

Article

# Interpolation and Prediction of Spatiotemporal Data Based on XML Integrated with Grey Dynamic Model

Luyi Bai <sup>1,2,\*</sup>, Li Yan <sup>3,4</sup> and Zongmin Ma <sup>1,4</sup>

<sup>1</sup> College of Information Science & Engineering, Northeastern University, Shenyang 110819, China; mazongmin@ise.neu.edu.cn

<sup>2</sup> Qinhuangdao Branch Campus, Northeastern University, Qinhuangdao 066004, China

<sup>3</sup> School of Software, Northeastern University, Shenyang 110819, China; yanl@swc.neu.edu.cn

<sup>4</sup> College of Computer Science & Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China

\* Correspondence: baily@neuq.edu.cn; Tel.: +86-135-1335-0400

Academic Editors: Ozgun Akcay and Wolfgang Kainz

Received: 10 January 2017; Accepted: 5 April 2017; Published: 7 April 2017

**Abstract:** Interpolation and prediction of spatiotemporal data are integral components of many real-world applications. Thus, approaches of interpolating and predicting spatiotemporal data have been extensively investigated. Currently, the grey dynamic model has been used to enhance the performance of interpolating and predicting spatiotemporal data. Meanwhile, the Extensible Markup Language (XML) has unique characteristics of information representation and exchange. In this paper, we first couple the grey dynamic model with the spatiotemporal XML model. Based on a definition of the position part of the spatiotemporal XML model, we extract the corresponding position information of each time interval and propose an algorithm for constructing an AVL tree to store them. Then, we present the architecture of an interpolating and predicting process and investigate change operations in positions. On this basis, we present an algorithm for interpolation and prediction of spatiotemporal data based on XML integrated with the grey dynamic model. Experimental results demonstrate the performance advantages of the proposed approach.

**Keywords:** spatiotemporal Extensible Markup Language (XML); grey dynamic model; AVL tree; interpolation and prediction

## 1. Introduction

The management of spatiotemporal data has been intensively studied in recent years on account of the global spatiotemporal data increase [1–4]. Among various applications (e.g., flight control, land management, blood forecasting, traffic supervision), the objects whose positions change over time, such as airplanes, ships, clouds, and so on, significantly contribute to the study of spatiotemporal data.

Researchers have addressed a variety of approaches in modeling and querying the location of the moving spatiotemporal data both in relational databases [5–8] and spatiotemporal databases [9–11]. Early studies on the positions of spatiotemporal data assumed that the spatiotemporal data remain static in each time interval [12,13]. In addition, representing and querying the locations of moving spatiotemporal data as a linear function of time, which assigns a set of possible values rather than a unique value, was introduced in [14]. With respect to continuously moving regions, Tøssebro and Güting [15] provided algorithms to interpolate between two snapshots, transitioning from simple convex polygons to arbitrary polygons. Güting et al. [6] identified the  $k$  nearest neighbors of a set of moving object trajectories for any instant of time within the lifetime. However, the fixed value model and linear model generate too many errors on account of the limited sampling rate and indeterminacy of the snapshot when increasing indeterminacy is concerned [16]. Furthermore, the linear prediction

models have limits [17]. For example, the movement is not always linear; if it is linear, it is not usually known. If it is linear and known, it changes so fast that prediction using the movement parameter information is meaningless. Moreover, many spatiotemporal applications not only query the historical spatiotemporal data, but they also strive to retrieve the near-future evolution of spatiotemporal data. Although the fixed value model and moving objects spatiotemporal (MOST) model [18] can provide the near-future evolution of spatiotemporal data, they assume the movement functions are already known. Additionally, when information is limited, the originally sampled data will generate errors. Therefore, a new modeling technology of interpolation and prediction of spatiotemporal data is required to overcome these problems.

In recent years, the grey system theory [19] has developed rapidly, and its applications have extended to the spatiotemporal data field. The grey prediction model, GM (1, 1), is used to enhance the estimation performance because it can reduce the randomness inside the history query results sequence and it generates a holistic measure. The grey system theory is fairly appropriate for predicting spatiotemporal information because the accumulated generating operation is the most important characteristic for the grey system theory. Its purpose is to reduce the randomness of data.

However, for grey generation in snapshots, the model reduces the randomness in the snapshot data sequence. Moreover, the model can more precisely predict the near-future evolution of spatiotemporal data. In [16], the grey modeling approach was used to generate the discrete spatiotemporal model for two-dimensional indeterminate moving point objects. In [20], Bao introduced a histogram-based grey estimator for spatiotemporal selectivity estimation according to the continuity of movement. However, their study was based on the traditional database. Thus, interpolation and prediction of spatiotemporal data are relatively complex primarily because of the imposing of strict restrictions on structure [21] and data types [22] of spatiotemporal data. In such a case, interpolation and prediction of spatiotemporal data require a new efficient medium.

The next-generation language of the Internet, the Extensible Markup Language (XML) is playing an increasingly important role as a medium for integrating and exchanging data from different sources. Furthermore, XML significantly influences data management [23,24] (e.g., interpolation and prediction of spatiotemporal data) because the data can remain in a tree structure and a node or subtree can be considered metadata. Thus, interpolation and prediction of spatiotemporal data based on XML seems to have greater performance advantages than a traditional database. This can be explained by two factors. Firstly, the basic data structure in XML is the labeled ordered rooted tree [25], which can facilitate extraction of the corresponding position nodes for interpolating and predicting spatiotemporal data. Secondly, XML can handle time by adding labeled edges with time intervals as the time dimension [26], as opposed to the complex extensions of traditional models [22,27]. This handling of time can facilitate extraction of the corresponding temporal information. In that case, it makes interpolation and prediction of spatiotemporal data much easier because the desired interpolating or predicting of spatiotemporal data can be achieved by directly extracting the corresponding position nodes and corresponding temporal information. Accordingly, it is significant to lead XML into interpolation and prediction of spatiotemporal data.

To the best of our knowledge, there are no reports on interpolation and prediction of spatiotemporal data based on XML integrated with the grey dynamic model. In this paper, we present a novel approach to interpolating and predicting the positions of moving spatiotemporal data, as well as the first use of GM (1, 1) on the spatiotemporal XML model. According to the structure of the spatiotemporal XML model, we add development Functions along the  $x$ - and  $y$ -axis into the position part. Then, we extract the position information of each time interval and construct an AVL tree to store them. After discussing the architecture of the interpolating and predicting processes based on GM (1, 1), we investigate the change operations in the position containing the update, delete, and insert functions. Then, we provide an algorithm for interpolation and prediction based on XML integrated with GM (1, 1). Our experimental results validate our analysis and show a substantial performance improvement.

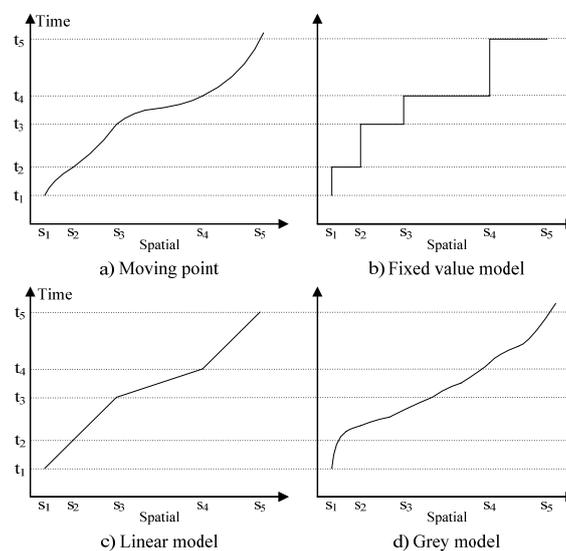
The remainder of this paper is organized as follows. In Section 2, we discuss basic information on types of interpolation and prediction models, the grey system theory, and GM (1, 1). We then couple GM (1, 1) with the spatiotemporal XML model in Section 3. Section 4 presents the architecture of interpolating and predicting processes, and Section 5 investigates the change operations in the position. In Section 6, we propose an algorithm for interpolation and prediction. Section 7 presents the experimental results, and Section 8 concludes the paper.

## 2. Basic Knowledge

### 2.1. Types of Interpolation and Prediction Model

In many areas of spatiotemporal databases, positions of spatiotemporal data are usually interpolated or predicted for both continuous and discrete changes because only finite position data are recorded in the database. Accordingly, some desired position information of spatiotemporal data should be interpolated or predicted. Continuous changes mean that changes occur throughout their entire lifespans, while discrete changes indicate that changes occur at specific time instants. For several phenomena (e.g., blood forecasting, forest fires, cloud waves, typhoon movements, etc.), we must track and monitor the respective spatiotemporal data to analyze or predict near-future movements.

We first introduce interpolation and prediction models of spatiotemporal data, which are mainly comprised of a fixed value model, linear model, and grey model. The fixed value model [13] assumes values of spatiotemporal data between snapshots to be constant, and it changes when a new snapshot is generated. Figure 1a shows the actual tracking of a moving point; Figure 1b represents it using the fixed value model. In Figure 1b, there are only five values in each time interval, and the values change when a new snapshot is generated. For instance, an individual waiting in a queue for tickets is found at a certain position, but the individual may change position from time to time (a new snapshot is generated).



**Figure 1.** Types of interpolation and prediction models (a–d).

The linear model [14] assumes that the development of spatiotemporal data during a time interval is a linear function. In each time interval, the beginning spatiotemporal data are stored as well as its development function. They are both used for interpolation or prediction when the spatiotemporal data are required at an arbitrary time point. In Figure 1c there are four lines to record the spatiotemporal data tracking (the trends of the first two lines are the same). The grey model [19] generates movement functions from a sequence of spatiotemporal data. Figure 1d simulates tracking of the moving point in Figure 1a. It can be observed that the approximation in Figure 1d is the closest one to the actual track.

## 2.2. Grey System Theory and GM (1, 1)

The grey system theory [19] was first presented in the late 1970s. Its main task is to seek the mathematic relations and movement rule among factors from the behavioral data of a system. The core of this theory is the dynamic grey model, whose characteristic is the generating function and the grey differential equation. The dynamic grey model uses differential fitting as a modeling method based on the concept of the grey generating function. Assuming the system's internal behavioral data is unknown, the single series first-order linear dynamic model GM (1, 1) of the grey system theory uses grey generations on the random external behavioral data to reduce randomness. It then establishes the grey difference equation for calculation and prediction. The arithmetic to generate GM (1, 1) from behavior data sequence  $X$  can be described as follows:

Let  $X$  be a non-negative behavioral data sequence:

$$X^{(0)} = \{x^{(0)}(1), x^{(0)}(2), x^{(0)}(3), x^{(0)}(4), \dots, x^{(0)}(n)\}, n > 3$$

Step 1: Use the accumulated generation operation on  $x^{(0)}$ ,

$$X^{(1)}(k) = \sum_{m=1}^k x^{(0)}(m), k = 2, 3, \dots, n$$

Step 2: Use the mean operation on  $x^{(1)}$ ,

$$Z^{(1)}(k) = 1/2[x^{(1)}(k-1) + x^{(1)}(k)], k = 2, 3, \dots, n$$

Step 3: Use the least squares difference approach to obtain coefficients  $a$  and  $b$ :

$$P = \begin{bmatrix} a \\ b \end{bmatrix} = [B^T B]^{-1} B^T y_n, B = \begin{bmatrix} -z^{(1)}(2) & -z^{(1)}(3) & \dots & -z^{(1)}(n) \\ 1 & 1 & \dots & 1 \end{bmatrix}^T$$

and  $y_n = [x^{(0)}(2), x^{(0)}(3), \dots, x^{(0)}(n)]^T$ .

The grey difference equation,  $x^{(0)}(k) + az^{(1)}(k) = b$ , is called the GM (1, 1) of sequence  $X$ . It can be used in near-future evolution prediction, even when the behavioral data sequence has a relatively high error rate or randomness [28]. In addition, it can predict data not recorded in database.

## 3. Coupling GM (1, 1) with Spatiotemporal XML Model

To interpolate and predict the positions of spatiotemporal data more accurately and efficiently, GM (1, 1) is integrated in the spatiotemporal XML model. In this section, we first illustrate the utilized spatiotemporal XML model. We then demonstrate how the spatiotemporal XML model is integrated with GM (1, 1) to interpolate and predict the positions of the spatiotemporal data.

An XML document is a labeled ordered rooted tree. We thus regard spatiotemporal data as a structured tree for metadata. As a result, operations between spatiotemporal data are actually operations between trees. According to the nature of spatiotemporal data, the definition of the spatiotemporal XML model is defined as follows.

**Definition 1** Spatiotemporal XML model  $XSP$  is a five-tuple,  $XSP = (OID, ATTR, P, M, T)$ , where

- $OID$  is the changing history of the spatiotemporal data.
- $ATTR$  denotes the attributes of the spatiotemporal data.
- $P$  is the position of the spatiotemporal data.
- $M$  is motion of the spatiotemporal data.
- $T$  is time of the spatiotemporal data.

Because GM (1, 1) uses grey generations to seek the mathematic relations and movement rule, the corresponding part of the spatiotemporal XML model only involves the position part. Accordingly, we mainly investigate how to use GM (1, 1) on the position part of the spatiotemporal XML model.

We consider the position part of the spatiotemporal XML model as a sub-data tree and provide its definition as follows.

**Definition 2** (Position part of the spatiotemporal XML model.) The position part of the spatiotemporal XML model  $XP$  is a five-tuple,  $XP = (x, y, f_x, f_y, T)$ , where

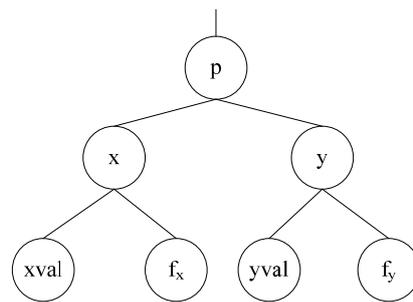
- $x$  and  $y$  are abscissa and vertical coordinates.
- $f_x$  and  $f_y$  are development functions along the  $x$ - and  $y$ -axis, respectively.
- $T$  is the time of the position.

Following the GM (1, 1) model, the position part of the spatiotemporal XML model can be expressed as a five-tuple. The development functions  $f_x$  and  $f_y$  can be generated from the GM (1, 1) model according to  $x$  and  $y$  in history records at each time  $T$ . Note that the position part of the spatiotemporal XML model defined above is used for zero-dimensional spatiotemporal data because it is the most common case. Furthermore, some one-dimensional or even two-dimensional spatiotemporal data can be considered zero-dimensional spatiotemporal data when the shape of the spatiotemporal data is irrelevant in terms of the whole situation.

The approach can be applied to one-dimensional or even two-dimensional spatiotemporal data as the development of functions. For instance, when we study an airplane route, we can consider the airplane as a point, even though it is actually two-dimensional spatiotemporal data. Another case is a typhoon. We are primarily concerned about the central part of the typhoon called the typhoon eye when its track is studied.

The position part fragment of the spatiotemporal XML model is shown in Figure 2. Node  $p$  is an abbreviation connoting that it is the position part of the spatiotemporal data node. For the case of  $x$  and  $y$  nodes, we not only add their value nodes, but we also add their development function nodes. Firstly, we use two child nodes of node  $p$ , denoted  $x$  and  $y$ , to identify the  $x$  node and  $y$  node, respectively. Then, for each child node of node  $p$ , the two child nodes are added to store the value and development functions. For the value nodes  $xval$  or  $yval$ , they are usually obtained by observation or even interpolation and prediction; for the development function nodes  $f_x$  and  $f_y$ , they are computed by GM (1, 1) using original values of  $x$  nodes and  $y$  nodes. Accordingly, the position node can store the position information ( $xval$  and  $yval$ ) as well as the development functions ( $f_x$  and  $f_y$ ). In that case, we can interpolate and predict the position of spatiotemporal data by reading the corresponding position values and the development functions. Although it may generate a great amount of redundancy since the development functions of  $x$  and  $y$  should be generated and stored at each transaction time interval, it saves the query time because we need only read the corresponding node values; it is not necessary to compute the development functions when querying. On the other hand, the so-called redundancy is actually useful and may be used when interpolating position nodes at a distant-past times.

Since the structure of the spatiotemporal XML model is not a regular tree, and the attribute nodes may be multiple, searching for the corresponding position information becomes complicated. To overcome this difficulty, we extract the position information in the spatiotemporal XML model and construct an AVL tree to store them. The advantage is mainly comprised of the following: (a) useful position information is extracted; (b) the data used in the experiment in the later section are relatively small and are suitable for using an AVL tree to store the extracted information; and (c) both the AVL tree and B-tree are similar in that they are data structures that cause the heights of their respective trees to be minimized according to their requirements. This “shortness” allows searching to be performed in  $O(\log n)$  time because the largest possible number of reads corresponds to the height of the tree.



**Figure 2.** Structure of the position fragment.

An AVL tree is a self-balancing binary search tree that is balanced to maintain  $O(\log n)$  height. A B- tree is a balanced tree, but it is not a binary tree. Nodes have more children. This increases the per-node search time, but it decreases the number of nodes the search must visit. We choose the AVL tree because its overall performance is better than that of the B- tree and B+ tree. In addition, inserting, deleting, updating, and querying nodes in the AVL tree are faster than those in the B- [29] tree and B+ [26] tree. Moreover, (d) techniques of creating, deleting, inserting, and searching nodes in the AVL tree have been studied and are applicable. Accordingly, we introduce an AVL tree to store the extracted position information.

In the algorithm *Construct\_AVL\_Tree*, we construct an AVL tree from the spatiotemporal XML model, extract the position information used, and store them in the constructed tree. Function *Grey\_Estimate* calculates the GM (1, 1) model from the given subsequence. Its algorithm is given in Section 7.

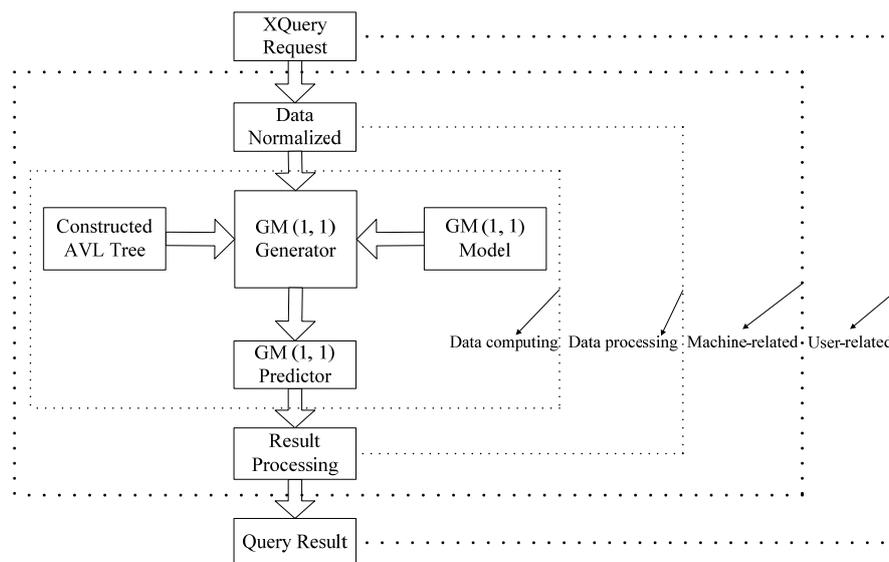
#### Algorithm *Construct\_AVL\_Tree* ( $x, y, f_x, f_y$ )

```

01 let t = null
02 for (m = i; m <= i; m--)
03 let ai = $o../position/x/xvali [@Ts = Ti]
04 let bi = $o../position/y/yvali [@Ts = Ti]
05 let ci = $o../position/x/fxi [@Ts = Ti]
06 let di = $o../position/y/fyi [@Ts = Ti]
07 Sequence (si) = add (ai, bi, ci, di)
08 SequenceB () = add (si)
09 end for
10 for each si in SequenceB () do
11 let x = Grey_Estimate (ai.x)
12 let y = Grey_Estimate (bi.y)
13 insert the current node (x, y, ci, di) to result tree t
14 end for
15 return t
  
```

#### 4. Interpolation and Prediction Process Architecture

The architecture of the interpolating and predicting process is based on the spatiotemporal XML model integrated with GM (1, 1), as illustrated in Figure 3. When there is an XQuery request, we should firstly normalize the corresponding data. Take the typhoon eye route as an example. The data recorded are the longitude and latitude coordinates for predicting the typhoon eye position. We should normalize the position data into coordinates along the  $x$ - and  $y$ -axis. Then, the GM (1, 1) generator is set up according to the GM (1, 1) model using a certain number of the most recent history data in the constructed AVL tree as the initial sequence for interpolating and predicting the query result at a given time.



**Figure 3.** Interpolating and predicting process architecture.

Furthermore, we should also process the result into the readable data since the corresponding generating process data are based on the spatiotemporal XML model and coordinate system. In that case, we should process the computed data into the exact position (longitude and latitude) so that they are readable. Finally, the query result can be returned. In the case of the interpolation or prediction of position of the spatiotemporal data, GM (1, 1) is actually used twice to generate  $x$ - and  $y$ -axis values, respectively.

The interpolating and predicting process architecture, as shown in Figure 3, mainly contains two parts: a user-related part and a machine-related part. The former provides two user interfaces, including the readable input and output data; the latter deals with the data and gives it the capability of satisfying user needs based on the spatiotemporal XML model integrated with GM (1, 1). The latter can be further divided into two parts: data processing and data computing. Data processing contains data normalization, which transforms the user-readable data into machine-readable data. The result processing reverses the normalized data. Data computation interpolates and predicts the desired result according to the GM (1, 1) model using a certain number of the most recent history data items in the constructed AVL tree as the initial sequence.

## 5. Position Change Operations

Spatiotemporal XML documents are subject to continuously changing operations, namely *update*, *delete*, and *insert*. These continuously changing operations may lead to different effects on elements and attributes in the position part of the spatiotemporal XML model. These change operations are related to interpolation or prediction of spatiotemporal data. Because the integrated grey dynamic model only deals with the position part of the spatiotemporal XML model, in this section we demonstrate how their effect on elements and attributes can be represented in the position part of the spatiotemporal XML model. The following is the effect of performing each operation on position elements and attributes.

**Update.** Updating the value of a position can be seen as replacement of the corresponding elements and attributes with other ones that have the same name but different content. Actually, not only the content of the position ( $x$  and  $y$ ), but also the temporal interval of the position and development functions ( $f_x$  and  $f_y$ ) should be updated. We discuss the transaction time and data recorded in the database. When position element  $p$  is updated at time  $t$ :

- (i) A new element with the same name  $p$  and all its descendants  $(x, y, f_x, f_y)$  are appended after the original ones.
- (ii) Compute  $f_x$  and  $f_y$  according to the original corresponding elements using GM (1, 1).
- (iii) The attributes  $T_s$  and  $T_e$  of position element  $p$  and all its descendants are set to  $[t, \text{Now}]$ .

**Delete.** Deleting the value of a position involves only deleting the position node and all its descendants and corresponding attributes. However, the deletion of the position node affects the development functions of later position nodes because the development functions are generated by later position nodes.

When position element  $p$  is deleted at time  $t$ :

- (i) For  $t \in [T_1, T_2] \wedge T_1 = \max(T_s) \wedge T_2 = \min(T_e)$ , delete position element  $p$  and all its descendants at time interval  $[T_1, T_2]$ .
- (ii) For position element  $p$  where  $// p [ @T_s > @T_2 ]$ , again compute  $f_x$  and  $f_y$  of the position element nodes after position element node  $p$  according to fresh original corresponding elements using GM (1, 1), respectively.

**Insert.** Inserting the value of a position can be considered insertion of the corresponding elements and attributes. The position element whose attribute time is involved in the insertion should also be updated. In addition, the insertion of the position node affects the development functions of later position nodes because the development functions are generated by later position nodes. In that case, these development functions should be updated.

When position element  $p$  is inserted at time  $t$ :

- (i) For the position element  $p$  whose time interval contains  $t$  ( $t \in [T_s, T_e]$ ), the inserted element with the same name  $p$  and all its descendants  $(x, y, f_x, f_y)$  are appended after this position  $p$ .
- (ii) The time attributes of the inserted element and all its descendants are set to  $[t, T_e]$ .
- (iii) The time attributes of position element  $p$  and all its descendants are set to  $[T_s, t - 1]$ .
- (iv) For the position elements after this position element  $p$ , again compute  $f_x$  and  $f_y$  according to the fresh original corresponding elements using GM (1, 1), respectively.

## 6. Interpolations and Predictions

In this section, we propose an approach to interpolating and predicting the positions of spatiotemporal data. Following the definition of the position part of the spatiotemporal XML model, the position of spatiotemporal data can be expressed as  $(x_1, y_1, f_{x1}, f_{y1}, T_1), (x_2, y_2, f_{x2}, f_{y2}, T_2), \dots, (x_i, y_i, f_{xi}, f_{yi}, T_i), \dots, (x_n, y_n, f_{xn}, f_{yn}, T_n)$ , where  $x_i$  and  $y_i$  are coordinates along the  $x$ - and  $y$ -axis, and  $T_i$  is the time of the position as  $(x_i, y_i)$ . The development functions,  $f_{xi}$  and  $f_{yi}$ , are the GM (1, 1) model generated from  $x_i$  and  $y_i$ , respectively.

Because the position contains two variables, it can be resolved into  $(x_1, f_{x1}, T_1), (x_2, f_{x2}, T_2), \dots, (x_i, f_{xi}, T_i), \dots, (x_n, f_{xn}, T_n)$  and  $(y_1, f_{y1}, T_1), (y_2, f_{y2}, T_2), \dots, (y_i, f_{yi}, T_i), \dots, (y_n, f_{yn}, T_n)$ . When the position information at time  $t$  is needed, simply find  $i$ , which makes  $t \in [T_{i-1}, T_i]$ . Then, the interpolation is obtained as initial value  $(x_{i-1}) + f_x(t)$  and initial value  $(y_{i-1}) + f_y(t)$ . The formula can also be used for the near-future prediction of the positions of spatiotemporal data. However, the prediction time length should be limited in  $[T_m, T_m + r]$ , where  $r \leq \min(T_i - T_{i-1})$ . The algorithm for interpolation and prediction in the spatiotemporal XML model using GM (1, 1) is described below.

**Algorithm Grey\_Estimate (o)**

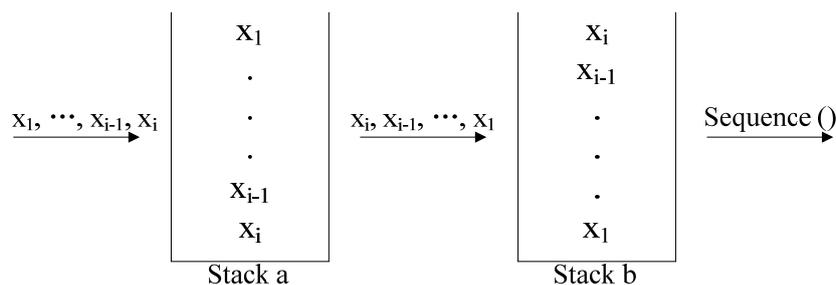
```

01 for (m = i; m <= i; m--)
02 let ai = $o/./position/x/xvali [@Ts = Ti]
03 let bi = $o/./position/y/yvali [@Ts = Ti]
04 Sequencex () = add (ai)
05 Sequencey () = add(bi)
06 end for
07 if Length (Sequencex > 3) and Length (Sequencey > 3)
08   Coefficientx = Calculate_GM_Coefficient (Sequencex)
09   Coefficienty = Calculate_GM_Coefficient (Sequencey)
10   Resultx = Estimate_at (Coefficientx, Sequencex , Ti)
11   Resulty = Estimate_at (Coefficienty, Sequencey , Ti)
12 end if
13 return Resultx
14 return Resulty

```

According to the algorithm *Grey\_Estimate*, we can interpolate or predict the position of spatiotemporal data. The difference of interpolation and prediction is whether the time queried is between the time interval of recorded spatiotemporal data or after the maximum time point of recorded spatiotemporal data. In the process of interpolation or prediction, we should obtain the latest data of the position information. To employ the GM (1, 1) model, the latest data of the position should be more than three. The functions *Calculate\_GM\_Coefficient* and *Estimate\_at* generate the GM (1, 1) model from the sequence data and interpolate or predict the required value. Their algorithms can be designed from the arithmetic presented in Section 2; they are not addressed here on account of space constraints.

When we use the GM (1, 1) model, the latest data of the position information are required. According to definition 2, the latest data of *Sequence ()* are obtained in reverse order (from recent time to distant-past time). In addition, the number of data in *Sequence ()* is random according to actual needs and must be more than three. In other words, it is difficult to acquire the latest required data since the latest data are computed in reverse order. Consequently, we use two stacks to process the latest data so they are easily obtained, as shown in Figure 4.



**Figure 4.** Stacks for processing the most current data.

Firstly, the required latest data produce a data stream  $(x_1, \dots, x_{i-1}, x_i)$ . Stack *a* processes the data stream. Hence, the recent data are located at the bottom of stack *a* and the distant-past data are located at the top of stack *a*. Then, these data are extracted so that the data stream is transformed into  $(x_i, x_{i-1}, \dots, x_1)$ . Finally, stack *b* processes the extracted data stream so that the distant-past data are located at the bottom of the stack, and the recent data are located at the top of the stack. After processing the two stacks, the latest data can be easily added to *Sequence ()*, and it can guarantee that the latest data in *Sequence ()* are ordered from the most recent time to the distant-past time.

## 7. Experiments

To evaluate the effectiveness of interpolation and prediction of spatiotemporal data based on XML integrated with the grey dynamic model, we evaluated the feasibility and accurateness using GM (1, 1). We compared its performance with that of GM (1, 1) based on MATLAB. The experiments were implemented in MATLAB 7.0.1 and Microsoft Visual C++ 6.0, and they were performed on a system with a 1.8-GHz Pentium IV processor, 1 GB RAM, and the Windows XP operating system.

### 7.1. Experimental Setup

**Data set:** The data set, shown in Figure 5, was from tropical storm data [30]. The left part of Figure 5 shows the track of the Muifa storm; the right part of Figure 5 displays data information of the Muifa storm containing the name, date, time (UT), latitude (degrees), longitude (degrees), course (true), speed (knots), pressure (mb), and wind (knots). Because interpolation and prediction of spatiotemporal data only addresses the position part, we extracted the latitude and longitude information of each time from the data set, which consisted of 60 data items. In the case of the generated AVL tree, there were 60 nodes, and the depth of the AVL tree was six. In other words, the generated AVL tree had six levels.

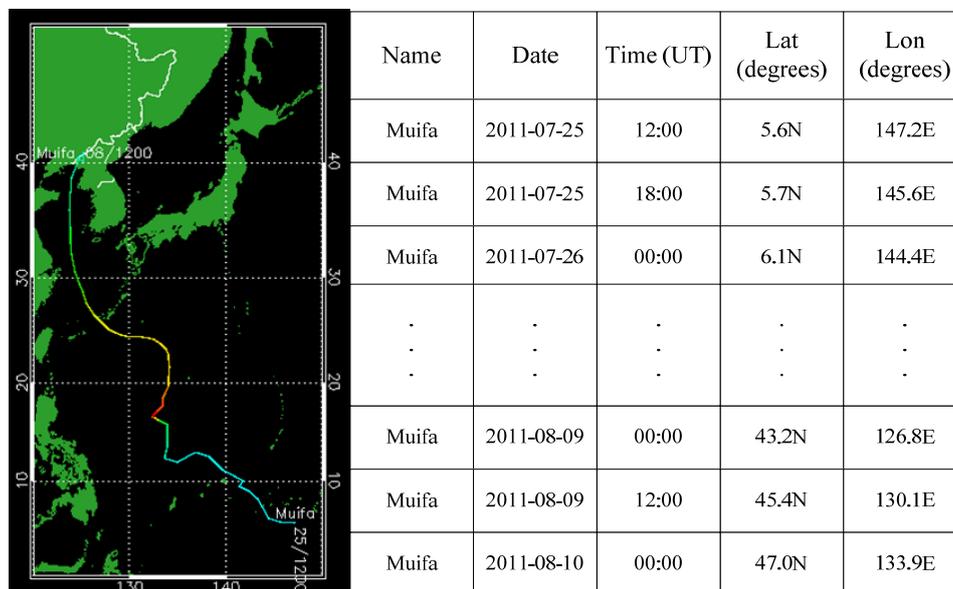


Figure 5. Data of Muifa storm tracking.

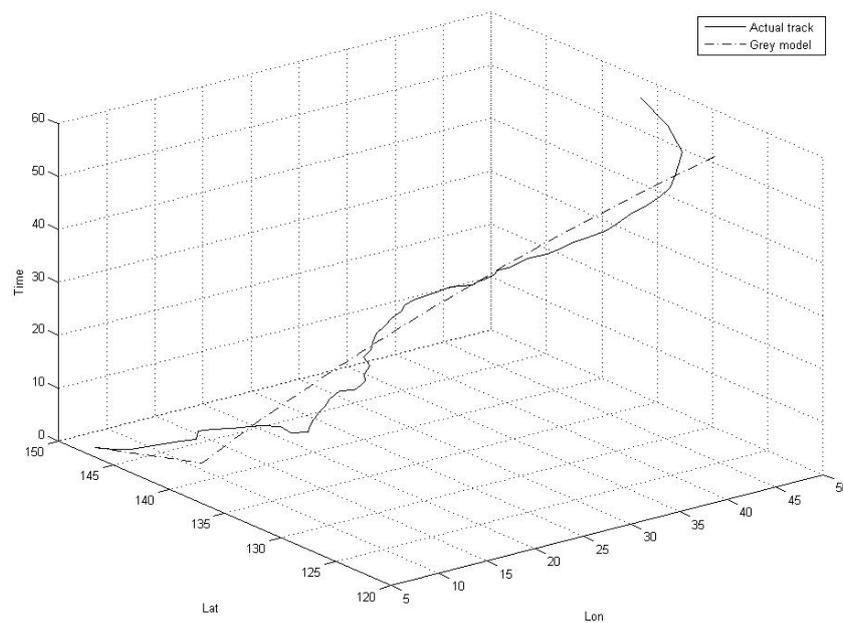
**Experimental environment:** The experimental environment was comprised of MATLAB 7.0.1 and Microsoft Visual C++ 6.0. In the MATLAB environment, we evaluated the effectiveness of the interpolation and prediction of storm tracking based on XML integrated with a grey dynamic model with the actual track. We considered the 60 time instants of the actual time as isomorphic to a natural number in the experiment. For a performance comparison, both experimental environments were used. Searching for the corresponding node in the AVL tree and reading its contents were evaluated in Microsoft Visual C++ 6.0, while generating the development function and computing the values of  $x$  and  $y$  were evaluated in MATLAB.

### 7.2. Experimental Results

We compared the interpolation and prediction of spatiotemporal data based on XML integrated with the grey dynamic model to the interpolation and prediction in a normal environment over the above data set. To analyze the effectiveness of our proposed approach, we performed three groups of experiments. The first two experiments were performed to demonstrate the feasibility and accurateness

of the proposed approach using GM (1, 1), and the last experiment was performed to demonstrate the performance advantages of our approach.

In the first group of experiments, we performed interpolation and prediction of Muifa storm tracking based on the spatiotemporal XML model integrated with GM (1, 1). Figure 6 shows the comparison between the actual track of the Muifa storm and the fitting storm tracked by the GM (1, 1) model. The solid curve represents the actual track of the Muifa storm; the dashed curve represents the fitting storm tracked by the GM (1, 1) model. As shown in Figure 6, a considerable deviation of the fitting storm track from the actual Muifa storm track is apparent. The deviation increases because all 60 data items are fitted in one time; thus, the deviation of the fitting track increases when the track tendency varies. However, the fitting track by GM (1, 1) still holds the holistic moving tendency and remains stable.



**Figure 6.** Storm track with a high indeterminacy.

In the second group of experiments, we improved the fitting track by the GM (1, 1) model to reduce the deviation, as Figure 7 shows. As depicted in the figure, the solid curve represents the actual track of the storm; the dashed curve represents the fitting storm tracked by the GM (1, 1) model. The approach we used in this experiment involved dividing the 60 data items into 12 groups and generating the corresponding storm track, respectively. The improvement enabled the storm tracking to flexibly adjust to the actual storm track. In that case, Figure 6 shows the storm track with high indeterminacy; Figure 7 shows the storm track with low indeterminacy. It is observed that the latter performs better than the former.

Finally, in the third group of experiments, we compared the performances. To this end, we evaluated the execution time and memory cost of interpolating and predicting the Muifa storm tracking based on XML integrated with the grey dynamic model (noted as GXML for simplicity), as well as the execution time of that in the relational databases (here we used MATLAB for the evaluation), as shown in Figures 8 and 9. Figure 8 shows the performance comparison between the execution times computed with GXML and MATLAB. We observe that the execution time of GXML increases slightly more than that in MATLAB in accordance with the increasing number of data items. The execution time of GXML is longer than that of MATLAB when the number of data items is small (less than approximately five data items), and the GXML execution time is less than that of MATLAB with an obvious increase in the difference over time. This can be explained by the fact that GXML first needs structural transforming; then, the position information can be directly extracted.

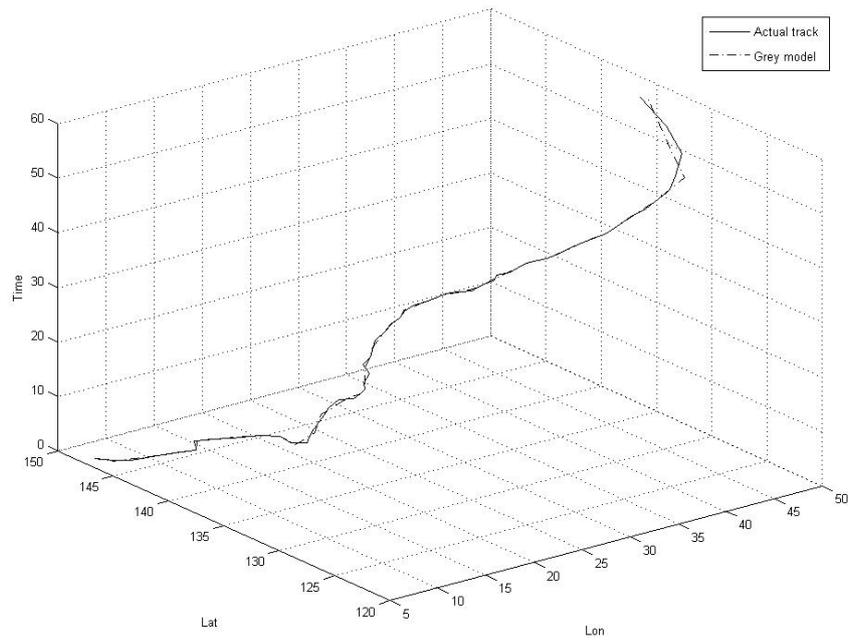


Figure 7. Storm track with low indeterminacy.

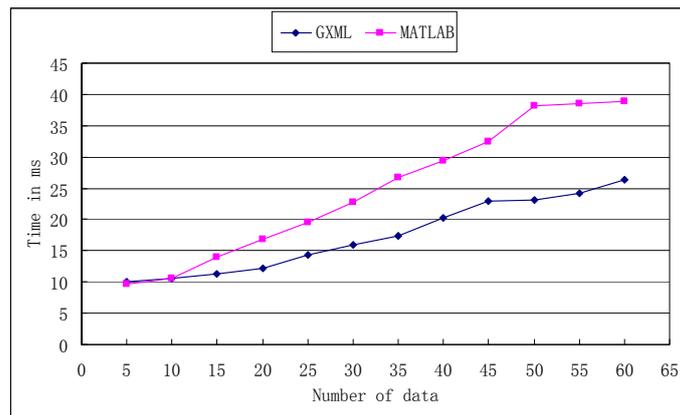


Figure 8. Execution time comparison.

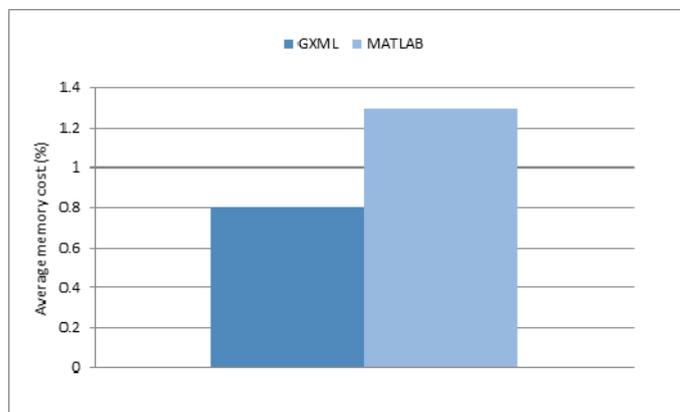


Figure 9. Memory cost comparison.

Figure 9 shows a memory cost comparison between GXML and MATLAB. It is observed that the memory cost of GXML is smaller than that of MATLAB. This can be explained by the fact that the memory costs not only depend on the cost size of the dataset, but they also depend on the complexity of the approach. In our approach, after structural transforming, only the position nodes of the structural model should be considered.

## 8. Conclusions

To effectively process the interpolation and prediction of spatiotemporal data, a novel approach for interpolation and prediction of spatiotemporal data based on the spatiotemporal XML model integrated with the grey dynamic model was introduced in this paper. We extracted the corresponding position information of each time interval from the position part of the proposed model. We then constructed an AVL tree to store them. After discussing change operations, we additionally presented an algorithm for interpolation and prediction based on the spatiotemporal XML model integrated with GM (1, 1). The experimental results showed that our approach efficiently performed interpolation and prediction of spatiotemporal data.

Interpolation and prediction of spatiotemporal data based on XML with the grey dynamic model require further research. In this regard, most spatiotemporal data should be studied. Another future research topic is to extend our studies on the basis of other formats, such as JSON and YAML.

**Acknowledgments:** This work was supported by the National Natural Science Foundation of China (61402087, 61370075), the Natural Science Foundation of Hebei Province (F2015501049), the Scientific Research Fund of Hebei Education Department (QN2014339), and in part by the Program for New Century Excellent Talents in University (NCET-05-0288).

**Author Contributions:** In this paper, Luyi Bai and Z.M. Ma conceived the idea and continuously advised during the design phase. Luyi Bai and Li Yan designed the algorithms, performed measurements, and wrote the paper.

**Conflicts of Interest:** The authors declare no conflicts of interest. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; and in the decision to publish the results.

## References

1. Blekas, K.; Nikou, C.; Galatsanos, N.; Tsekos, N.V. A regression mixture model with spatial constraints for clustering spatiotemporal data. *Int. J. Artif. Intell. Tools* **2008**, *17*, 1023–1041. [[CrossRef](#)]
2. Jadidi, A.; Mostafavi, M.A.; Bédard, Y.; Shahriari, K. Spatial representation of coastal risk: A fuzzy approach to deal with uncertainty. *ISPRS Int. J. Geo-Inf.* **2014**, *3*, 1077–1100. [[CrossRef](#)]
3. Pelekis, N.; Theodoulidis, B.; Kopanakis, I.; Theodoridis, Y. Literature review of spatio-temporal database models. *Knowl. Eng. Rev.* **2004**, *19*, 235–274. [[CrossRef](#)]
4. Hallot, P.; Billen, R. Enhancing spatio-temporal identity: States of existence and presence. *ISPRS Int. J. Geo-Inf.* **2016**, *5*, 62. [[CrossRef](#)]
5. Dobra, A.; Garofalakis, M.; Gehrke, J.; Rastogi, R. Sketch-based multi-query processing over data streams. *Lect. Notes Comput. Sci.* **2004**, *2992*, 621–622.
6. Güting, R.H.; Behr, T.; Xu, J. Efficient k-nearest neighbor search on moving object trajectories. *Int. J. VLDB* **2010**, *19*, 687–714. [[CrossRef](#)]
7. Alamri, S.; Taniar, D.; Safar, M. A taxonomy for moving object queries in spatial databases. *Future Gener. Comp. Syst.* **2014**, *37*, 232–242. [[CrossRef](#)]
8. Wu, Y.; Agrawal, D.; Abbadi, A. Applying the golden rule of sampling for query estimation. In Proceedings of the ACM SIGMOD International Conference on Management of Data, Santa Barbara, CA, USA, 21–24 May 2001.
9. Choi, Y.; Chung, C. Selectivity estimation for spatio-temporal queries to moving objects. In Proceedings of the SIGMOD Conference, Madison, WI, USA, 3–6 June 2002.
10. Hadjieleftheriou, M.; Kollios, G.; Tsotras, V. Performance evaluation of spatio-temporal selectivity estimation techniques. In Proceedings of the 15th International Conference on Scientific and Statistical Database Management, Cambridge, MA, USA, 9–11 July 2003.

11. Tao, Y.; Kollios, G.; Considine, J.; Li, F.; Papadias, D. Spatio-temporal aggregation using sketches. In Proceedings of the 20th International Conference on Data Engineering, Boston, MA, USA, 30 March–2 April 2004.
12. Langran, G. A review of temporal database research and its use in GIS applications. *Int. J. Geogr. Inf. Syst.* **1989**, *3*, 215–232. [[CrossRef](#)]
13. Segev, A.; Shoshani, A. Logical modeling of temporal data. In Proceedings of the ACM SIGMOD International Conference on Management of Data, San Francisco, CA, USA, 27–29 May 1987.
14. Sistla, A.P.; Wolfson, O.; Chamberlain, S.; Dao, S. Modeling and querying moving objects. In Proceedings of the 13th International Conference on Data Engineering, Birmingham, UK, 7–11 April 1997.
15. Tøssebro, E.; Güting, R.H. Creating representations for continuously moving regions from observations. In Proceedings of the 7th International Symposium on Spatial and Temporal Databases, Redondo Beach, CA, USA, 12–15 July 2001.
16. Bao, L.; Qin, X. The discrete representation of continuously moving indeterminate objects. *Chin. J. Aeronaut.* **2005**, *18*, 59–64. [[CrossRef](#)]
17. Sun, J.; Papadias, D.; Tao, Y.; Liu, B. Querying about the past, the present, and the future in spatio-temporal databases. In Proceedings of the 20th International Conference on Data Engineering, Boston, MA, USA, 30 March–2 April 2004.
18. Wolfson, O. Querying the uncertain position of moving objects. *Lect. Notes Comput. Sci.* **1998**, *1399*, 310–337.
19. Kayacan, E.; Ulutas, B.; Kaynak, O. Grey system theory-based models in time series prediction. *Expert Syst. Appl.* **2010**, *37*, 1784–1789. [[CrossRef](#)]
20. Bao, L.; Zhou, M.; Li, Q. A Histogram-based grey estimator for spatiotemporal selective queries. In Proceedings of the 5th International Conference on Machine Learning and Cybernetics, Dalian, China, 13–16 August 2006.
21. Pfoser, D.; Tryfona, N. Requirements, definitions, and notations for spatiotemporal application environments. In Proceedings of the 6th ACM Symposium on GIS, Washington, DC, USA, 6–7 November 1998.
22. Claramunt, C.; Theriault, M. Managing time in GIS: An event-oriented approach. In Proceedings of the International Workshop on Temporal Databases, Zürich, Switzerland, 17–18 September 1995.
23. Algergawy, A.; Mesiti, M.; Nayak, R.; Saake, G. XML data clustering: An overview. *ACM Comp. Surv.* **2011**, *43*, 1–41. [[CrossRef](#)]
24. Guo, L. Pioneering GML deployment for NSDI—Case study of USTIGER/GML. *ISPRS Int. J. Geo-Inf.* **2013**, *2*, 82–93. [[CrossRef](#)]
25. Ma, Z.M.; Liu, J.; Yan, L. Fuzzy data modeling and algebraic operations in XML. *Int. J. Intell. Syst.* **2010**, *25*, 925–947. [[CrossRef](#)]
26. Rizzolo, F.; Vaisman, A.A. Temporal XML: Modeling, indexing, and query processing. *Int. J. VLDB* **2008**, *17*, 1179–1212. [[CrossRef](#)]
27. Peuquet, D.; Duan, N. An event-based spatiotemporal data model for temporal analysis of geographical data. *Int. J. Geogr. Inf. Syst.* **1995**, *9*, 7–24. [[CrossRef](#)]
28. Liu, S.; Lin, Y. *An Introduction to Grey Systems: Foundations, Methodology and Applications*; Emerald Publishing Limited: Bingley, UK, 1998.
29. Belbaraka, M.; Stojmenović, I. On generating B-trees with constant average delay and in lexicographic order. *Inform. Process. Lett.* **1994**, *49*, 27–32. [[CrossRef](#)]
30. Mufia Data. Available online: [http://rammb.cira.colostate.edu/products/tc\\_realtime/storm.asp?storm\\_identificier=WP112011](http://rammb.cira.colostate.edu/products/tc_realtime/storm.asp?storm_identificier=WP112011) (accessed on 17 September 2012).

