA), t_k(random past time point)
Output: x_t_k(x value of time point t_k), y_t_k(y value of time point t_k)
OBJECT: object information relation
HISTORY: history information relation
Begin
If (Oid = oid_O in OBJECT relation) Then
    code ← code value of the object
    If (Code = code and Time = t_k in HISTORY relation) Then
        x_t_k ← X value of the searched tuple
        y_t_k ← Y value of the searched tuple
    Else
        search two previous and following tuples of time period t_k in HISTORY relation
        Time1, Time2, Time3, Time4 ← time values of 4 tuples
        X1, X2, X3, X4 ← X values of 4 tuples
        Y1, Y2, Y3, Y4 ← Y values of 4 tuples
        x_t_k ← CSIP(Time, X, t_k)
        y_t_k ← CSIP(Time, Y, t_k)
Else
    x_t_k and y_t_k ← error value
return x_t_k and y_t_k
End
```

The past location estimation function PLEF described in Algorithm 1 with the input of the identifier oid_O of the random moving object OA. The random past time point t_k produces the output of the (x, y) coordinates of x_t_k and y_t_k at the time point t_k. First, check if the input identifier of the object oid_O is the instance of Oid which is stored in object relation. The input identifier is not included in OBJECT relation, assign the error value to the result variable x_t_k and y_t_k, and finish the function.

In other cases, search the location coordinates of the object at t_k time point, which is stored in the HISTORY relation. If the location values at t_k time point exist in movement information relation, finish the function after converting the searched values to the result. Otherwise, call piecewise cubic spline function and find the values of x_t_k and y_t_k. In this case, the input Time is the values of the 4 points to call the CSIP function. X and Y are x and y coordinates of the 4 points. The locations of the 4 points are two history tuples just before the input time point t_k and two history tuples right after t_k. The algorithm to process the spline function is described in Algorithm 2.

Algorithm 2. Cubic Spline Interpolation Polynomial

```
Algorithm CSIP(time, loc, t_k)
Input: 1) The set of coordinates to produce three-dimensional spline polynomial
          : (time0,loc0), (time1,loc1), (time2,loc2), (time3,loc3)
       2) Random past time point t_k
Output: Location value of x or y at a random time point t_k(variable: Coord_XorY)
N: The number of input coordinates (N = 4 at the start)
Begin
Do i = 0, N-2
   h[i] ← time[i+1] - time[i]
End Do
Do i = 0, N-3
   m4[i] ← 6 * ((loc[i+2] - loc[i+1]) / h[i+1] - (loc[i+1] - loc[i]) / h[i])
End Do
m1[1] ← h[1]
m2[0] ← 3 * (h[0] + 2 * h[1])
m2[1] ← 3 * (h[2] + 2 * h[1])
m3[0] ← h[1]
Do i = 1, N-3
   m2[i] ← m2[i] - ((m1[i] / m2[i-1]) * m3[i-1])
   m4[i] ← m4[i] - ((m1[i] / m2[i-1]) * m4[i-1])
End Do
s[1] ← m4[1] / m2[1], s[0] ← s[1], s[3] ← s[2]
Do i = 0, N-2
   a[i] ← (s[i+1] - s[i]) / 6 * h[i]
   b[i] ← s[i] / 2
   c[i] ← (loc[i+1] - loc[i]) / h[i] - (2 * h[i] * s[i] + h[i] * s[i+1]) / 6
   d[i] ← loc[i]
If (time[i] < t_k < time[i+1]) Then
   Coord_XorY ← d[i] + (t_k-time[i]) * (c[i] + (t_k-time[i])* (b[i] + a[i]*(t_k-time[i])))
End Do
Return Coord_XorY
```

To the CSIP function of Algorithm 2, the sets of coordinates of the 4 points to produce the three-dimensional spline polynomial and the random past time point t_k are the inputs. The sets of coordinates of the 4 points are in the form of (time0, loc0), (time1, loc1), (time2, loc2), and (time3, loc3). In this case, time[i] is the time value stored in the history information relation, and loc[i] is the x or y coordinate value of the location of the moving objects at the time point of time[i].

The resulting value of this function is the value stored at the variable Coord_XorY at the time point t_k, which is produced from the spline function of input coordinates. The algorithm of CSIP function shows the same development as the three-dimensional spline interpolation polynomial. First, calculate the 4 coefficients of spline polynomial: a[i], b[i], c[i], and d[i]. Using the coefficients and the input coordinates of the 4 points, find x and y values at the time point t_k and return the result.

3.2 Future Location Estimation Algorithm

The future location estimation algorithm is used for OBJECT relation in Section 2.1 and HISTORY relation. Denotations of the future location estimation function are shown in the following expression.

- ◦ t_f : input value of FLEF(t_f), meaning future special time point
- ◦ d_{mx} : average moving distance on x-axis
- ◦ d_{my} : average moving distance on y-axis
- ◦ t_{mf} : moving time from now to future special time point
- ◦ d_{tx} : total moving distance on x-axis
- ◦ d_{ty} : total moving distance on y-axis
- ◦ t_{tx} : total moving time on x-axis
- ◦ t_{ty} : total moving time on y-axis
- ◦ t_{xn} : the latest time point of x-axis history coordinates
- ◦ t_{yn} : the latest time point of y-axis history coordinates
- ◦ x_n : the latest time point on the x-axis
- ◦ y_n : the latest time point on the y-axis

The location estimation function to obtain x, y coordinates of future random time point t_f uses denotations as definition 3 on a target of the x-axis history coordination P_x and the y-axis history coordination P_y. We obtain the future location estimation of the moving object O_A using $x = FLEF(t_f)$ and $y = FLEF(t_f)$ function. First, x coordinate value of the future special time point is obtained from the following equation from the history set P_x :

$$x = (d_{mx} \times t_{mf}) + x_n$$
$$= ((d_{tx} \div t_{tx}) \times (t_f - t_{xn})) + x_n$$
$$= (\sum_{i=2}^{n}(x_n - x_{n-1}) \div (\sum_{i=2}^{n}(t_{xn} - t_{xn-1})) \times (t_f - t_{xn}) + x_n$$

y coordinate value of the future special time point is also obtained from the history set P_y :

$$y = (d_{my} \times t_{mf}) + y_n$$
$$= ((d_{ty} \div t_{ty}) \times (t_f - t_{yn})) + y_n$$
$$= (\sum_{i=2}^{n}(y_n - y_{n-1}) \div (\sum_{i=2}^{n}(t_{yn} - t_{yn-1})) \times (t_f - t_{yn}) + y_n$$

Input values of algorithm FLEF of Algorithm 3 are the identifier of the random moving object OA and future random time point t_f. Output values are x_t_f and y_t_f, which are (x, y) coordinates of time point t_f. First, we investigate whether the identifier oid_O of the input object is the instance of Oid. If the identifier of the input object is not included in the relation object, we assign error values to result in the variables x_t_f and y_t_f and terminate the function.

Algorithm 3. Future Location Tracking of Moving Position

```
Algorithm FLEF(oid_O, t_f )
input: oid_O(identifier of an object OA), t_f(random future time point)
output: x_t_f(x coordinates of time point t_f), y_t_f(y coordinates of time point t_f)
OBJECT: object information relation
HISTORY: moving information relation
N: the number of tuples of recent moving information used to predict future location
    coordinates.
Begin
If (Oid = oid_O, in OBJECT relation) Then
    code ← code value of the object
    If(Code = code and Time = t_f in HISTORY relation) Then
        x_t_f ← X value of the searched tuple
        y_t_f ← Y value of the searched tuple
    Else
        current ← the last time value in HISTORY relation
        current_x ← x value of current
        current_y ← y value of current
        search N tuples before current in HISTORY relation
        DO i = 1, N
            t[i] ← ith time of the search result
            x[i] ← ith xvalue of the search result
```

```
        y[i] ← ith y value of the search result
    END DO
    d_t_x, d_t_y, t_t_x ← initial value
    DO i = 0, N-2
        d_x ← x[i+1] - x[i]
        d_t_x ← d_t_x + d_x
        d_y ← y[i+1] - y[i]
        d_t_x ← d_t_x + d_x
        t_x ←t[i] +1 - t[i]
        t_t_x ← t_t_x + t_x
    End DO
    d_m_x ← d_t_x/t_t_x
    d_m_y ← d_t_y/t_t_x
    t_m_f ← t_f - current
    x_t_f ← (d_m_x * t_m_f) + current_x
    x_t_f ← (d_m_y * t_m_f) + current_y
Else x_t_f, y_t_f ← error value
  return x_t_f and y_t_f
End
```

Otherwise, we search the location coordinates of time point t_f about the input object from a history relation. If the location value at t_f exists in a HISTORY relation, we return the retrieved value as a result and terminate the function. Otherwise, finding x_t_f and y_t_f using the future location estimation function, we return the result value. Variable N used in Do loop structure of the future location estimation algorithm is the number of past history information which will be used for a future location estimation. If the value of N is bigger than 2, it does not matter, and the contents of the remaining algorithms are the same as in definition 3.

4. Performance Analysis

To analyze the location tracking algorithm of the moving objects, an experiment with virtual data samples was carried out. Under the environment of Windows XP, Java language was used. We assumed the moving object as a moving point and it was limited to a vehicle moving on a load network at a regular speed. For the location estimation experiment to calculate the difference between the predicted location and the actual location, full history information on the trajectory of moving object is necessary. In this experiment, two random trajectories on the load network were chosen. And the full location change data on the distance between the two points over time are stored on the database.

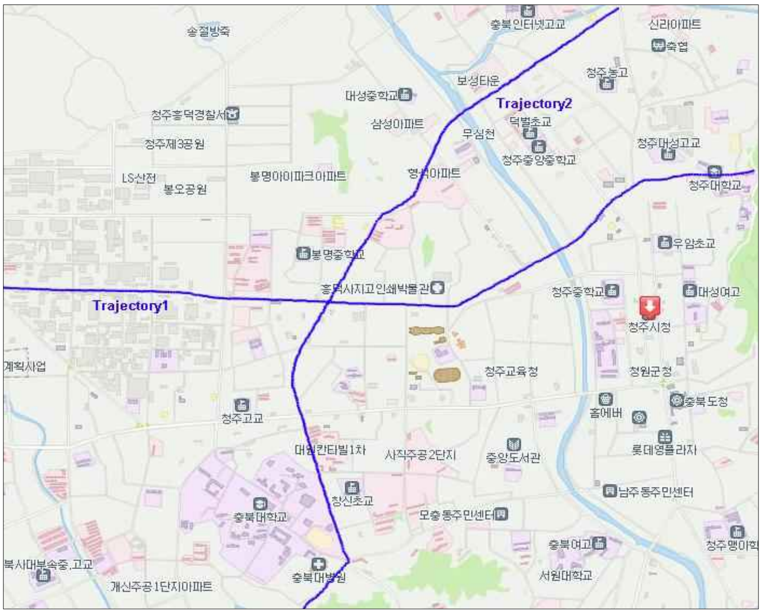


Figure 2. Moving Trajectories

Figure 2 shows sample trajectories on the map of cheongju city. The sample history relation stores the location change information of the trajectories 1 and 2. The total number of tuples is 200. The coordinate system of Java programming language is used and the unit of measuring coordinate is pixel.

4.1 Results of the Past Location Tracking

Location samples of the two trajectories were taken to be compared with the location estimation results and errors from the piecewise cubic spline function proposed in the chapter 2 regarding the location estimation relation about random past time point. Samples were taken by clicking at regular intervals on the users interface map along the trajectory and stored the coordinates in history information relation automatically. The movement time stored in the history relation was stored in accordance with the sampling intervals. Samples were taken at the interval of 2~10m for 5 times and stored in different relations.

The experiment was about errors in predicting the locations between conventional linear algorithm and the proposed algorithm using cubic spline interpolation polynomial. The results from the two different methods were compared.

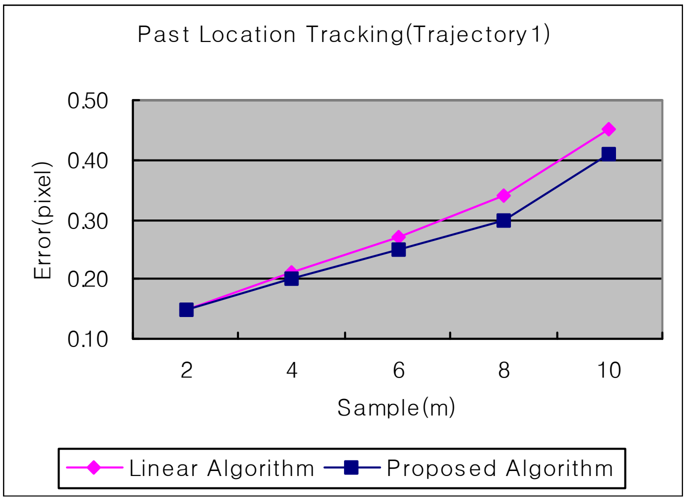


Figure 3. Error Comparison Graph of the Past Trajectory 1

Figure 3 compares the errors in past location tracking results of the samples at the regular intervals using linear algorithm and the proposed algorithm. The errors were calculated by subtracting the predicted location values from the actual values stored in the history relation information. Figure 3 shows that the errors by the linear algorithm are similar in the samples 2m. However, the errors by the proposed algorithm were smaller in average.

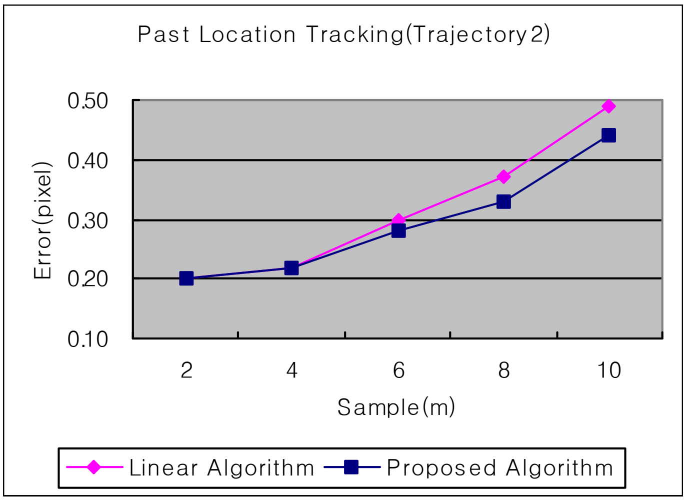


Figure 4. Error Comparison Graph of the Past Trajectory 2

Figure 4 shows the errors of the samples on trajectory 2 in finding the past location. On trajectory 2, the error by the linear function is similar in the approximately 2m samples. In most other samples, the errors by the proposed algorithm were smaller. From the experiment of finding the past location on trajectories 1 and 2, we conclude that the proposed algorithm using the piecewise cubic spline function method in this paper is better than the linear interpolation in finding the past location. Thus, for the comparatively curved route of this experiment, the estimation of the linear interpolation expressed in a straight line is less accurate than the cubic spline interpolation polynomial, which applies different polynomials to the piecewise.

In order to compare with the accuracy of the proposed algorithm, we use the linear algorithm. The time complexity of the linear algorithm is O(n2) and the proposed algorithm is O(n3). In this paper our experimentation focused on the accuracy of the location estimation results. Therefore we leave improvement in the processing time of the proposed algorithm as our future work.

4.2 Result of the Future Location Tracking

The moving route relation used in the future location estimation experiment is the same as the history information relation(HISTORY relation) of trajectories 1 and 2 used for the past location estimation. For the error analysis of estimation result using the future location tracking algorithm in chapter 3, and in the future location tracking algorithm, we assigned the values from 2 to 9 for the variable N. Random future time point used for the future location estimation was chosen randomly among the stored time values on the history relation. Moreover, we calculated the absolute error of the location values stored in the predicted location and history relation, and compared them with the number of the history objects.

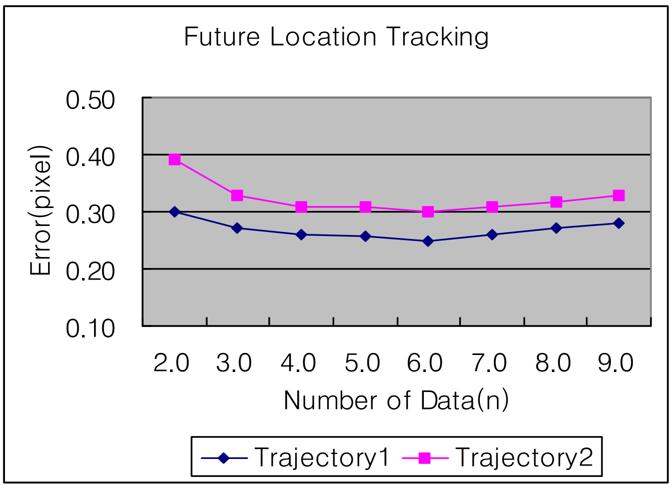


Figure 5. Error Comparison Graph of the Future Trajectory 1 and 2

Figure 5 shows the predicted result at future time points on the trajectories 1 and 2, in changing the number of history objects from 2 to 9. When the value of N is 6, the error between the predicted result and location stored in the history information relation is the least. The trajectories 1 and 2 show the least location estimation errors, when N is 6. If numbers of the history data are large, the location estimation result would be more accurate, whereas in moving routes such as in the trajectories 1 and 2, which are used in this experiment, when the value of N is bigger than 7, the produced errors are some larger. this result does not mean that the error is the least when the number of the history object is 6 in all the moving routes. Due to the characteristics of moving objects′ routes which are applied in various application fields, the value of N becomes changeable.

5. Conclusions

The uncertainty of moving objects causes problems in modeling database, processing the queries, indexing, and inaccuracies of replying to the queries. The inaccuracy in answering the queries may cause wrong decisions by the user.

This paper proposed location tracking algorithm to forecast both past and future uncertain locations of the moving objects. To find out the location at the past random time point which was not stored in the database, the piecewise cubic spline interpolation was used. In addition to estimate the location at future random time point, we used the average value of the past moving information. To analyze the characteristics of the proposed algorithms, location samples were taken from the sample trajectories and were stored in the history relation. We experimented with location estimation of the past and future using sample data.

From the experimentation, we found that the proposed location estimation algorithm using the piecewise cubic spline interpolation was in average smaller than the conventional linear interpolation algorithm. Also, we experimented the future location estimation as changing the number of the past history information. When the number of history information was 6, the location estimation errors were the least. However, the optimal value, the proper number of history information about future location estimation, can be changeable in accordance with the characteristics of the moving object applied in various application fields.

The suggested location tracking algorithm can be used in a home network service, a health care service, or a daycare service, etc., which are examples of various ubiquitous application services. In addition, the suggested algorithm can be used for regulating the output form of web content based on the characteristics of a user's connection device, the map service based on the user's location, bus route information service, a schedule notification service, a service of providing meeting subjects or related material, etc.

We will study that the proposed algorithm can supply application service by implement some location tracking application system. Also there will be a research on context modeling and location inference technique using the proposed algorithm for moving object management system.

References

1. Almeida, V. T. and Güting, R. H. (2005). Indexing the Trajectories of Moving Objects in Networks, *GeoInformatica* 9:1, 33-60.
2. Behr, T., Almeida, V. T. and Güting, R. H. (2006). Representation of periodic moving objects in databases, *14th ACM International Symposium on Geographic Information Systems*, ACM-GIS, 43-50.
3. Cao, H., Wolfson, O., Xu B. and Yin, H. (2005). MOBI-DIC: MOBIle DIscovery of loCal Resources in Peer-to-Peer Wireless Network, *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, Vol. 28, No. 3, 11-18.
4. Cruz, I., Khokhar, A., Liu, B., Sistla, A. P., Wolfson, O. and Yu, C. T. (2002). Research Activities in Database Management and Information Retrieval at the University of Illinois at Chicago, *SIGMOD Record*, Vol. 31(3), 103-108.
5. Ding, Z. and Güting, R. H. (2004 a). Managing Moving Objects on Dynamic Transportation Networks, *Proceedings of the 16th International Conference on Scientific and Statistical Database Management*(SSDBM, Santorini Island, Greece), 287-296.
6. Ding, Z. and Güting, R. H. (2004 b). Uncertainty Management for Network Constrained Moving Objects, Procceedings of the 15th *International Conference on Database and Expert Systems Applications*(DEXA, Zaragoza, Spain).
7. Forlizzi, L., Güting, R. H., Nardelli, E. and Schneider, M. (2000). A Data Model and Data Structures for Moving Objects Databases, *Proceedings of the ACM SIGMOD Conference*, 319-330.
8. Güting, R. H., Bohlen, M. H., Erwig, M., Jensen, C. S., Lorentzos, N. A., Schneider, M. and Varzirgiannis, M. (2000). A Foundation for Representing and Querying Moving Objects, *ACM Transactions on Database Systems*, Vol. 25, No. 1, 1-42.
9. Güting, R. H. (2007). Moving Object Languages, *Encyclopedia of Geographic Information Systems*, Springer-Verlag.
10. Ilarri, S., Wolfson, O., Mena, E., Illarramendi, A. and Rishe, N. (2006). Processing of Data Streams with Prediction Functions, *Proceedings of Hawaii International Conference on System Sciences* (HICSS-39), Kauai, Hawaii, 4-7.
11. Jeong, S. G., Lee, S. R. and Gweon, U. H. (2008). Numerical Value

Interpretation using CEM Tool, http://icat.snu.ac.kr:3000/.
12. Lema, J. A. C., Forlizzi, L., Güting, R. H., Nardelli, E. and Schneider, M. (2003). Algorithms for Moving Objects Databases, *The Computer Journal* 46:6, 680-712.
13. Pfoser, D. and Tryfona, N. (2000). Fuzziness and Uncertainty in Spatiotemporal Applications, CHOROCHRONOS *Technical Report* CH-00-04, http://www.dblab.ntua.gr/~choros/.
14. Trajcevski, G., Cao, H., Scheuermann, P., Wolfson, O. and D. Vaccaro (2006). On-Line Data Reduction and the Quality of History in Moving Objects Databases, *Proceedings of the MobiDE'06*, 5th ACM International Workshop on Data Engineering for Wireless and Mobile Access, Chicago, IL., USA, 19-26.
15. Trajcevski, G., Wolfson, O., Hinrichs, K. and Chamberlain, S. (2004). Managing Uncertainty in Moving Objects Databases, *ACM Transactions on Database Systems*(TODS), 29(3), 463-507.

[Received July 25, 2008, Revised August 20, 2008, Accepted August 23, 2008]