퍼지 시공간 데이터베스를 위한 질의 연산

Vu Thi Hong Nhan, 지정희, 류근호
충북대학교 데이터베이스 연구실
{nhanvth,jhchi,khryu}@dblab.cbu.ac.kr

Query Operations for Fuzzy Spatiotemporal Databases

Vu Thi Hong Nhan, Jeong Hee Chi, Keun Ho Ryu
Database Laboratory of Chungbuk National University

Abstract

GIS (geographic information system) applications increasingly require the representation of geospatial objects with fuzzy extent and querying of time-varying information. In this paper, we introduce a FSTDB (fuzzy spatiotemporal database) to represent and manage states and events causing changes of dynamic fuzzy objects using fuzzy set theory. We also propose the algorithms for the operators to be included in a GIS to make it able to answer queries depending on fuzzy predicates during a time interval and a method to identify the development process of objects during a certain period based on the designed database. They can be used in application areas handling time-varying geospatial data, including global change (as in climate or land cover change) and social (demographic, health, ect.) application.

1. Introduction

Most natural spatial phenomena are not crisply delineated, but are bounded by transition zone. For example, the boundary between the grassland *A* and woodland *B* may be gradual through a transition zone rather than a crisp boundary. Moreover, these objects change from time to time. Questions relating to spatial relations of those objects can be "Does a region *A a little overlap* or *somewhat contains* a region *B* at some time *t*?" GIS applications increasingly require the representation of geospatial objects with fuzzy extent and querying of time-varying information. So far there have been several models proposed to represent objects with indeterminate boundaries without concerning the time dimension of data, and A few researches have been reported on the dynamic behavior of such objects [3,5,7,8,13]. On the other hand, a large number of spatiotemporal data models have been

* 본 연구는 대학 IT 연구센터 육성 및 지원사업의 연구결과로 수행되었음

proposed to deal with dynamic spatial objects, but they do not accommodate fuzzy nature of objects [2,6,9,12]. In this study, we describe a fuzzy spatiotemporal database based upon fuzzy set theory, which is capable of not only managing lifelines of objects or states of objects, but also describing the events causing object changes during their lifelines. Then, we introduce the algorithms for operators to be included in a GIS to make it able to answer queries depending on fuzzy spatial predicates during a time interval. The method for approximately analyzing binary topological relations between fuzzy spatial objects based upon 9IM model in [4]. We also describe a method of identifying the development process of objects during a certain period using the designed database. They can be used in application areas handling time-varying geospatial data, including global change (as in climate or land cover change), social (demographic, health, ect.) applications.

The rest of this paper is organized as follows: in the next section we review previous works followed by a design of FSTDB. In section 4 we construct fuzzy spatiotemporal relationship operators and an algorithm to retrieve the development of an object. Finally, we draw conclusions and give future work in section 5.

2. Related Work

This section summarizes several typical spatiotemporal data models, models for geospatial objects with fuzzy boundaries, and a definition of fuzzy objects.

2.1 Spatiotemporal Data Models

Till now, most spatiotemporal data models assume that spatial objects have crisp boundaries and precisely define relationships with others. Snapshot models that support the world's existence at present or as a collection of the temporal snapshots, which is unable to track different versions of the same object over several non-related tuples within the same table. As an alternative of the snapshot models, time-based models such as amendment vector approach and ESTDM have been suggested in order to identify individual changes or events to the data set between snapshots [2,6,9]. Another model proposed together with a specification of spatiotemporal database query language [12] being lately introduced is directly applicable to applications without any modification.

2.2 Geospatial Objects with Fuzzy Boundaries

There are several approaches available to represent vague objects. The approach in [3] extends the indeterminate boundary of a region into a boundary zone, called broad boundary, which is situated around the region. Because this model treats a boundary as a thick boundary, finer distinctions between points lying within it cannot be made [11]. The concept of vague regions in [5] generalize this approach in the sense that such a region can be a pair of arbitrarily located and discrete crisp regions. This view presents a very coarse and restricted description of fuzzy regions, since it differentiates only between three parts. Thus the original gradation in the membership values of the points of the boundaries gets lost [7]. They do not base on fuzzy set theory while it is considered as a tool for dealing with fuzziness and vagueness.

2.3 Definition of Fuzzy Regions and Fuzzy Spatial Relationships

Fuzzy regions defined based on fuzzy set theory take a view as a collection of nested crisp α-cut regions [7,11]. Spatial objects are considered in 2D space. Let $\mu_F: X \times Y \rightarrow [0,1]$ be the membership function of a fuzzy region F. An α-cut region F_α for $\alpha \in [0,1]$, is defined as $F_\alpha = \{(x,y) IR^2 | F(x,y)\}$, whose boundary is defined by all points with membership values of α. The kernel of F is then equal to F_1. Hence F is represented as $F = \{F_{\alpha i} | 0 \leq i \leq \Lambda_F\}$ with $\alpha_i > \alpha_{i+1}$ $F_{\alpha i} \subseteq F_{\alpha i+1}$ for $0 \leq i \leq |\Lambda_F| - 1$ and Λ_F is the level set $\alpha \in [0,1]$ representing distinct α-cuts of the F.

A fuzzy region F is a simple fuzzy region if it is fully connected and convex. A fuzzy region is said to be fully connected if and only if its α-cuts are connected for all $\alpha > 0$.

With such the view of a fuzzy region, fuzzy spatial topological relationships are determined by computing the topological relationship between collections of α-cut regions [7,11]. Denote by $\pi_f(P,S)$ the value representing the topological relations between two fuzzy regions P and S is determined by the formula below:

$$\pi_f(P,S) = \sum_{i=1}^{N} \sum_{j=1}^{N} (\alpha_i - \alpha_{i+1})(\alpha_j - \alpha_{j+1}) \pi_{cr}(P_{\alpha i}, S_{\alpha j})$$

where N is the number of α-cuts; $\pi_{cr}(P_{\alpha i}, S_{\alpha j})$ is the topological relation between two crisp α-cuts identified by 9IM [4]. The values of the aggregated topological relations between P and S fall into the range [0,1]. Consequently a set of eight fuzzy spatial predicates is gained T_f = {*disjoint*$_f$, *meet*$_f$, *overlap*$_f$, *covers*$_f$, *coveredBy*$_f$, *equal*$_f$, *inside*$_f$, *contains*$_f$}.

3. A Design of FSTDB

This section launches a design of FSTDB based on integration of independent researches on spatial and temporal databases. In term of spatial databases, whenever a new object instance is inserted into a database, the old one should be deleted. It does not support the ability of managing the historical information about geospatial objects that have been changed with temporal evolution. In view of temporal databases it is extremely hard to mange directly spatial objects without any modification.

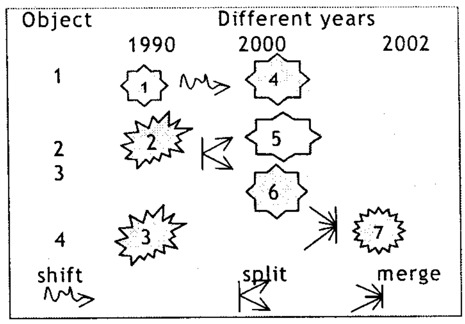

Figure.1 States and process of objects through time serious

Fuzzy regions are extracted from field observation data. It is assumed that the change of natural phenomena is a gradual continuous process. Thus regions at different times should be linked to form lifeline of the objects. This implies that if two regions are the spatial extents at two subsequent times of the same object, their overlap should be larger than their overlaps with the region of any other object. Under this assumption we can find the successor of a region at time t_n by calculating its spatial overlaps with all the regions that appeared at each time t_{n+1}. The one that has the maximum overlap will be identified as the successor. Regions, as illustrated in figure.1, thus can be linked to form lifelines of objects that may have *shifted, expanded, shrunk* between two successive times. The region that appears at a specific moment represent a new object, and regions that disappear at some moment represent disappearing objects. Furthermore, *merging* and *splitting* objects can be identified.

Design of relations in database as follows: Current_Object(<u>*OId*</u>, *AlpCut, BWidth, spat$_1$,...,spat$_n$, Geo, NonSpat, VTs, EType*) where *OId* is identifier of objects; *AlpCut* represents the value of α-cuts; *BWidth* denotes the width of object boundaries; *spat$_i$* are geometric elements describing spatial extents of objects, show geometric elements; *Geo* shows the actual shape and size of the objects in a metric sense; *NonSpat* is a non-spatial attribute vector of objects; *VTs* is the starting time of the valid time vector *VT< VTs, VTe>* and *VTe* is the ending time; *EType* indicates state transitions of objects. Relation Link(<u>*OId*</u>, *HId*) is generated since a state transition can be related to many objects, for example *merge* event, in that *Hid* is a historical pointer identifying father of object existing in History_Object. History_Object(<u>*HId*</u>, *AlpCut, BWidth, spat$_1$,...,spat$_n$ Geo, NonSpat, VT, EType, Prev*) in which *Prev* denotes the previous pointer of the object determined by *OId* in this relation.

4. Integration of Fuzzy Spatial and Temporal Relationship Operators

This part describes an algorithm defining fuzzy spatiotemporal relationship operators and another one allows to identify the history of a given object.

4.1 Fuzzy Spatiotemporal Relationship Operator

Fuzzy spatiotemporal operations are motivated by the simultaneous integration of temporal and fuzzy spatial operations in queries. Incorporating time values into a query language is to determine the temporal relationships between objects.

Table 1. Time relational semantic expression

Temporal Operators	Temporal Semantics Expression
a equals b	*Begin(a)=Begin(b)* ∧ *End(a)=End(b)*
a overlaps b	*(Begin(a)<Begin(b)* ∧ *End(a)>Begin(b)* ∧ *End(a)<End(b))* ∨ *(Begin(a)>Begin(b)* ∧ *Begin(a)<End(b)* ∧ *End(a)>End(b)*
a during b	*(Begin(a)>Begin(b)* ∨ *Begin(a)=Begin(b))* ∧ *(End(a)<End(b)* ∨ *End(a)=End(b)*
a starts	*Begin(a)=Begin(b)* ∧ *End(a)=End(b)*
a finishes b	*Begin(a)>Begin(b)* ∧ *End(a)=End(b)*
a before b	*End(a)<Begin(a)*

The semantics of some temporal relationship operators [1] are presented as relational expressions as in table 1, where *a* and *b* are two time interval whose elements are extracted by functions *Begin*() and *End*()

Fuzzy spatial relationship predicates have values falling in the range [0,1] which are embedded in queries by using their qualitative linguistic descriptions as an interpretation of the membership values. For example, depending on the value of $overlap_f$ we can distinguish *not overlap, quite overlap, and completely overlap. not, quite,* and *completely* are called *fuzzy quantifiers,* each is represented by an appropriate fuzzy set with the membership function. Before applying we need to define a classification for them: Create Fuzzy_Quantifier(*not, quite, · completely*); then active it: Set Fuzzy_Quantifier.

We are now ready to define fuzzy spatiotemporal operators by the algorithm 1. Given two fuzzy regions *P, S*; a fuzzy quantifier *Fq*, and a spatial predicate *Spre* which create a fuzzy spatial predicate *Fq.Spre,* and a temporal predicate *Top.* The algorithm proceeds two steps. First, check the temporal relationship using table 1 for *Top.*

If this condition is satisfied, go ahead checking fuzzy spatial condition. In this step, the aggregated spatial relationship between two given regions have to be calculated, the determined value is used to examine whether it matches values defined for *Fq.Spre.* The algorithm returns TRUE if both fuzzy spatial condition and temporal condition are satisfied, otherwise FALSE value will be returned.

Algorithm 1. Fuzzy spatiotemporal operators

```
Function F_Spatiotemporal_Predicates(P, S, Top,
Fq, Spre)
Denote by fst_value=FALSE value indicating the
aggregated spatial relationships between P&S.
Step 1: Check temporal condition
 Get two time periods (T1sT1e) and (T2sT2e) of
P, S respectively. Check their relation Top using
table 1
Step 2: Examine fuzzy spatial condition
//calculate the aggregated spatial relationship
for(i=1;i<n;i++)//n is number of α-cuts
for(j=1;j<n;j++)
temp=0; crispRel=0; result=0;
Identify α-cut regions P_αi,S_αi
If their spatial relation matches Spre then
temp+ =(alp_j-alp_j1)*crispRel;
Identify α-cut regions P_αj,S_αj .
result+ =(alp_j-alp_j1)*temp;
if result coincides the values defined for Fq in
Fuzzy_Quantifer then fst_value=TRUE;
return fst_predicate;
End
```

4.2 The Operator Tracking the Development Process of Objects

Our FSTDB described above allows us to manage all historical objects that vary their states through time. It is necessary to retrieve objects during a given period. We propose an operator used in the select clause of the query statement to extract historical information of designed fuzzy spatiotemporal objects. All historical objects in FSTDB can be retrieved by using historical pointers. In addition the ability of extracting historical information, what events happened to objects at some time in their lifelines are also known. The steps processing the algorithm are described clearly in algorithm 2 with input being a given object *id* and a given time period *TP.*

Algorithm 2. The development process of objects

Function Development_Process(id)
historical_pointer=NULL;
Step1: Scan the relation Current_Object to find *id*.
If *id* exists then
If its valid time *VT* does not overlaps *TP* then exit;
Otherwise, Retrieve all the pointers *HId* such that *OId* matches *id* from the relation link.
Step 2: For each *HId* found,
If HId !=NULL set *historical_pointer=HId* which refer to the relation History_Object;
Step 3: For(;*historical_pointer* !=NULL;)
If (an object overlaps the valid time *VT*)
- Display the attributes of its object;
- Set its previous pointer Prev of designed object into *historical_pointer*;
End

4.3 Example Queries

We now illustrate how data are stored in our FSTDB using dynamic fuzzy geospatial objects describing in the figure 1. Next, we apply this design to we represent the expressions of some typical queries in which our newly proposed operators are used.

Whenever a new object instance is inserted, its old version is inserted into the relation History_Object and update its existing time period. For simplicity, some attribute fields such as central location of fuzzy objects and non-spatial attributes. Objects and their data are represented in the below relations.

OId	*VTs*	*Area*	*a-cuts*	*BWidth*	*EType*
4	2000	1137.8	1...100	..	shift
5	2000	586.8	1...100	..	split
7	2002	609.8	1...100	..	merge

Current_Object

OId	*HId*
4	1
5	2
7	6
7	3

Link

HId	*VTs*	*VTe*	*Area*	*a cuts*	*BWidth*	*EType*	*Prev*
1	1990	2000	1108.1	..	..	appear	null
2	1990	2000	644.3	..	..	appaer	null
6	2000	2002	586.8	..	..	split	2
3	1990	2002	28.0	..	..	appear	null

History_Object

Query 1: Find all current objects that completely inside a beach region having fuzzy boundary *A* since 2001.

Select *OId*
From Current_Object
Where *F_Spatiotemporal_Predicate(OId, A, complelely, inside 'VTs, Now' after '2001, Now')*;

Query 2: What objects were quite overlap the foreshore region having vague boundary *A* during 1989-1999

Select *OId*
From *History_Object*
Where *F_Spatiotemporal_Predicate(HId, B, quite, inside, '1989-1999 contains' VTs, VTe')*;

Query 3: List all of the histories of the region 7 from 1995-2003

Select *Development_Process(id=7)*
From *Current_Object a, Link, History_Obeject*
Where '*a.Ts, Now' ovelaps '1995, 2003'*;

Both query 1 and query 2 are fuzzy spatiotemporal queries. To answer them, we use the defined operator *F_Spatiotemporal_Predicate* in the WHERE clause with the parameters identified from the question. Output is a set of objects retrieving from the relations in database satisfying both temporal relationship and fuzzy spatial relationship with the given objects in the query.

Query 3 requires historical information during a specific time period *TP*. To process this query we use the operator *Development_Process* for object with *id* equal 7. The process of this object evolution is expressed by the sequence of states and the related state transitions provided by the stored data in relations. Concretely, object with identifier 7 is found in the Current_Object, its valid time overlap *TP*. We know that this object was merged by two objects 6 and 3. Following their pointers *Hid* in relation Link to retrieve their information in the relation History_Objects. For object 3, print all information of this object. For object 6, we know that it is splitted from object 2, so after printing its information continue to follow its previous pointer to retrieve its father in the same table. The processing stops after outputting information of 2.

5. Conclusions

This paper presents a design of FSTDB that manages all historical information of fuzzy spatial objects. We then proposed new fuzzy spatiotemporal relationship operators. Specially, the development process operator allows us to not only retrieve all historical information of objects, but also show what events such as split, merge, or shift occurred to objects at a specific time during a given time periods. This study can be used in application areas handling time-varying geospatial data, including global change (as in climate or land cover change) and social (demographic, health, ect.) applications.

We are going to build a FSTDB system for a specific application for example costal geomorphology study. Moreover, a suitable indexing method to efficiently retrieve fuzzy spatial objects is a research direction in the future.

References

1. F.J. Allen. Maintaining Knowledge About Temporal Intervals. Communication of the ACM 26, pp.832-843, 1983.
2. C. Claramun & M. Theriault. Managing Time in GIS: An Event-Oriented Approach. Recent Advances in Temporal Databases, Springer-Verlag, pp.23-42, 199.
3. E. Clementini & P.D. Felice. A Spatial Model for Complex Objects with a Broad Boundary Supporting Queries on Uncertain Data. DKE. vol.37, No.30, pp.285-305, 200
4. M.J. Enhofer & R. Franzosa. Point-set Topological Spatial Relations. Int. J GIS, vol.3, No.2, pp.161-174, 199
5. M. Erwig & M. Schneider. Vague Regions. 5th Int.Sym. on Advances in Spatial Databases, LNCS 1262, pp.298-320, Springer-Verlag, 1997
6. D.J. Peuquet & E.A. Wentz. An Approach for Time-base Analysis of Spatiotemporal Data. Advances in GIS Research Proceedings, pp.489-504, 199.
7. M. Schneider. A Design of Topological Predicates for Complex Crisp and Fuzzy Regions. Int.Conf. on Conceptual Modeling, pp.103-116, 1997
8. E. Stefanakis. A Unified Framework for Fuzzy Spatiotemporal Representation and Reasoning. Pro. 20th Int.Conf , pp.6-10, 2001
9. M.F. Worboys. A Unified model for Spatial and Temporal Information. The Computer

Journal. Vol.37, No.1, pp.27-34, 1994
10. L.A. Zadeh. Fuzzy Sets Information and Control, vol.8, pp.338-353, 1965.
11. F.B. Zhan Approximate Analysis of Topological Relations between Geographic Regions with Indeterminate Boundaries, Soft Computing, Vol.2, pp. 28-34, 199
12. D.H.Kim, K.H.Ryu, & C.H.Park. Design and Implementation of Spatiotemporal Database Query Processing System. Journal of System and Software, Vol.60, pp.37-49, 2002.
13. T. Cheng, M. Molenaar, & H. Lin. Formalizing objects from uncertain classification results.Int. J. Geographic information science, vol.15. no.1, p.27-42, 2001.