# Examination of Database Supported Spatio-Temporal Intersection Queries*

**Martin Breunig[2], Armin B. Cremers[1], Wolfgang Müller[1], Jörg Siebeck[1]**

[1]Institute of Computer Science III
University of Bonn
Römerstr. 164
53117 Bonn
Germany

e-mail: (abc,muellerw,siebeck)@cs.uni-bonn.de

[2]Institute of Environmental Sciences
University of Vechta
P.O. Box 1553
49364 Vechta
Germany

e-mail: mbreunig@iuw.uni-vechta.de

**Abstract.** The quickly increasing number of spatio-temporal applications in fields like environmental monitoring, geology and mobile communication is a new challenge to the development of geo-databases. However, the query functionality of today´s geo-information systems is still limited to the thematic attributes of spatial objects and to spatial 2D objects. In this paper we present two examples for the definition and database implementation of spatio-temporal intersection operations to be used in geo-scientific 3D/4D applications: the intersection of a spatio-temporal object with a given axis-aligned bounding box and the intersection of a spatio-temporal object with a user-defined 3D plane. Both operations are implemented within the GeoToolKit project. We then show the application of the two operations within two geo-scientific applications: the balanced restoration of the Lower Rhine Basin and the generation of artificial, time-dependent landform data in 3D space, respectively. Finally, we give an outlook on our future research work on spatio-temporal database operations.

## Introduction

The requirements for the modelling of time are manifold in geoscientific applications. They are covering time conceptions like the modelling of discrete time stamps or dynamic geo-processes. The fact that entities of the dynamic environment may continuously change both location and shape makes their modelling a complex task.

Similarly, the management of such time-dependent geometries imposes challenges on the database community (Sellis 1999). Indeed, database models must be capable of representing these entities adequately. At the same time, data analysis demands interactive query processing facilities ranging from selections for online animation with different levels of detail to complex set-based operations like spatio-temporal joins (Güting et al. 2000). On the way to working systems, one must carefully define and implement *object-based* operations, which are the pre-condition of spatio-temporal querying.

Defining these operations it is advantageous to apply an orthogonal design. This facilitates composition of operations. For instance, the snow cover modeller could be interested in the question when the snow cover below a given height *h* has been largest. This query could be answered by intersecting the snow cover – regarded as a temporal 3D surface – with the region below *h* and finding for the new temporal surface the times of largest area. Hence, only because the datatypes of the query input correspond to the datatypes of the spatio-temporal operations, the results of database queries can again be used as input to further querying.

In this paper, we define and demonstrate the processing of two intersection-operations, each receiving one spatio-temporal and one purely spatial object as input. The first operation enables the user to cut out a part of a spatio-temporal object that is contained in a given bounding box at a given temporal interval. The second operation enables the user to specify a 3D plane, which a spatio-temporal object is intersected with. From a user's point of view, both operations aim at restricting spatio-temporal objects to the "interesting" subregions of $\mathbb{R}^3$ and the temporal domain. They support therefore a closer object-inspection and, in particular, object-animation. To evaluate the approaches taken, two different scenarios are described. The first one stems from the cooperation with a group of geologists, whereas the second one demands the processing of artificially generated landform data. The techniques for generating this artificial data is described in detail.

* This work has been supported by the German Research Foundation (DFG) within the collaborative research centre "SFB 350" and the postgraduate school "Landform" at Bonn University.

## Conceptual Spatial and Spatio-Temporal Model

The work presented in this paper is embedded into the "GeoToolKit"-projects. GeoToolKit (Balovnev et al. 1997, Balovnev et al. 1998, Balovnev et al. 1999, Shumilov and Siebeck 2001) is a collection of software tools intended to facilitate the development of 3D/4D geo-applications, especially those having to deal with the management of large amounts of spatial and spatio-temporal data.

GeoToolKit is not realized in the fashion of a GIS-in-a-box package. Instead, it is a component toolkit that allows the incorporation of spatio-temporal functionality within an application and encourages the development and deployment of re-usable and open software. It not only enables the management of abstract geometric primitives, such as points, curves, and surfaces, but also of real world entities such as drilling wells, geological sections and strata. The latter entities are particularly advanced by the object-oriented modelling technique. Applications can simply inherit or aggregate as much geometric functionality from GeoToolKit as needed, extending it with the application-specific semantics.

### Conceptual Spatial Model

GeoToolKit offers data types for the representation and manipulation of geometries in three dimensional Euclidean space. The basic notion of our conceptual model of spatial data is that of simplicial complexes (Egenhofer et al. 1990). In particular, the core set of spatial types comprises 0D-3D simplices (points, segments, triangles, and tetrahedra) and 1D-3D simplicial complexes (polylines, triangular networks, and tetrahedral networks). Furthermore, *gtGroup* is the spatial type which allows for aggregating instances of spatial types. All spatial types support geometric operations, functions, and predicates.

Although every 0D-3D simplicial complex can be represented within GeoToolKit, there are the "first class citizens" polyline, triangular network, and tetrahedral network which are subject to two restrictions. First, a triangular network contains only triangles and their faces, even though the above definition also allows of non-facial segments or "isolated" vertices. Secondly, at most two triangles share a one-dimensional face (segment). Polylines and tetrahedral networks have the corresponding restrictions. More general topologies, e.g. non-manifolds, may be represented through type *gtGroup*. These restrictions have so far been in compliance with the requirements of our applications (Balovnev et al. 1998).

By defining the complex types such that the simplices of a complex only hold references to their vertex-data, we separated topological from geometric information. This enables different meshes of the same set of vertices. Equally important, it yields non-redundancy.

In a spatio-temporal setting, we regard pure-spatial objects as entities not related to any specific time at all, and we therefore let them implicitly carry the temporal interval $[-\infty, +\infty]$. In such way, these objects become *time-independent*.

### Conceptual Spatio-Temporal Model

To provide the appropriate maintenance of spatio-temporal objects needed for geoscientific applications, GeoToolKit's spatial entities have been extended by the concept of time. In the following we describe our temporal extensions to the pure spatial types (cf. our previous work Shumilov and Siebeck 2001).

The spatio-temporal model has been designed in analogy to the pure-spatial model. Recall that we separated geometry (location of vertices) from topology (how to mesh the vertices). Let us transfer this design principle to the 4D case. Thus, we will start our description by considering the model of vertex-movement, followed by temporal simplices and temporal complexes. Beforehand, though, we define concepts from the time-domain.

Time is interpreted as being isomorphic to the set of real numbers, which from now on we will use as the time domain. One element of this domain is called an *instant*. A temporal interval $I$ is given by two instants $t$, $t'$, where $t \leq t'$, permitting degenerate intervals [t, t]. Furthermore, there is the special temporal interval *empty*. A temporal interval may be open on one or both sides. We denote the closure of a temporal interval $I$ by $\overline{I}$ and the boundary of $I$ by $\partial I = \{\inf I, \sup I\}$. A *temporal element* is defined as a set $E$ of non-overlapping temporal intervals.

The location of a vertex $v$ is regarded as a function of time. This location is given by

$$loc_v\colon \mathbb{R} \rightarrow \mathbb{R}^3 \qquad \text{such that } loc_v \text{ is continuous.}$$

For simplicity, we identify $loc_v$ with $v$ itself by writing $v(t)$. The *trajectory* of $v$ is defined as the 3D curve

$$traj(v) = \{\, loc_v(t) \mid t \in \mathrm{def}(loc_v) \,\}$$

Currently, our conceptual model limits the trajectory to be piecewise linear. Therefore, for each trajectory $T$ there is an ordered set of points $\{p_1,...,p_n\}$ such that $T$ can be calculated through the linear interpolation between two consecutive points $p_i$ and $p_{i+1}$, $1 \leq i \leq n$-1.

Furthermore, we also use linear interpolation concerning time. This means that if vertex $v$ is at location $p_i$ at time $t_i$ and at location $p_{i+1}$ at time $t_{i+1}$, then the *direction* of the velocity vector of $v$ is that of vector $p_{i+1}-p_i$. The velocity vector itself is obtained by dividing by $t_{i+1}-t_i$. The location of $v$ at instant $t$, $t_i < t < t_{i+1}$, is then obtained by

$$v(t) = p \;+ \frac{t - t_i}{t_{i+1} - t_i}\left(p_{i+1} - p_i\right), \quad t \in \left(t_i : t_{i+1}\right)$$

This implies constant speed (zero velocity) between $p_i$ and $p_{i+1}$.

To summarize, in our model the function $loc_v$ is given by a sequence of points $S$ ordered by the time-value time($v_i$) assigned to each point $p_i$ in $S$. We call these points the *support points* of $v$. Of course, points with equal coordinates may be present in $S$, but no two points may have the same time-value. Some points in $S$ mark changes in *direction*, some mark changes only in *velocity*, and some mark both. No other restrictions, for example physical restrictions, apply to the points in $S$ except for those mentioned. Restrictions have to be imposed at the application level. In our model, the data type representing function $loc_v$ is called a *moving vertex*. We define the time-steps of a moving vertex $v$ as the set of time values assigned to the support points of $v$.

A temporal $d$-simplex $s$ references $d+1$ moving vertices $v_1,..., v_{d+1}$. Furthermore, $s$ is assigned a temporal interval $I$ of validity. Then, for each $t$ in $I$, a $d$-simplex is defined to be the convex hull conv($loc_{v_1}(t)$, ..., $loc_{v_{d+1}}(t)$). In our model, the data type representing such temporal simplices is called a *temporal simplex*.

For reasons of integrity, the referenced vertices must be valid during $I$. Furthermore, for all $t$ in $I$ the points $loc_{v_1}(t)$, ..., $loc_{v_{d+1}}(t)$ have to be affine independent. Hence, no degenerate segments, no triangles without area, and no flat tetrahedra are allowed.

In our model, a temporal simplicial complex $C$ is composed of a set of temporal simplices and a temporal interval $I$ of validity. In parallel to the 3D case, $C$ is subject to the following restriction: for all $t$ in $I$ the complex should suffer no self-intersection.

Note that our model allows the geometry of a complex to change smoothly over time[1]. Nevertheless, discrete changes to the geometry are not ruled out. Furthermore, changes to the external topology can also be represented.

## Intersection-Queries for Spatio-Temporal Objects

The conceptual model presented in the previous section will prove useful only if it is accompanied by retrieval facilities. In our previous work we focused on efficient update operations as well as on efficient time-step and time-interval queries on spatio-temporal objects (Shumilov and Siebeck 2001).

Here, we go one step further and derive two intersection-operations. The first operation enables the user to cut out a part of a spatio-temporal object that is contained in a given bounding box at a given temporal interval. The second operation enables the user to specify a 3D plane, which a spatio-temporal object is intersected with. From a user's point of view, both operations aim at restricting spatio-temporal objects to the "interesting" subregions of $\mathbb{R}^3$ and the temporal domain. They support therefore a closer object-inspection and, in particular, object-animation.

From a technical point of view, several advantages arise. Integrating these operations into the database kernel prevents client applications from loading parts of objects not contained in the current region of interest (both spatial and temporal), such that the overall system performance is likely to benefit. Furthermore, by an orthogonal design the operations' results can themselves be used as input to further querying. The reason for this is that the result is itself a spatio-temporal object. In the remainder of this section we define and demonstrate the processing of these operations.

### Intersection with a Bounding-Box

The operation of intersecting a spatio-temporal object with a given axis-aligned bounding box comes in two flavours. First, it can be defined as creating a subset of a spatio-temporal object (as explained below). Secondly, it can be defined such that simplices not completely contained in the bounding box are to be re-triangulated.

[1] This is, of course, in contrast to the discretization which does not change continuously.

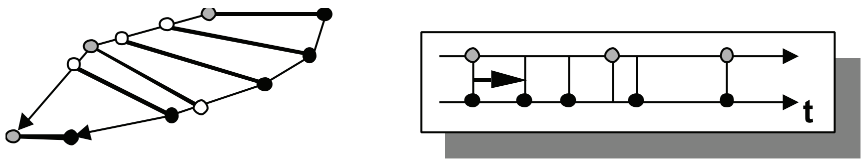


*Figure 1: Left: Two moving vertices define a temporal segment s.*

*Right: Decomposition of time(s) into temporal intervals.*

*Intersection without Re-Triangulation*

This operation can also be regarded as a spatio-temporal *range query* on a single spatio-temporal object. Conceptually, given a bounding box *b*, a temporal interval *I*, and a temporal simplicial complex (as a set *C* of temporal simplices), the operation returns a set *C'* of temporal simplices defined as follows.

$$C' = \bigcup_{s \in C} \text{intersection}(s,b,I)$$

Here, intersection(*s,b,I*) denotes the set *S* of temporal simplices such that for each simplex *s'* in *S* the following propositions hold: (1) *time*(*s*') is a subset of *I*; (2) *s*' references vertices(*s*); and (3) *time*(*s*') corresponds to a maximal time interval during which *s* intersects *b*. Note that this operation does not generate new point data; instead, newly created simplices only reference existing moving vertices.

The computation of *C'* is performed by breaking it down to the simplex-level. Additionaly, the movement of the simplices is decomposed into temporal intervals, during which the location of each referenced vertex can be interpolated (cf. figure 1). This decomposition is governed by the support points of the referenced vertices. To return a spatio-temporal object of our model, an implementation must further establish topological information (see above). Details of the computation will be omitted here.

*Intersection with Re-Triangulation*

This operation is similar to the previous one; however, for time intervals, in which a simplex is not completely contained in the given bounding box, this simplex must be re-triangulated. During this process, new moving vertices are generated. They stem from the intersection of a simplex with the box's boundary; however, the trajectory of such a new vertex can be non-linear as the following result shows.

***Proposition 1****. Non-rigid object-motion based on linear vertex-movement is not closed under intersection.*

***Proof:*** This proposition can be seen best by an example. Assume, we are given a temporal segment referencing two moving vertices. Let the first vertex perform a linear move from the origin to, say, (1,1,0). Let the second vertex perform a linear move from (0,1,1) to (1,0,1). The trajectory of the so defined moving segment is a *bilinear interpolation* of the aforementioned points: a curved surface. Therefore, intersecting the temporal segment with a static triangle can result in a moving vertex trajectory that is non-linear. Indeed, only if the triangle is placed such that it lies within a plane parallel to either *xy*- or *yz*-plane, a linear vertex movement results; however, all other placements would result in non-linear movement. Hence, the proposition holds. □

To nevertheless create a temporal complex by intersection, the trajectory must therefore be approximated piecewise linear. One of our future aims is to also implement this operation.

**Intersection with a 3D Plane**

Intersecting a time-dependent geometry with a user-defined 3D plane allows a two dimensional investigation of the geometry's temporal evolution. While for animation purposes it suffices to compute such an intersection on a per-time-step basis, this approach falls short of orthogonality. Hence, we derive an operation that results in a temporal complex.

Of course, proposition 1 (preceding section) also applies to this intersection operation. Alternatively, we chose a variant of this operation. Parallel to the first bounding box intersection variant, this operation does not create any new vertices. Instead, given a temporal complex *C*, a temporal interval *I*, and a 3D plane *p* it returns a subset of temporal simplices *C'*, in which each simplex intersects *p* during its whole lifetime:

$$C' = \bigcup_{s \in C} \text{intersection}(s,p,I)$$

The complete description of the operation will be omitted here.

## Applications

We have tested the temporal extension of GeoToolKit within two different scenarios. The first one stems from a cooperation with geologists, in which one of the objectives was to examine the balanced restoration of structural Rhine Basin evolution (Thomsen et al. 2000). The second one demands the processing of artificial landform data. In the following we describe both scenarios in more detail.

### Balanced Restoration of the Lower Rhine Basin

We have tested the temporal extension of GeoToolKit with a geological model consisting of stratigraphic and fault surfaces extended by a time attribute. The animated surfaces of that model were visualized in Gocad (Mallet 1992) and sent to GeoToolKit for spatio-temporal database querying.

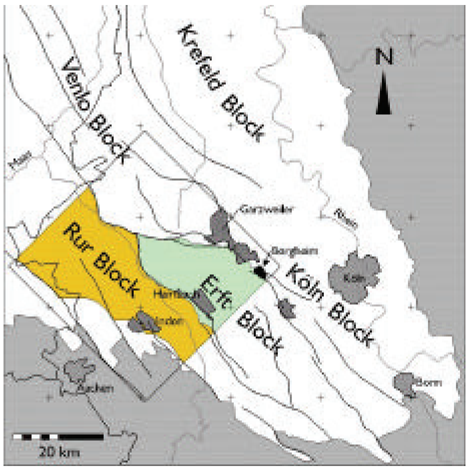


*Figure 2: Position of the Bergheim mine (arrow) in the Lower Rhine Embayment.*

One of the objectives of the geological model was to examine the balanced restoration of structural Rhine Basin evolution (Thomsen et al. 2000). This geological process started in tertiary period, i.e. about 65 million years ago. The exact geological movements, however, are unknown. Therefore different possible variants for the time sequences have to be examined by the geologists.

The region under consideration is the area of an open pit lignite mine "Bergheim" which has a diameter of about 4km and a relative depth of about 500 m (see figure 2). The Bergheim mine has been in operation by Rheinbraun AG, Köln, since 1984 with an annual output of 15.7 Mill. tons (1996/97) of lignite.

The originally static 3D-model on the basis of approx. 20 digitized profile sections of Rheinbraun AG was prepared for the movements. The class library Grape (Polthier, Rumpf 1994) and other custom programs were used for the calculation of the movement and Gocad (Mallet 1992) for the visualization of geological objects.

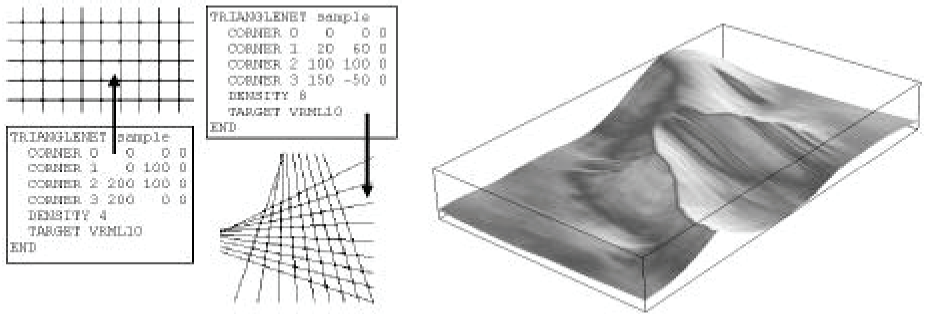
### Generation of Artificial, Time-Dependent Landform Data in 3D Space

Within geo-database research an agreed-upon benchmark data-set exists for the 2D case (the so-called Tiger/Line data-set from the U.S. Bureau of the Census). Unfortunately, there seems to be no comparable 3D or even 4D data-set. Hence, to evaluate the approaches taken, one must either rely on small and hardly obtainable data-sets (cf previous section) or use artificial data. In the following we describe an example for the latter method that we implemented.

To obviate the need to generate data interactively, we tried to automate this process. It works in four stages. First, an initial (artificial) part of the earth's surface is generated. Second, this surface is subjected to change, resulting in a sequence of time-steps. Third, for each such time-step so-called form elements are derived. These are regions of homogeneous parameter distribution in the meso-scale (Dikau 1989). Fourth, the temporal development of these form elements are tracked over the different time-steps, resulting in moving and changing landform objects. In the following, we describe each stage in more detail.

*Generating an Initial Surface*

The first stage is based on a *generic* generation of a surface, i.e. the location and the form of the surface are controlled by few parameters. The objective is to non-interactively generate triangulated surfaces with naturally shaped geometries. The input then is a coarse textual description of geometrical features to randomly produce "flat ground reliefs" as well as "mountain ranges."



```
TRIANGLENET sample
   CORNER 0    0    0 0
   CORNER 1    0  100 0
   CORNER 2  200  100 0
   CORNER 3  200    0 0
   DENSITY 4
   TARGET VRML10
END
```

```
TRIANGLENET sample
   CORNER 0    0    0 0
   CORNER 1   20   60 0
   CORNER 2  100  100 0
   CORNER 3  150  -50 0
   DENSITY 8
   TARGET VRML10
END
```

*Figure 3: Left: Examples for basic grids and their definitions. Right: Artificially generated initial surface with 75000 triangles*

The right part of figure 3 shows an example of an artificial surface containing 75000 triangles constructed by GeoSurf.

*Changing the Initial Surface*

This stage is performed by a simple diffusion process. In geoscientific modelling this approach is often taken in *one* dimension on a *small* scale, e.g., a slope profile. Applying this process in our setting results in a series of time-steps, altering the initial surface.

*Deriving Form Elements*

For each time-step, so-called form elements are derived. Form elements are areas with a homogeneous internal distribution of primary landform parameters. Dikau has given a system for the meso-scale, in which such a classification of landform units is based on the parameters slope, exposition, and curvature (Dikau 1989).

*Constructing Temporal Complexes*

To construct a temporal complex the final stage keeps track of the initial surface of a form element over different time-steps.

## Conclusions and Outlook

In this paper we have presented two examples for the definition and implementation of spatio-temporal database operations to be used in geo-scientific 3D/4D applications: the intersection of a spatio-temporal object with a given axis-aligned bounding box and the intersection of a spatio-temporal object with a user-defined 3D plane. Both operations have been implemented within the GeoToolKit project. We then showed the application of the two operations within two geo-scientific applications: the balanced restoration of the Lower Rhine Basin and the generation of artificial, time-dependent landform data in 3D space, respectively.

In our future work, we intend to define and implement more spatio-temporal database operations as building blocks of a spatio-temporal algebra for time-dependent geometric and topological objects. These operations could then be used in geo-scientific or in other geo- applications like mobile information systems.

## Acknowledgements

We thank the groups of Agemar Siehl (Geological Institute) and Richard Dikau (Geographical Institute) of Bonn University for many discussions on geoscientific problems. Without their suggestions our conceptual and implementation work would have been less successful. Our special thanks go to Andreas Thomsen for his help by analysing the Bergheim open pit mine application.

## Bibliography

Balovnev, O., Breunig, M., Cremers, A. B., Pant, M., 1998: Building geo-scientific applications on top of geotoolkit: a case study of data integration. In: *Proc. of the 10th International Conference on Scientific and Statistical Database Management*, pp. 260-269.

Balovnev, O., Breunig, M., Cremers, A.B., Shumilov, S., 1999: Space-time modelling of the lower rhine basin supported by an object-oriented database. In: M. Goodchild, M. Egenhofer, R. Fegeas, C. Kottman (Eds.): *Interoperating Geographic Information Systems.*, Kluwer Academic Publishers.

Balovnev, O., Breunig, M., Cremers, A.B., 1997: From GeoStore to GeoToolKit: The Second Step. In: *LNCS No. 1262*, Springer, Heidelberg, pp. 223-237.

Dikau, R., 1989: The Application of a Digital Relief Model to Landform Analysis in Geomorphology, In: Raper, J. (Ed.). *Three Dimensional Applications in Geographical Information Systems*, pp. 51-77, Taylor & Francis.

Egenhofer, M. J., Frank, A. U., Jackson, J. P., 1990: A topological data model for spatial databases. In: *Proceedings of the 1st Symposium SSD on Design and Implementation of Large Spatial Databases.* A. Buchmann, Günther, O., Smith, T.R., Wang, Y.-F. (Eds.). LNCS No. 409, Springer, pp. 271-286.

Güting, R. H., Bohlen, M. H., Erwig, M., Jensen, C. S., Lorentzos, N. A., Schneider, M., Vazirgiannis, M., 2000: A foundation for representing and querying moving objects. In: *ACM Transactions on Database Systems* 25, 1, pp. 1-42.

Mallet, J.-L., 1992: GOCAD: A Computer Aided Design Program for Geological Applications. In: Turner, A.K. (Ed.), *Three-Dimensional Modeling with Geoscientific Information Systems,* NATO ASI 354, Kluwer Academic Publishers, Dordrecht, pp. 123-142.

Polthier, K., Rumpf, M., 1994: A concept for Time-Dependent Processes. In: Goebel et al., (Eds), *Visualization in Scientific Computing,* Springer, Vienna, pp. 137-153.

Sellis, T., 1999: Research Issues in Spatio-temporal Database Systems. In: Güting, R.H., Papadias, D., Lochovsky, F. (eds.): *Advances in Spatial Databases*. Lecture Notes in Computer Science 1651. Springer-Verlag

Shumilov, S., Siebeck, J., 2001: Database Support for Temporal 3D Data: Extending the GeoToolKit. In: *Proc. of the 7th EC-GI & GIS Workshop.* Potsdam, Germany.

Thomsen, A., Jentzsch, T., Siehl, A., 2000: Towards a balanced kinematic model of a faulted domain in the lower rhine graben. In: *Proc. of the GOCAD ENSG Conference: 3D Modeling of Natural Objects – A Challenge for the 2000's.*