

A Spatiotemporal Database Prototype for Managing Volumetric Surface Movement Data in Virtual GIS

Mohd Shafry Mohd Rahim¹, Abdul Rashid Mohamed Shariff²,
Shattri Mansor², Ahmad Rodzi Mahmud², and Daut Daman¹

¹ Faculty of Computer Science and Information System,
University Technology Malaysia, 81310 Skudai, Johor, Malaysia

² Institute of Advance Technology, University Putra Malaysia,
43400 Serdang, Selangor, Malaysia

shafry@utm.my, rashid@eng.upm.edu.my
shattri@eng.upm.edu.my, arm@eng.upm.edu.my, daut@utm.my

Abstract. Virtual GIS have the capability to present geographic features in three-dimension. This will contribute to increasing user understanding of phenomena that occurred in certain areas. Virtual GIS need to include temporal element in the system for managing dynamic phenomena to simulate real process. Spatiotemporal database system can be used to manage temporal data or spatial data that involve changes. In this paper, we present the database system of a spatiotemporal database prototype for managing volumetric surface movement; Volumetric Surface Movement Spatiotemporal Data Model that has the ability to store changes of the surface data of a three dimensional object. Formalization of the surface movement phenomena on the volumetric object were defined. This model has been implemented using relational database model which store only the changing points in order to ensure an efficient data management. We have developed a visualization tool with the database system to visualize movement on the surface. The prototype system of this model has been tested and found to be efficient.

Keywords: Spatiotemporal Data Model, Database, Surface Movement, Temporal GIS, Virtual GIS.

1 Introduction

Virtual Geographical Information Systems (VGIS) is generally defined as the use of computer graphic technology to improve the presentation of geographical information. Therefore, Geographical Information Systems (GIS) will be able to become more realistic with real process and presentation precisely as in the real world. In many cases, simply introducing an additional orthogonal axis (Z) is convenient but undeniably insufficient. This is because important spatial and temporal characteristics and relationships may be imperceptible in this approach. Although visualization techniques for three or more dimensions have become popular in recent years, data models and formal languages have not yet

fully developed to support advanced spatial and temporal analysis in multiple dimensions [1].

In Virtual Geographical Information Systems (VGIS), data model is the abstraction of the real world phenomena according to a formalized, conceptual scheme, which is usually, implemented using the geographical primitives of points, lines, and polygons or discredited continuous fields [2]. Data model should define data types, relationships, operations, and rules in order to maintain database integrity [2]. In VGIS, data model is also used to enhance the focus on 3D data. Thus, spatiotemporal data model in VGIS is an abstraction of managing 3D spatial with temporal elements. Spatiotemporal data model is very important in creating a good database system for VGIS which deals with space and time as main factors in the system [3,15].

A variety of spatiotemporal data model were developed previously and for the purpose of this research, we have collected and analyzed 9 data models namely, GEN-STGIS [5], Cell Tuple-based Spatiotemporal Data Model [6], Cube Data Model [7], Activity-based Data Model [8], Object-based Data Model, Data Model for Zoning [9], Object Oriented Spatial Temporal Data Model [10], Multigranular Spatiotemporal Data Model [11] and Feature-Based Temporal Data Model [12].

We agree that in order to create a VGIS with a realistic process, an appealing spatiotemporal data model should focus on the volumetric data and geographical movement behavior. Based on our analysis, an issue which is related to the capability of spatiotemporal data model is the 3D visualization of volumetric spatiotemporal object. This capability is vital in order to increase user understanding in the geographic phenomena so to create simulations or future predictions. Thus, data model in VGIS must have the capability of user query to visualize information in the form of 3D with movements. This is indeed a very challenging issue. It has also been addressed by [13] that the current development of techniques and tools are simply unable to cope when expanded to handle additional dimensions. In this paper, we will discuss implementation of our proposed data model which is the Volumetric Surface Movement Spatiotemporal Data Model. This model was developed to manage movement of the volumetric object with less data redundancy, and also to retrieve data for visualization. Our discussion will focus on the system architecture in general which includes database development, data retrieval and visualization approach.

2 System Architecture

Figure 1 shows the architecture of our prototype system which consists of *Data Storage*, *Spatiotemporal Data Loader Processor*, *Spatiotemporal Data Retrieval Processor* and *Volumetric Surface Movement Simulator*.

Data Storage stores the volumetric surface movement object according to the database scheme. The database that were used in this system is MySQL which is open source. *Spatiotemporal Data Loader Processor* provide methods of loading data into the database. This processor will ensure no data redundancy in the database by storing only the points which were involved during the surface movement.

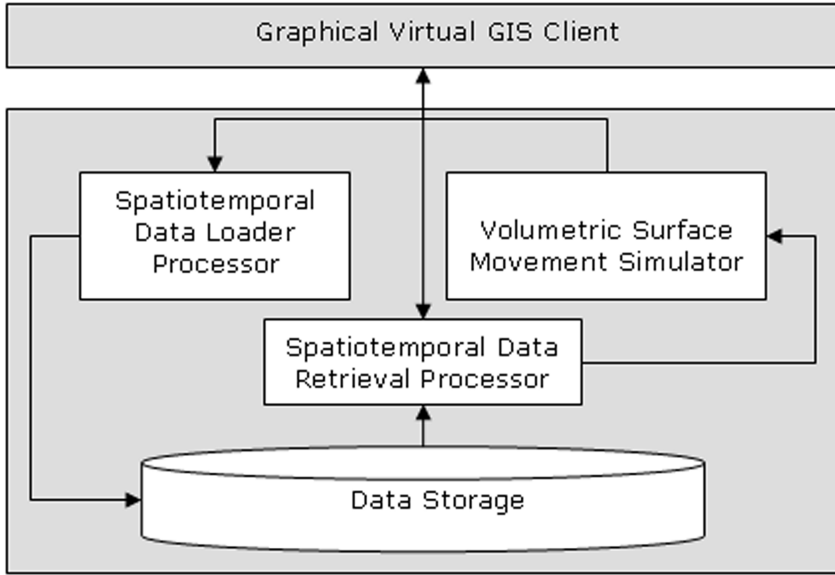


Fig. 1. Architecture of the system

Spatiotemporal Data Retrieval Processor will retrieve the data from the database based on what the user requires. In this case, the user need to input the area and duration of data. Data that were retrieved will be transferred into our own file format. *Volumetric Surface Movement Simulator* will read the retrieved data set and simulates the volumetric surface movement object. *Graphical Virtual GIS Client* accepts users queries, and sends it to the *Spatiotemporal Data Retrieval Processor*, and illustrate the query result in graphical presentation.

3 Volumetric Surface Movements Spatiotemporal Data Model

Based on the volumetric surface movement formalization by [4] we can see that conceptually the data that were used to define the surface of a volumetric object are actually the points with the value of x , y , and z . This refers to the location and the coordinate of the points. Whenever there is a change on the surface, these points also change. Therefore time is the utmost important component in efficiently managing the points that forms the surface.

The model is based on the object representation (1) and (2) below which relates to the data that forms the surface and are stored and extracted using the equation of $f(mv)t_1, t_2, \dots, t_m$ where it is a combination of the surface set. The combination from the surface set is classified as a volumetric surface movement object. This will be the base in building a data model to manage data that involves surface movements and the foundation in managing volumetric surface movement object in the real world.

$$f(mv)t_1, t_2, \dots, t_m \rightarrow f(v, t_1) \cup f(v, t_2) \cup \dots \cup f(v, t_m) \tag{1}$$

$$\begin{aligned} f(mv)t_1, t_2, \dots, t_m \rightarrow [\{ \langle x_1, y_1, z_1, t_1 \rangle, \langle x_2, y_2, z_2, t_1 \rangle, \dots \langle x_n, y_n, z_n, t_1 \rangle, \} \\ \cup \{ \langle x_1, y_1, z_1, t_2 \rangle, \langle x_2, y_2, z_2, t_2 \rangle, \dots \langle x_n, y_n, z_n, t_2 \rangle, \} \\ \cup \dots \cup \{ \langle x_1, y_1, z_1, t_m \rangle, \langle x_2, y_2, z_2, t_m \rangle, \dots \langle x_n, y_n, z_n, t_m \rangle, \}] \end{aligned} \tag{2}$$

In real process, not all of the points in the volumetric surface moves or changes. This raises the question as to whether it is necessary to store all of the points, which will increase the storage usage in the implementation. Therefore, in order to avoid data redundancy, the data model must be able to identify which point that has changed. To identify, data model must have the capability to check every point among version of data and capture the changing point. To perform this task, the formalism that was defined earlier will be used. The conceptual identification is as follows:

- Assume Point₁ is $\langle x_1, y_1, z_1 \rangle$ at time t_n*
- Point₂ is Point₁ at time t_{n+1}*
- If Point₂ - Point₁ = 0 then Point₂ = Point₁ at time t_{n+1}*
- If Point₂ - Point₁ \neq 0 then Point₂ = Point₂ at time t_{n+1}*

Visually, our proposed data model can be translated in table form to give a clearer understanding. Table 1 below describes the proposed data model.

Table 1. Volumetric Surface Movement Data Model

v/t	P ₁	P ₂	P ₃
t_n	(x_1, y_1, z_1)	(x_2, y_2, z_2)	(x_3, y_3, z_3)
t_{n+1}	$(x_1, y_1, z_1) \parallel (x_1, y_1, z_1)'$	$(x_1, y_1, z_1) \parallel (x_1, y_1, z_1)'$	$(x_1, y_1, z_1) \parallel (x_1, y_1, z_1)'$
.	.	.	.
.	.	.	.
t_{n+m}	$(x_1, y_1, z_1)'' \parallel (x_1, y_1, z_1)'''$	$(x_1, y_1, z_1)'' \parallel (x_1, y_1, z_1)'''$	$(x_1, y_1, z_1)'' \parallel (x_1, y_1, z_1)'''$

Each point will go through changes but if there is a change that occurs at time t, not all of the points that form the surface changes. Therefore to store data at time t, a comparison will be done to determine which points actually changes. Thus, the process identification as stated in algorithm above is used. The process is as follows:

$$\begin{aligned} P1(x_1, y_1, z_1, t_n) \text{ move} \rightarrow P1'(x_1, y_1, z_1, t_{n+1}) \text{ move} \rightarrow \dots \rightarrow P1'(x_1, y_1, z_1, t_{n+m}) \\ \text{If } (x_1, y_1, z_1, t_{n+1}) - (x_1, y_1, z_1, t_n) = 0 \\ \quad (x_1, y_1, z_1, t_{n+1}) = (x_1, y_1, z_1, t_n) \\ \text{Else if } ((x_1, y_1, z_1, t_{n+1}) - (x_1, y_1, z_1, t_n) > 0) \parallel \\ \quad ((x_1, y_1, z_1, t_{n+1}) - (x_1, y_1, z_1, t_n) < 0) \\ \quad (x_1, y_1, z_1, t_{n+1}) = (x_1, y_1, z_1, t_{n+1}) \end{aligned}$$

Therefore, the next point will be stored in the storage for every point after movement occurs

$$\text{from } t_n \text{ until } t_{n+1} \rightarrow (x_1, y_1, z_1) \parallel (x_1, y_1, z_1)'$$

By executing this process, movement data can be stored more easily while avoiding data redundancy. Hence, changes on a volumetric surface were managed more efficiently.

4 Database System Development

In Volumetric Surface Movement Spatiotemporal Data Model or VSMST Data Model, surface on the volumetric object are represented on a 3D space together with the temporal element which represents the changes or movement. To specify their position in space, this data model uses Cartesian frame for reference. This frame is chosen based on the successful development of the database system in GEN-STGIS by [5]. More specifically, VSMST Data Model uses 3 dimensional Cartesian (x, y, z) coordinate system and it is based on the Euclidean geometry, which concerns the study of points, lines, polygons and other geometric figures that can be formed from it (i.e. polygons, polylines, and arcs) [5] in an Euclidean plane. The VSMST Data Model is based on the linear model of time, where time is represented as a discrete variable. This is based on the definition in the previous section which defines behavior of time that relates to the surface movement. In reality, events are recorded in a discrete manner, hence time can be represented in the data model. The relational approach has been used to design and implement the VSMST Data Model to ensure that it is flexible enough so that this model can be implemented in various database systems in the market. In this section, we will discuss the development of the database to manage and visualize surface movement data in a volumetric surface.

4.1 Database Development

Volumetric Surface object is created by the surface (s), polygon (p), line (l), and point (v). In representing spatiotemporal objects, time (t) is also involved to represent the period of changes and duration of changes. The changes on the surface comes from the points of the object that affects the lines that are joined together to build a triangle or polygon and surface.

The normalization of entities, attributes and relationships provides the logical model. The first level of normalization process is identifying the entity and attribute with relationship. Second level is identify dependency of the attribute with the entity. In this case, a new attribute will be created if required. Third level is to identify recursive attribute in the entity and come out with minimum redundancy of the attribute. Here, the management of the temporal element (time) is very important so that it can represent a movement without any redundancy issue. For purposes of managing the surface movement, we need to create a new entity surface movement. Surface movement entity will cater surfaces and time in the spatiotemporal data management. However, a complicative

issue with point entity could be raised if time is included in the entity as an attribute. For this purpose, point entity were evolved into temporal point entity which contains point data with temporal element. In both entities we create an index as identification in the entity. Here, we produced suitable logical data model, represented in the Entity Relationship Diagram (ERD) in Fig. 2.

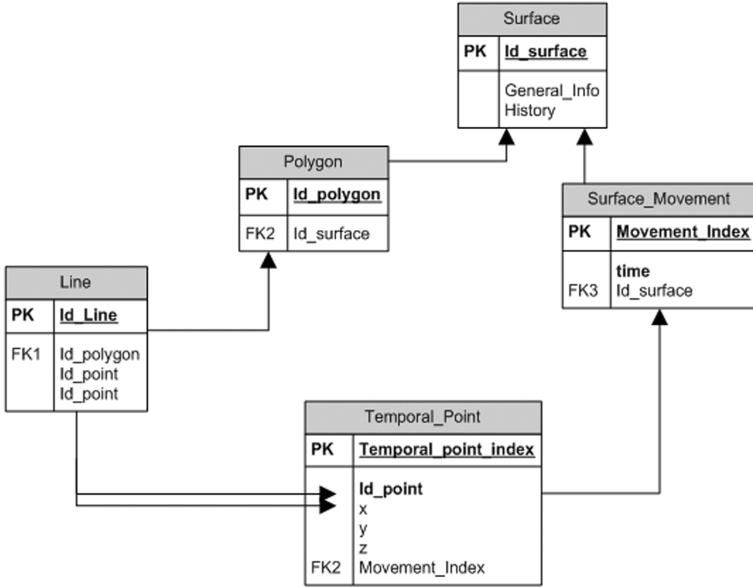


Fig. 2. Entity Relationship Diagram for VSMST Data Model

In the ERD for VSMST shown in Fig. 2, there are five entities that need to be created in the database to store and manage volumetric surface movement data. These are Surface_Movement, Surface, Polygon, Line and Temporal_Point. Every entity has relationships with related attribute. Table 2 shows the description of these entities. Table 3 shows the relationship of these entities with related entity.

4.2 Data Loading Process

In the data loading process, data will be loaded into the database from Triangulate Irregular Network (TIN). The data loading need to consider the format of the TIN structure and the database structure. From the TIN structure we captured the points, lines and triangles to be stored in the database, and we also considered the arrangement of the data to represent the correct surface. For loading of the data, identification of the data also needs to be indexed. Index of data currently is based on the feature identification with time. Every point has identification and their time as an attribute. The main principle of this database system, is when a point was not involved with movement, then it will not

Table 2. Description of the entities in the Logical Model

Entity	Description
Surface_Movement	Surface_Movement entities contain information about set of surfaces involved with movement. Attribute that will be stored in this entities are Movement_Index which is an indexing of the surface movement object, Id_surface which is surface involved with movement and also time of the occurring movement
Surface	Surface is an entity about the surface information. This is the place where this model can be extended with other information. In this entity we have Id_surface which is index of surface, General_Info and History which is an additional attribute based on user requirement
Polygon	Polygon is an entity that will store the information about the polygons on a particular surface. Thus, in this entity we have an Id_surface polygon and Id_surface
Line	Line stores information about construction of lines where the lines belongs to a particular surface. The attribute are are Id_line, id_point (start point), id_point (end point) and Id_polygon
Temporal_Point	Temporal_Point basically contains information about data which created the surface movement. So the attribute must be a coordinate of (x,y,z) and the temporal component. Temporal element is the same as time involved in the surface movement, so that we just include the Movement_Index as one of the entity. However, we have temporal_point_index to index the data which involved with the movement in the specific surface

Table 3. Description of Relationship in the ERD

Entity	Relationship	Entity
Surface_Movement	Has	Surface
Polygon	Member of	Surface
Line	Member of	Polygon
Line	Construct by	Temporal_Point
Temporal_Point	Movement	Surface_Movement

be stored. This will avoid data redundancy of the points in the database. The algorithm to perform this operation is given below.

1. Read data from the data source based on time (t)
2. While not end of file
3. read $\langle id_vertex, x, y, z \rangle$
4. if $\langle id_vertex, x, y, z \rangle$ not in database

5. *load the data*
6. *else if id_vertex on the database*
7. *if $\langle x, y, z \rangle = \langle x, y, z \rangle$ in database*
8. *does not involve with changes, do not store the point*
9. *else store $\langle id_vertex, x, y, z \rangle$ with time (t)*
10. *end*

4.3 Data Retrieval Process

Query process was designed to retrieve data from database by using area identification. A surface entity in the database has a `Id_Surface` with a start time and end time. Thus, results from this query will be a set of points which represents volumetric surface at the start time and end time. Below is the query algorithm that was used.

Data Retrieval Process:

1. *Get input from user; Area of the surface (A), Start Time (Ts) and End Time (Te)*
2. *Select Surface required based on the A and Time in between Ts and Te*
3. *Select Point based on the coordinate of area A and Time in between Ts and Te*
4. *Create surface based on the Time Start (Ts)*
5. *Calculate number of changes involved in the movement process*
6. *Identify list of vertex involved with changes based on time between Ts and Te*

This data will then be transformed into a file format which will be used to visualize the volumetric surface movement interactively by navigation of time. The file format uses data which were constructed from the start time given by the user. Number of changes in the selected area will be stored effective from the reformed start time. Then data (set of points) will follow subsequent changes and continue until the end of changes. Every change will only contain a set of data which were involved during the changes.

5 Data Visualization

In general, there are two steps in visualizing these changes. The first step is to establish a mapping from each point on the source object to some point at the targeted object. Once the correspondence is established, a sequence of intermediate models is created by interpolating corresponding point on the surface of the source objects to their position on the surface of targeted objects. Computing correspondences are done with respect to the minimal center distances. The established correspondence is one-to-one and thus, one vertex on source object

is paired with one vertex on targeted object. The interpolation part is solved by using parametric equation. The equation between two values used is:

$$Mv = P_{ti} + \{ (P_{u+1} - P_{ti}) * l \}$$

Whereby *l* represents value that ranges from (*t_s*) to (*t_e*)

M_v represents the InBetween value

P_{ti} represents point in start time

P_{u+1} represents point in end time.

This formula needs to be applied to every parameter that will vary during the interpolation. For instance, in this simple application, 3 spatial dimensions system is used and those parameters would be *x*, *y* and *z*.

6 Results of Implementation

This section will discuss the result of implementation of the model with the prototype application. For the testing phase, conceptual data were used which is 16 points with two sets of movements, 12.5% percent of the data were involved

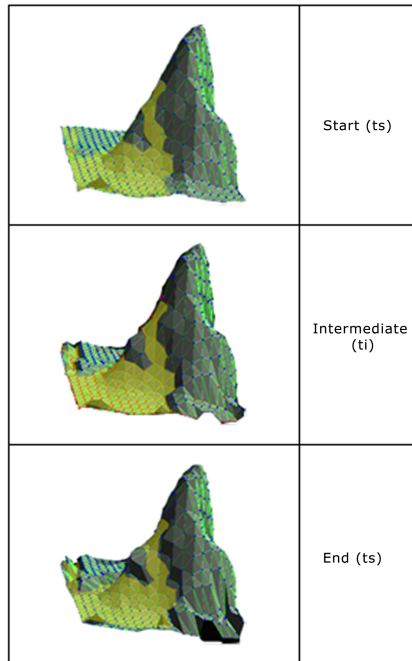
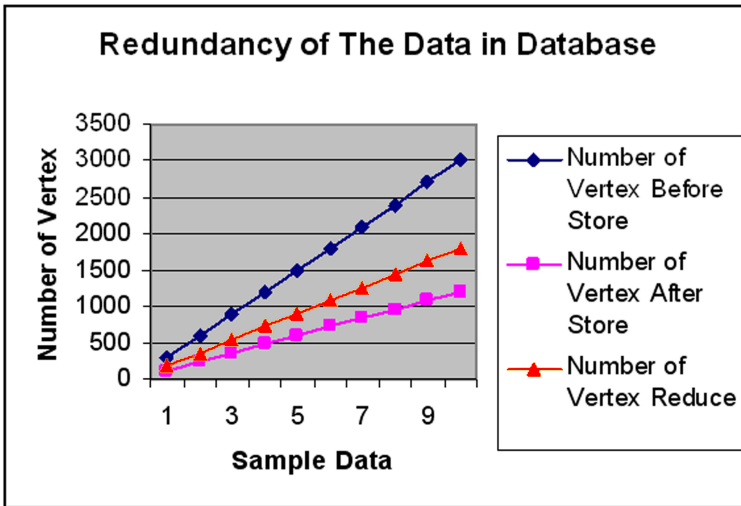


Fig. 3. Simulation of the Surface from Start Time (*t_s*) to End Time (*t_e*)

Table 4. Redundancy test before and after loading process

Data Set	No. of Vertex Before Store	No. of Vertex After Store	Vertex Reduced
1	304	122	182
2	612	252	360
3	923	385	538
4	1212	492	720
5	1507	614	893
6	1809	729	1080
7	2123	863	1260
8	2454	966	1488
9	2732	1124	1608
10	3008	1235	1773

**Fig. 4.** Graph of the Data Redundancy before and after loading data into database

with movement. Before loading, there are 48 points in which only 20 remain after loading process. So, we were able to reduce the data by 58.3% by avoiding data redundancy. Figure 3 shows the simulation of data from the start time (ts) to end time (ts).

From the testing result of ten data samples, this model were able to reduce around 60% of the data. Overall the data has been sampled by assuming that 10% of the vertex points will be involved in the movement in the surface with three times of movement. However the reduction of the data is based on how many data is stored and how many vertex points involved with movement. Table 4 shows the result of the testing process and Figure 4 illustrates the graph result.

7 Conclusion and Future Works

According to [14], a successful VGIS query process should be able to support different user preferences in spatiotemporal scene queries, and do not have a fixed-metric approach where all users are considered equal. This requires a spatiotemporal database that can integrate series of data. The Volumetric Surface Movement Spatiotemporal Data Model can be an ideal solution to solve this kind of issue. We have developed a prototype having the ability to reflect surface changes of a volumetric object. We have also included the ability to efficiently compress data not involved in the change. This model has shown to be able to reduce data redundancy by 60%.

As a conclusion, our major contribution in this paper is the implementation of the Volumetric Surface Movement Spatiotemporal Data Model by using the relational database model in a prototype version. The proposed data model can be considered in the development of the related application. In the visualization tool we have used a data format which were generated from the Spatiotemporal Data Retrieval Processor to load into the Volumetric Surface Movement Simulator. This aids the use in visualizing the result.

Acknowledgments. This research has been sponsored by the Ministry of Science, Technology and Innovation (MOSTI), Malaysia. Research has been managed by Research Management Centre, University Technology Malaysia under research grant 79102. Special thanks to the Institute of Advanced Technology for the advice and guidance given during research.

References

1. Yuan, M., Mark, D.M., Egenhofer, M.J., Peuquet, D.J.: Extensions to Geographic Representations. In: McMaster, R.B., Usery, E.L. (eds.) *A Research Agenda for Geographic Information Science*, Boca Raton, Florida, ch. 5, pp. 129–156. CRC Press, Boca Raton (2004)
2. Nadi, S., Delavar, M.R.: Spatio-Temporal Modeling of Dynamic Phenomena in GIS. In: *Proceeding of ScanGIS 2003*, pp. 215–225 (2003)
3. Rahim, M.S.M, Shariff, A.R.M, Mansor, S., Mahmud, A.R.: A Review on Spatiotemporal Data Model for Managing Data Movement in Geographical Information Systems (GIS). *Journal of Information Technology, FSKSM, UTM* 1(9), 21–32 (2005)
4. Rahim, M.S.M, Shariff, A.R.M, Mansor, S., Mahmud, A.R.: Volumetric Spatiotemporal Data Model, *Lecture Note in Geoinformation and Cartography*, pp. 547–556. Springer, Heidelberg
5. Narciso, F.E.: *A Spatiotemporal Data Model for Incorporating Time in Geographical Information Systems (GEN-STGIS)*, Doctor of Philosophy in Computer Science and Engineering, University of South Florida (April 1999)
6. Ale, R., Wolfgang, K.: Cell Tuple Based Spatio-Temporal Data Model: An Object Oriented Approach, *Geographic Information Systems, Association of Computing Machinery (ACM)*, 2-6 November 1999, pp. 20–25. ACM Press, New York (1999)

7. Moris, K., Hill, D., Moore, A.: Mapping The Environment Through Three-Dimensional Space and Time, Pergamon, Computers. Environment and Urban Systems 24, 435–450 (2000)
8. Donggen, W., Tao, C.: A Spatio-temporal Data Model for ActivityBased Transport Demand Modeling. International Journal of Geographical Information Science 15, 561–585 (2001)
9. Philip, J.U.: A Spatiotemporal Data Model for Zoning, Masters of Science Thesis, Department of Geography, Brigham Young University (2001)
10. Bonan, L.,Guoray, C.: A General Object-Oriented Spatial Temporal Data Model, Symposium on Geospatial Theory, Processing and Applications, Ottawa (March 2002)
11. Elena, C., Michela, B., Elisa, B., Giovanna, G.: A Multigranular Spatiotemporal Data Model. In: GIS'03, New Orleans, Louisiana, USA, November 7-8, 2003, pp. 94–101 (2003)
12. Yanfen, L.: A Feature-Based Temporal Representation and Its Implementation with Object-Relational Schema for Base Geographic Data in Object-Based Form, UCGIS Assembly (2004)
13. Roddick, J.F., Egenhofer, M.J, Hoel, E., Papadias, D.: Spatial, Temporal, and Spatiotemporal Databases. In: SIGMOD Record Web, vol. 33(2) (2004)
14. Mountrakis, G., Agouris, P., Stefanidis, A.: Similarity Learning in GIS: An Overview of Definitions, Prerequisites and Challenges, Spatial Databases Technologies, Techniques and Trends, pp. 295–321. Idea Group Publishing, USA (2005)
15. Anders, W.: Book Review: Next Generation of Geospatial Information from Digital Image Analysis to Spatiotemporal Databases by Peggy Agououris and Arie Croitoru. Transaction in GIS 10(5), 773–775 (2006)