The Relationship among GIS-Oriented Spatiotemporal Databases

Lixin Li and Peter Revesz
Department of Computer Science and Engineering
University of Nebraska-Lincoln, Lincoln
NE 68588, U.S.A.
e–mail: {lli, revesz}@cse.unl.edu

Abstract

We overview three major types of GIS-oriented spatiotemporal databases: (1) point-based, (2) region-based, and (3) constraint-based. We analyze the relationship among these spatiotemporal databases and show how they can be translated into each other. We illustrate the translations with examples from the Unified Climate Access Network (UCAN) and National Agricultural Statistics Service (NASS) databases. Finally, we also discuss the advantages and disadvantages of using the various types of spatiotemporal databases.

1. Introduction

GIS (Demers 2000, Longley et al. 2001, Worboys 1995) applications increasingly require the use of spatiotemporal data, that is, data that combine both space and time (Langran 1992). For example, land-use change through time is a typical spatiotemporal data. Future GIS systems need to efficiently manage spatiotemporal databases (STDBs) containing such data. The study of the representation and the algorithmic methods to query and visualize spatiotemporal data is still a growing research area. However, it is already becoming clear that we can distinguish three major types of GIS-oriented spatiotemporal databases.

We argue in this paper that spatiotemporal data can be represented by three basic primitives: points, regions, and constraints. We describe point-based STDBs in Section 2, region-based STDBs in Section 3, and constraint-based STDBs in Section 4. It should be emphasized that these three types of spatiotemporal databases are only alternative representations of the same data. Any one type of spatiotemporal database can be translated into another type as shown in Figure 1. In the figure, each edge represents a translation algorithm. For example, edge A represents the algorithm that converts a point-based spatiotemporal database into a constraint-based spatiotemporal database. We postpone further discussion of Figure 1 and the translation algorithms until Section 5.

2. Point-Based Spatiotemporal Databases

A point-based STDB consists of a set of point-based spatiotemporal relations. For 2-D space and 1-D time problems, a point-based spatiotemporal relation should have the schema of (x, y, t, w_1, w_2, ..., w_m). The attributes (x, y) specify point locations and t specifies a time instance. In GIS applications, (x, y) may be given as UTM (Universal Transverse Mercator) coordinates for easting and northing. The last m attributes w_i ($1 \leq i \leq m$) record the other features at location (x, y) and time t.

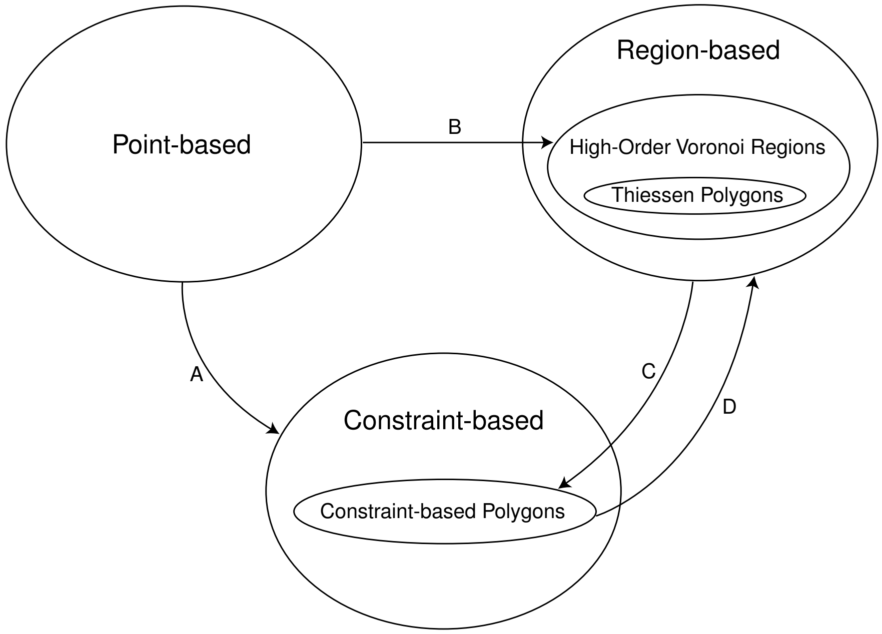

Figure 1: The relationship among the spatiotemporal databases.

Example 2.1 The point-based spatiotemporal relation *Drought_Point(x, y, year, SPI)* stores the average yearly SPI (Standardized Precipitation Index) values sampled by 48 major weather stations in Nebraska from year 1992 to 2002. SPI is one of the common and simple measures of drought which is based solely on the probability of precipitation for a given time period. Values of SPI range from 2.00 and above (extremely wet) to -2.00 and less (extremely dry) with near normal conditions ranging from 0.99 to -0.99. A drought event is defined when the SPI is continuously negative and reaches a value of -1.0 or less, and continues until the SPI becomes positive. We obtained the *Drought_Point* relation as shown in Table 1 from the Unified Climate Access Network (UCAN).

Drought_Point

x (easting)	y (northing)	year	SPI
-315515.56	2178768.67	1992	0.27
-315515.56	2178768.67	1993	-0.17
⋮	⋮	⋮	⋮
-315515.56	2178768.67	2002	0.19
-163932.36	2075263.16	1992	-0.19
⋮	⋮	⋮	⋮
-133759.02	1985122.32	2002	-0.22

Table 1: A point-based spatiotemporal relation.

However, from this example, we can see that the disadvantage of point-based STDBs is that the measured information, such as SPI, only exists at certain sampled locations and times. Therefore, we are not able to query the values at unsampled locations and times.

3. Region-Based Spatiotemporal Databases

A region-based STDB database has both spatial and temporal parts. The spatial part has schema *(region-id, boundary)*. The *region-id* is a unique identifier of each polygonal shaped region, and the *boundary* is the sequence of its corner vertices. The spatial part can be stored in an Arc/GIS database (Johnston et al. 2001). The temporal part has schema (*region-id*, t, w_1, w_2, ..., w_m), where t is the time attribute and each w_i represents some other characteristics of the region.

Nebraska_Corn_Space_Region

county	boundary
1	{ (-656160.3, 600676.8), (-652484.0, 643920.3), (-607691.1, 639747.6), (-608934.8, 615649.0), (-607875.6, 615485.8), (-610542.0, 576509.1), (-607662.7, 576138.5), (-611226.9, 537468.5), (-607807.7, 536762.1), (-608521.1, 527084.0), (-660885.4, 531441.2), (-661759.8,532153.1) }
⋮	⋮

Nebraska_Corn_Time_Region

county	year	practice	acres	yield	production
1	1947	irrigated	2700	49	132300
1	1947	non-irrigated	81670	18	1470060
1	1947	total	84370	19	1602360
⋮	⋮	⋮	⋮	⋮	⋮
1	2000	irrigated	141300	161	22749300
1	2000	non-irrigated	27900	73	2036700
1	2000	total	169200	146.5	24786000
⋮	⋮	⋮	⋮	⋮	⋮

Table 2: A region-based spatiotemporal database with separate spatial and temporal relations.

Example 3.1 A NASS (National Agricultural Statistics Service) region-based spatiotemporal database shows the yearly corn yield and production in each county of the state of Nebraska. The spatial part of the database is shown in the upper half of Table 2 which uses the vector representation of counties in Nebraska, while the temporal part is shown in the lower half of Table 2. Note that in the relation *Nebraska_Corn_Time_Region*, {*county*, *year*, *practice*} is the primary key, because it is the minimal set of attributes that functionally determine the other three attributes. The attributes used in these two tables are the following:

- *county*, the common attribute between the spatial and temporal relations, is the Federal Information Processing Standards (FIPS) id that is unique for each county in a state.
- *boundary* is a sequence of corner vertices on the boundary of a county. For example, the county with id 1 is a polygon that is represented by its 12 corner vertices.
- *year* is the time.
- *practice* is one of the following types: irrigated, non-irrigated, and total.
- *acres* is the number of acres that are harvested with total = irrigated + non-irrigated.
- $yield = bushels/acres$.

- *production* = *acres* × *yield* = *total bushels* (total = irrigated + non-irrigated).

Although region-based STDBs cover the continuous area and do not have the problem of missing information as in point-based STDBs, the information accuracy in region-based STDBs is reduced by assigning one uniform value to each region.

4. Constraint-Based Spatiotemporal Databases

Constraint databases (Kanellakis et al. 1995, Revesz 2002) provide a natural representation for spatiotemporal objects when their trajectory can be described as simple mathematical functions. Constraint-based STDBs can represent both point-based and region-based spatiotemporal objects, where each attribute is associated with an attribute variable. For example, the spatial attributes can be associated with x and y variables for representing point-based spatiotemporal objects or with a *region_id* variable for region-based spatiotemporal objects, while the temporal attributes can be associated with *year* or other time variables. A constraint-based relation is a finite set of constraint tuples. The value of the attributes in a relation is specified implicitly using (arithmetic) constraints such that each constraint tuple is a conjunction of constraints using the same set of attribute variables.

Example 4.1 The region-based NASS corn data in Example 3.1 can be represented by spatial and temporal parts in the constraint-based STDB shown in Table 3. In the spatial part of Table 3, each county polygon can be either convex or concave, we break each county polygon into a set of adjacent triangles for the convenience of implementation, where each triangle can be represented by a conjunction of three linear arithmetic constraints. For example, the county with FIPS code 1 consists of three triangles.

The temporal part of Table 3 compresses the region-based *Nebraska_Corn_Time_Region* relation in Example 3.1 by storing the linear regression functions of *acres*, *yield* and *production* according to *year*. The linear regression functions show the trend of these attributes during the past years. So they are useful to predict the future events. For example, what will the the yield of irrigated corn in county 1 approximately be in 2010? The entire *Nebraska_Corn_Time_Constraint* relation consists of only 279 constraint tuples, while *Nebraska_Corn_Time_Region* contains 14, 788 original tuples. Although the accuracy in this example is not very high because of the approximation by only one linear regression functions, we can improve it by inserting more segments of lines. Revesz et al. (2001) discusses the detail of how to use piece-wise linear functions to compress data in constraint databases. We also could use non-linear approximation functions to increase the accuracy. Note that because of the tradeoff between the accuracy and storage, the above improvements will increase the number of tuples in the temporal relation (i.e. *Nebraska_Corn_Time_Constraint*) of Table 3.

In summary, users can query the measured information at any location and time in constraint-based STDBs. This is similar to region-based STDBs: there is no missing information. Moreover, constraint-based STDBs are more sophisticated than region-based STDBs. In region-based STDBs, all the points in a region can only be given one uniform value. However, in constraint-based STDBs, the points in a region can be assigned with different values. If we could find appropriate functions (constraints) of x, y and t to interpolate the values inside each region and store them in the constraint-based STDBs, we can get a good result.

Nebraska_Corn_Space_Constraint

county	easting	northing	
1	x	y	$81.15x - y \leq -15827909.86$, $-0.02x - y \geq -356911.63$, $-1.11x - y \leq -108714.89$
1	x	y	$7.69x - y \geq -2116031.10$, $33.21x - y \geq -7938392.36$, $29.71x - y <= -7135823.80$
1	x	y	$-0.03x - y \leq -315048.67$, $-1.11x - y \geq -108714.89$, $29.71x - y >= -7135823.80$
⋮	⋮	⋮	⋮

Nebraska_Corn_Time_Constraint

county	year	practice	acres	yield	production	
1	t	irrigated	a	y'	p	$1947 \leq t$, $t \leq 2000$, $a = 3453t - 6729678$, $y' = 2t - 3890$, $p = 540016t - 1054525400$
1	t	non-irrigated	a	y'	p	$1947 \leq t$, $t \leq 2000$, $a = -1000t + 1995747$, $y' = 2t - 3200$, $p = 6839t - 12608086$
1	t	total	a	y'	p	$1947 \leq t$, $t \leq 2000$, $a = 2453t - 4733931$, $y' = 3t - 5069$, $p = 546854t - 1067133500$
⋮	⋮	⋮	⋮	⋮	⋮	⋮

Table 3: A constraint-based spatiotemporal database with separate spatial and temporal relations.

5. The Relationship among Spatiotemporal Databases

Now we are ready to explain the edges in the Figure 1 as follows:

(A) Point-based STDBs can be converted to constraint-based STDBs by certain interpolation methods, such as shape functions (Zienkiewics and Taylor 2000), spline functions (Goodman and O'Rourke 1997), IDW (Shepard 1968), and Kriging (Deutsch and Journel 1998). The details about using shape functions to convert point-based STDBs to constraint-based STDBs can be found in (Revesz and Li 2002b, Li and Revesz 2003 in press); and the translation details using IDW can be found in (Revesz and Li 2002a). The comparison of using shape functions and IDW to represent point-based data in constraint-based STDBs is analyzed in (Li and Revesz 2002) based on a specific set of data.

(B) Point-based STDBs can be used to generate region-based STDBs, including Voronoi region-based STDBs, by treating a region as a unit and assigning a uniform value for each region. A special case of region-based STDBs is the Worboys relations (Worboys 1995) where each region is divided into a set of triangles.

(C) Region-based STDBs can be converted to constraint-based STDBs by using linear arithmetic constraints to represent region boundaries and compressing the temporal part by some types of functions in time, such as linear regression functions in Example 4.1.

(D) Constraint-based STDBs (Revesz 2002) can be converted to region-based STDBs by simply assign a uniform value to each region, such as the average of all the nearest neighbors.

References

M. N. Demers. *Fundamentals of Geographic Information Systems.* John Wiley & Sons, New York, 2nd edition, 2000.

C. V. Deutsch and A. G. Journel. *GSLIB: Geostatistical Software Library and User's Guide.* Oxford University Press, New York, 2nd edition, 1998.

J. E. Goodman and J. O'Rourke, editors. *Handbook of Discrete and Computational Geometry.* CRC Press, Boca Raton, New York, 1997.

K. Johnston, J. M. Ver Hoef, K. Krivoruchko, and N. Lucas. *Using ArcGIS Geostatistical Analyst.* ESRI Press, 2001.

P. C. Kanellakis, G. M. Kuper, and P. Revesz. Constraint query languages. *Journal of Computer and System Sciences*, 51(1):26–52, 1995.

G. Langran. *Time in Geographic Information Systems.* Taylor and Francis, London, 1992.

L. Li and P. Revesz. A comparison of spatio-temporal interpolation methods. In M. Egenhofer and D. Mark, editors, *Proc. of the Second International Conference on GIScience 2002*, volume 2478 of *Lecture Notes in Computer Science*, pages 145–160. Springer, 2002.

L. Li and P. Revesz. Interpolation methods for spatio-temporal geographic data. *Computers, Environment and Urban Systems*, 2003 in press.

P. A. Longley, M. F. Goodchild, D. J. Maguire, and D. W. Rhind. *Geographic Information Systems and Science.* John Wiley, Chichester, 2001.

P. Revesz. *Introduction to Constraint Databases.* Springer, New York, 2002.

P. Revesz, R. Chen, and M. Ouyang. Approximate query evaluation using linear constraint databases. In *Proc. Symposium on Temporal Representation and Reasoning*, pages 170–175, Cividale del Friuli, Italy, 2001.

P. Revesz and L. Li. Constraint-based visualization of spatial interpolation data. In *Proc. of the Sixth International Conference on Information Visualization*, pages 563–569, London, England, 2002a. IEEE Press.

P. Revesz and L. Li. Representation and querying of interpolation data in constraint databases. In *Proc. of the Second National Conference on Digital Government Research*, pages 225–228, Los Angeles, California, 2002b.

D. Shepard. A two-dimensional interpolation function for irregularly spaced data. In *Proc. 23nd National Conference ACM*, pages 517–524. ACM, 1968.

M. F. Worboys. *GIS: A Computing Perspective.* Taylor & Francis, 1995.

O. C. Zienkiewics and R. L. Taylor. *Finite Element Method, Vol. 1, The Basis.* Butterworth Heinemann, London, 2000.