Efficient Querying and Animation of Periodic Spatio-Temporal Databases *

Peter Revesz [a] Mengchu Cai [b]

[a] *Department of Computer Science and Engineering*
University of Nebraska–Lincoln, Lincoln, NE 68588, USA
E-mail: revesz@cse.unl.edu
[b] *IBM Silicon Valley Lab. San Jose, CA 95141, USA*
E-mail: mcai@us.ibm.com

We propose a representation of spatio-temporal objects with continuous and cyclic or acyclic periodic movements. We also describe an extended relational algebra query language for databases with such objects. We show that the new spatio-temporal databases are closed under the extended relational algebra queries, and each fixed relational algebra query can be evaluated in PTIME in the size of the input database.

Keywords: Spatio-Temporal databases, animation, data model, relational algebra

AMS Subject classification: H.2.1;H.2.3

1. Introduction

Many spatio-temporal objects such as clouds, cars, deserts, lakes, planets, ships and tornados change position or shape continuously and also sometimes periodically. Although in the last decade substantial research was done independently in spatial [26,36] and temporal [31] data modeling, continuously and periodically changing objects require new data models that can capture the interdependency of the spatial and temporal extents of these objects.

We introduce a new model for continuously and possibly periodically moving, growing or shrinking objects. The intuition for our model is that at any time instance the shape of a moving, growing or shrinking planar object can be

* This work was supported in part by NSF grants IRI-9625055 and IRI-9632871 and by a Gallup Research Professorship.

approximated by the union of a set of rectangles whose sides are parallel to the axes. Further, as the object changes the rectangles also change. Such changing rectangles can be described as rectangles whose x and y dimensions are functions of time. Therefore, we call such rectangles *parametric rectangles*. We can also generalize from 2 to higher dimensional parametric rectangles (or rectangular bodies).

While the shape of objects can be approximated by other ways, for example by general polygons, the simplicity of the rectangular representation has several clear advantages. For example, we might also try to extend with a time parameter general polygons. However, such objects are not closed under intersection (for linear functions of time) [14], while they are closed for rectangles as we showed earlier [10]. In this paper we show the following additional advantages of parametric rectangles.

First, we show that parametric rectangles can be extended with both cyclic and acyclic periodic movements, forming *periodic* parametric rectangles.

Second, we show in Theorem 23 that any fixed relational algebra expression can be evaluated in PTIME in the size of the input periodic parametric rectangle database.

Third, an advantage of our model is the combination of efficient querying with efficient *animation*, that is, the display of the instantiation of the relations in the database at successive time instants. Most spatio-temporal data models have difficulty combining effectively querying with animation. We implemented a system –PReSTO (short for *Parametric Rectangle Spatio-Temporal Objects*)– that proves that the combination is effective in practice as well as theory.

The paper is structured as follows. Section 2 describes the periodic parametric rectangle data model and illustrates how to represent spatio-temporal objects in this model. Section 3 extends relational algebra to spatio-temporal objects and proves that the evaluation of queries is in PTIME in the size of the input database. Section 4 discusses the implementation of the PReSTO system and presents run time results. Section 5 covers some additional related work. Finally, Section 6 lists some directions for further work.

2. Representation of Spatio-Temporal Objects

We first define non-periodic parametric rectangles in Section 2.1, then periodic parametric rectangles in Section 2.2.

2.1. Parametric Rectangles

Let **R** denote the set of real numbers and **N** the set of natural numbers. Below we give a formal definition of parametric rectangles.

Definition 1. A d-dimensional *parametric rectangle* r is a tuple of the form:

$$\langle X_1, \ldots, X_d,\ T \rangle$$

where T is a real interval represented as $[t_1,\ t_2]$, $t_1, t_2 \in \mathbf{R}$, and each X_i for $1 \leq i \leq d$ is a function from R to real intervals. We represent X_i simply as $[X_i^{[}, X_i^{]}]$ where $X_i^{[}$ and $X_i^{]}$ are functions from $\mathbf{R} \rightarrow \mathbf{R}$.

Intuitively, T represents the time instances within a real interval when the parametric rectangle exists and each X_i represents the spatial extent of the parametric rectangle in the ith dimension. That is, the function $X_i(t)$ gives the ith axis dimension of the parametric rectangle for each instance of time t within T. Therefore, the semantics of r, denoted by $sem(r)$, is a possibly infinite set of points defined as follows:

$$sem(r) = \{\ (a_1, \ldots, a_d, t) \mid a_1 \in X_1(t), \ldots, a_d \in X_d(t),\ t \in T.\}$$

Example 2. The semantics of the parametric rectangle $r = \langle [5-t, 10+t],\ [4-t, 6+t],\ [0,3] \rangle$, is the polyhedron in x, y and t dimensions as shown in Figure 1.

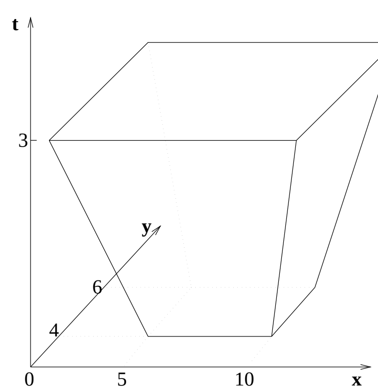


Figure 1. Semantics of the Parametric Rectangle

Example 3. Suppose that a sail boat which at time $t = 0$ occupies the space $9 \leq x \leq 19$, $10 \leq y \leq 20$ first moves east with a speed of 5 ft/sec until $t = 10$.

Then it goes northeast until $t = 20$, with a speed of 10 ft/sec in both the x and y axes. Finally, it goes north with a speed of 8 ft/sec until $t = 25$. We can represent the sail boat by 3 parametric rectangles, as shown in Table 1.

X	Y	T
$[5t + 9,\ 5t + 19]$	$[10,\ 20]$	$[0,\ 10]$
$[10t - 41,\ 10t - 31]$	$[10t - 90,\ 10t - 80]$	$[10,\ 20]$
$[159,\ 169]$	$[8t - 50,\ 8t - 40]$	$[20,\ 25]$

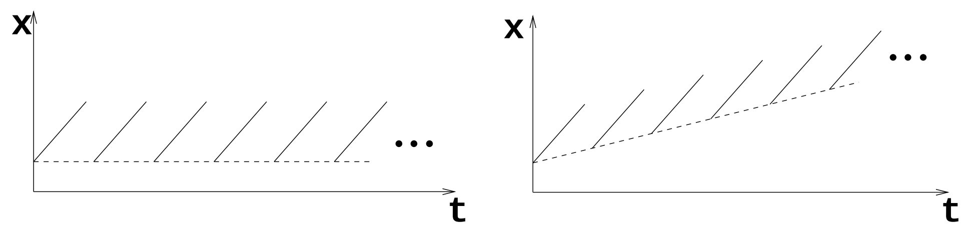
Table 1
Parametric rectangles for the sail boat

We call m-degree those parametric rectangles in which the bounds are at most m-degree polynomial functions of time. We also call $m = 1$ and $m = 2$ degree parametric rectangles linear and quadratic, respectively. For example, Table 1 contains only linear parametric rectangles.

2.2. Periodic Parametric Rectangles

Some spatio-temporal objects can also move periodically, for example, shuttle buses and planets in the solar system. It is not possible to finitely represent these periodic objects by the parametric rectangles that we have discussed so far. Hence we extend the parametric rectangle concept to *periodic parametric rectangle*.

We distinguish between *cyclic* and *acyclic* periodic movements. *Cyclic periodic movement* occurs when an object repeats its movement from the same position and with the same velocity every period as shown in Figure 2(a). For example, the pendulum of a stationary pendulum clock moves in a cyclic movement. Cyclic movement can be typically approximated by a function of the form $f(t \ mod \ p)$, where p is the period. *Acyclic periodic movement* is the composition of cyclic periodic movement and linear movement as shown in Figure 2(b), which can be represented by $g(t) + h(t)$, where $g(t)$ is a cyclic movement function and $h(t)$ is a linear function. For example, the pendulum of a pendulum clock on a ship moving with a uniform speed can be described as an acyclic periodic movement, with a cyclic component that is due to the pendulum only and a linear component that is due to the movement of the ship.



(a) cyclic periodic movement (b) acyclic periodic movement

Figure 2. Cyclic and acyclic periodic movements

Before giving the formal definition of periodic parametric rectangles, we first need to extend the modulus operator to real numbers as follows:

$$a \ mod \ p = a - p \times \lfloor \frac{a}{p} \rfloor$$

where $a \in \mathbf{R}$ and $p \in \mathbf{N}$.

Definition 4. A periodic parametric rectangle $r = \langle X_1, \ldots, X_d, T \rangle$ is a parametric rectangle such that $X_i = [X_i^{[}, X_i^{]}]$ for each $i = 1, \ldots, d$ where $X_i^{[}$ and $X_i^{]}$ are linear periodic functions of the form:

$$a((t - from) \ mod \ p) + b + c \lfloor \frac{t - from}{p} \rfloor \tag{1}$$

where a, b and c are constant real numbers, and T is:

$$[from, \ to]_{p,end} = \{t \mid t \in \mathbf{R}, \ from \leq t \leq end, \ (t - from) \ mod \ p \leq to\} \tag{2}$$

where $p \in \mathbf{N}$ is the period, and $from, to \in \mathbf{R}$ such that $0 \leq (to - from) \leq p$ are the lower and upper bounds of the time interval in the first period and $end \in \mathbf{R} \cup +\infty$ is the ending time of the object.

For simplicity, we allow the special case of $p = +\infty$ and $end = to$ to include linear parametric rectangles that are *non-periodic*. In this case, formula (1) simplifies to $a(t - from) + b$ and formula (2) simplifies to $T = [from, \ to]$. We also call $s = \frac{c}{p}$ for $p \neq +\infty$ the *slope* of the lower (or upper) bound start line as shown in Figure 3.

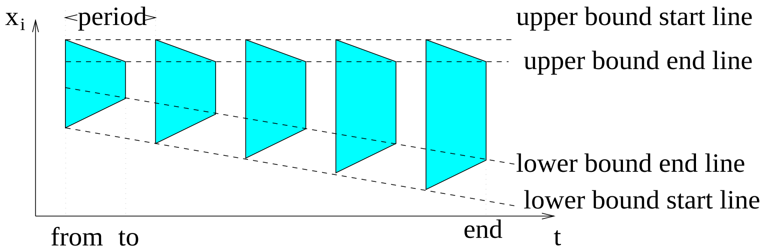


Figure 3. The projection of a PPR onto $(x_i,\ t)$ space

The extent of a periodic parametric rectangle (PPR) in each dimension is represented by an interval whose lower and upper bounds are periodic functions of time. Figure 3 shows that the position and extent of a PPR in x_i dimension change over time. In $(x_i,\ t)$ space, the positions of the lower bound $X_i^{[}$ at the start times (or end times) of different periods are located on a line, which is called the lower bound start (or end) line. The belt between the lower bound start and end lines is called the lower bound belt. Similarly the positions of the upper bound $X_i^{]}$ at the start (or end) times of different periods are located on the upper bound start (or end) line, and the belt between them is called the upper bound belt.

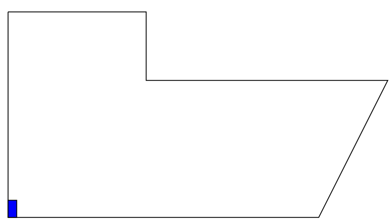


Figure 4. Route of the shuttle bus

Example 5. Suppose there is a shuttle bus running every 30 minutes around a route as shown in Figure 4. We can represent it by six periodic parametric rectangles, in a relation *shuttle*, as shown in Table 2.

Example 6. Suppose that a person A is swimming in a wavy sea. Figure 5 shows the positions of A at four different time instances.

X	Y	T
$[0,\ 1]$	$[5(t\ mod\ 30),\ 5(t\ mod\ 30)+1]$	$[0,\ 6]_{30,+\infty}$
$[f_1(t),\ f_1(t)+1]$	$[30,\ 31]$	$[6,\ 10]_{30,+\infty}$
$[24,\ 25]$	$[f_2(t),\ f_2(t)+1]$	$[10,\ 12]_{30,+\infty}$
$[f_3(t),\ f_3(t)+1]$	$[20,\ 21]$	$[12,\ 18]_{30,+\infty}$
$[f_4(t),\ f_4(t)+1]$	$[f_5(t),\ f_5(t)+1]$	$[18,\ 22]_{30,+\infty}$
$[f_6(t),\ f_6(t)+1]$	$[0,\ 1]$	$[22,\ 30]_{30,+\infty}$

$$\begin{aligned} \text{where } f_1(t) &= 6((t-6)\ mod\ 30) \\ f_2(t) &= -5((t-10)\ mod\ 30)+30 \\ f_3(t) &= 6((t-12)\ mod\ 30)+24 \\ f_4(t) &= -3((t-18)\ mod\ 30)+60 \\ f_5(t) &= -5((t-18)\ mod\ 30)+20 \\ f_6(t) &= -6((t-22)\ mod\ 30)+48 \end{aligned}$$

Table 2
Representation of the *shuttle* relation

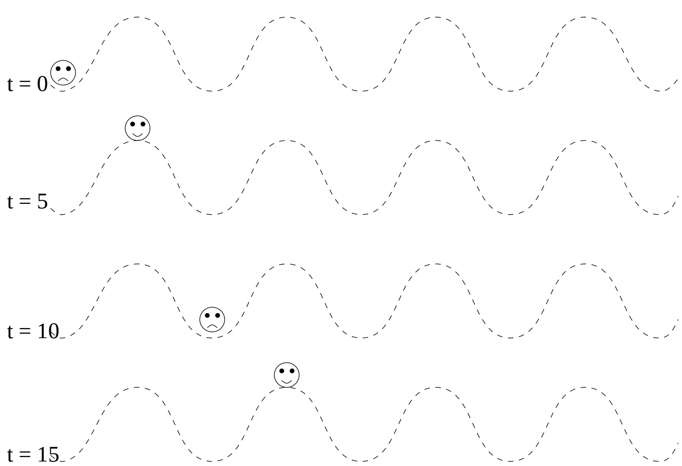


Figure 5. Movement of swimmer A

We represent the head of the person A as one unit in the height (z) and the swimming direction (x). We approximate A's movement as follows: in every period A first moves right in x with a speed of 1 unit per second for 2 seconds, then A goes towards the top of the wave with a speed of 2 units per second up in z and $\frac{1}{2}$ unit per second right in x, three seconds later A gets to the top, then

A moves right in x with a speed of 1 unit per second for 2 seconds, finally he falls down to the bottom with a speed of 4 units per second down and 2 unit per second right. Hence the movement of A can be decomposed into four periodic pieces and represented as PPRs as shown in Table 3.

X	Z	T
$[f_1(t),\ f_1(t)+1]$	$[0,\ 1]$	$[0,\ 2]_{10,+\infty}$
$[f_2(t),\ f_2(t)+1]$	$[2(t-2)\ mod\ 10,\ 2(t-2)\ mod\ 10+1]$	$[2,\ 6]_{10,+\infty}$
$[f_3(t),\ f_3(t)+1]$	$[9,\ 10]$	$[6,\ 8]_{10,+\infty}$
$[f_4(t),\ f_4(t)+1]$	$[4((t-8)\ mod\ 10),\ 4((t-8)\ mod\ 10)+1]$	$[8,\ 10]_{10,+\infty}$

where $f_1(t) = (t\ mod\ 10) + 10\lfloor \frac{t}{10} \rfloor$,
$f_2(t) = \frac{1}{2}((t-2)\ mod\ 10) + 2 + 10\lfloor \frac{t-2}{10} \rfloor$,
$f_3(t) = ((t-6)\ mod\ 10) + 10\lfloor \frac{t-6}{10} \rfloor$,
$f_4(t) = 2((t-8)\ mod\ 10) + 10\lfloor \frac{t-8}{10} \rfloor$.

Table 3
Representation of the *swimmer* relation

3. Querying Periodic Parametric Rectangle Databases

Let $\mathcal{P}$ denote the set of all d-dimensional parametric rectangles. Let $A_1, \ldots, A_k$ be a set of attributes, for each A_i there is an attribute domain $Dom(A_i)$ associated with it. A *parametric rectangle tuple* r on $A_1, \ldots, A_k$ is a tuple in $\mathcal{P} \times Dom(A_1) \times \ldots \times Dom(A_k)$.

The semantics of a parametric rectangle tuple $r = \langle r_1, a_1, \ldots, a_k \rangle$ is the cross product of the semantics of the parametric rectangle r_1 with the values $\{a_1\}, \ldots, \{a_k\}$.

A *parametric rectangle relation* is a finite set R of parametric rectangle tuples. The semantics of R is the union of the semantics of each tuple in R. An instantiation of R at time t_1, denoted by $R(t_1)$ is the union of instantiations of all tuples in R at time t_1. A *parametric rectangle database* is a finite set of parametric rectangle relations. In the following, by relations we mean parametric rectangle relations.

3.1. An Extended Relational Algebra

In this section we extend relational algebra to the parametric rectangle data model.

Definition 7. Let R_1 and R_2 be two d-dimensional parametric rectangle relations over the same set of attributes $A_1, \ldots, A_k$.

- *projection* ($\hat{\pi}_Y$) Let $\{t\} \subset Y \subseteq \{x_1, \ldots, x_d,\ t,\ A_1, \ldots, A_k\}$. The projection of R_1 on Y, denoted by $\hat{\pi}_Y(R_1)$, is a relation R over the attributes Y such that

$$R = \{r : \ \exists r_1 \in R_1, \ \forall A \in Y, \text{ the values of } A \text{ in } r \text{ and } r_1 \text{ are equal}\}$$

- *selection* ($\hat{\sigma}$) Let E be the conjunction of a set of comparison predicates of the form $A\,\theta\,B$, where $A \in \{x_1, \ldots, x_d, A_1, \ldots, A_k\}$, B is a constant or is in $\{A_1, \ldots, A_k\}$, and $\theta \in \{=, <, <=, >, >=\}$. The selection $\hat{\sigma}_E(R_1)$ is a relation R containing the parts of the parametric rectangle tuples in R_1 that satisfy E.
- *intersection* ($\hat{\cap}$) The intersection of R_1 and R_2, denoted by $R_1 \hat{\cap} R_2$, is a relation R over attributes $A_1, \ldots, A_k$ such that

$$sem(R) = sem(R_1) \cap sem(R_2)$$

- *union* ($\hat{\cup}$) The union of R_1 and R_2, denoted by $R_1 \hat{\cup} R_2$, is a relation R over attributes $A_1, \ldots, A_k$ that contains all tuples in R_1 and R_2.

$$sem(R) = sem(R_1) \cup sem(R_2)$$

- *difference* ($\hat{-}$) The difference of R_1 and R_2, denoted by $R_1 \hat{-} R_2$, is a relation R over attributes $A_1, \ldots, A_k$ such that

$$sem(R) = sem(R_1) \setminus sem(R_2)$$

- *complement* ($\hat{\neg}$) Let R be a relation with only the $x_1, \ldots, x_d, t$ attributes. The complement of R, denoted by $\hat{\neg} R$ is a parametric rectangle relation R', such that

$$sem(R') = \{(x_1, \ldots, x_d, t) : (x_1, \ldots, x_d, t) \notin sem(R)\}$$

The unary operators have higher precedence than the binary operators. intersection ($\hat{\cap}$) has higher precedence than union ($\hat{\cup}$) and difference ($\hat{-}$). A *rela-*

tional algebra expression over parametric rectangle databases is built up in the standard way, using the above operators.

Definition 7 defines the syntax and the semantics of the extended relational algebra operators. These largely follow standard relational algebra, but note that in the selection condition A and B cannot be both in $\{x_1, \ldots, x_d, t\}$. This restriction is needed to keep the spatial attributes independent of each other.

Each of the operators can be implemented by several possible algorithms. For example, Section 3.2 describes a particular implementation of $\hat{\cap}$ that satisfies $sem(R) = sem(R_1) \cap sem(R_2)$ for each input R_1 and R_2 periodic parametric rectangle relations and for which the output R is also a periodic parametric rectangle relation. Note that our algorithm may be substituted by another algorithm that returns a different output periodic parametric relation R' as long as $sem(R) = sem(R')$. Therefore, the semantic definition is a high-level language description that leaves open some implementation details. However, it is clear enough for users to write queries as we illustrate below.

Example 8. Consider the *shuttle* relation in Example 5. Let the relation *bus_stop* represent a bus-stop along the route of the shuttle bus. Suppose the relation *passenger* represents a man walking toward the bus-stop during some part of the day.

Query: "Will the passenger be able to catch the bus?"

$(shuttle \ \hat{\cap}\ bus\_stop)\ \hat{\cap}\ passenger$

Example 9. The nine planets of the solar system revolve around the sun in periodic orbits as shown in Figure 6. They are represented by 3D periodic parametric rectangle relations $Mercury$, $Venus$, …, $Pluto$. The motion of a comet is represented by the periodic relation *comet*.

Query: "Will the comet ever collide with any of the planets?"

$(Mercury\ \hat{\cup}\ Venus\ \hat{\cup}\ \ldots\ \hat{\cup}\ Pluto)\ \hat{\cap}\ comet)$

Example 10. Let us consider the swimmer example again. Suppose that swimmer A tries to find another person B. Suppose also that they can see each other only when they are both on the top of the waves and are within 20 meters of

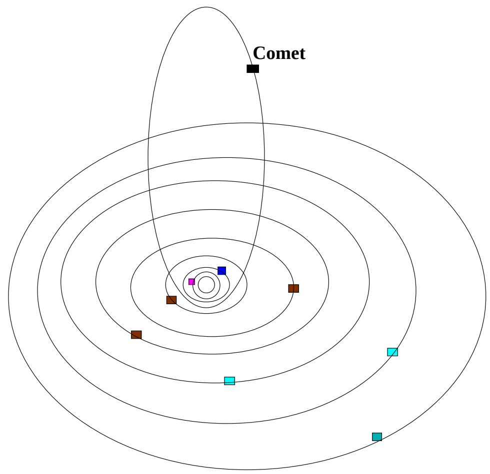


Figure 6. Orbits of the planets and comet

each other. A possible query of the database is:

Query: "Can A find B in 30 seconds?"

Suppose that the height of the wave is 8 meters. Let $RegionA$ and $RegionB$ be two PPR relations which represent the moving region within 10 meters of swimmers A and B, respectively. Then the query can be expressed as follows:

$$\hat{\sigma}_{z=8,\ t\leq 30}\ (RegionA\ \hat{\cap}\ RegionB)$$

Suppose now that a ship moves toward some direction in that region and A and B can be discovered if they get within 100 meters of the ship. Then a natural query is:

Query: "Can they be rescued by the ship?"

Suppose also that the PPR relation $Ship$ represents the region within 90 meters of the ship, then query can be expressed as follows:

$$Ship\ \hat{\cap}\ (RegionA\ \hat{\cup}\ RegionB)$$

Example 11. Another example from the area of physics is the following. Suppose that two photons travel in a reflecting path through an optical fiber cable as shown in Figure 7. A possible query is:

Query: "Will the two photons collide?"

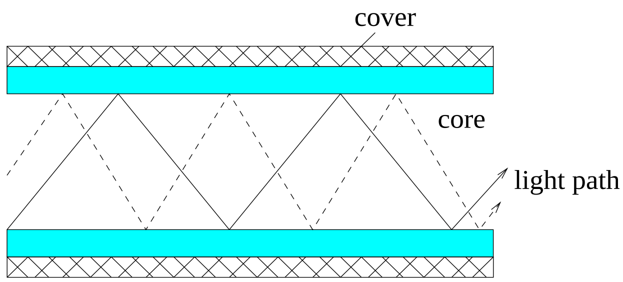


Figure 7. Photons travel through optical fiber

Suppose that photons traveling in an optical fiber are represented by a PPR relation *Photons*. We first select from *Photons* the photons with $id = 1$ and $id = 2$, where id is the identification number we assigned to the photons. Then we check whether their intersection is empty. This query can be expressed by the extended relation algebra as follows:

$$\hat{\Pi}_{x,y,t}(\hat{\sigma}_{id=1}Photons) \hat{\cap} \hat{\Pi}_{x,y,t}(\hat{\sigma}_{id=2}Photons)$$

3.2. Evaluation of Relational Algebra Queries

In this section, we prove that the relational algebra queries defined in the previous section can be evaluated in PTIME in the size of the input database in which all polynomial functions have at most a fixed m-degree. Theorem 15 deals with the non-periodic case, and Theorem 23 extends the result to the periodic case. We start with some definitions and lemmas about parametric functions.

Let f_1 and f_2 be two functions of time. We call "$f_1 <<_{[t1,t2]} f_2$" (read f_1 is not greater than f_2 in $[t1, t2]$) if for any time $t \in [t_1,\ t_2]$, $f_1(t) \leq f_2(t)$. We say there is a "$<<$" *order* of n functions $f_1, \ldots, f_n$ if for any f_i and f_j, either $f_i <<_{[t1,t2]} f_j$ or $f_j <<_{[t1,t2]} f_i$.

Lemma 12. Given n different m-degree polynomial functions of time $f_1(t), \ldots, f_n(t)$, we can find (precisely for $m < 5$, and approximately from $m \geq 5$) a partition of the time dimension into at most $N = O(mn^2)$ intervals $(-\infty, t_1), [t_1, t_2), \ldots, [t_{N-1}, +\infty)$ such that within each interval for any i and j, $0 \leq i,\ j \leq n$ and $i \neq j$, either $f_i << f_j$ or $f_j << f_i$.

Proof: Because any m-degree equation $f_i(t) = f_j(t)$ ($f_i \not\equiv f_j$) has at most m roots, there are at most $m \times \frac{n(n-1)}{2}$ intersections between the n curves corresponding to $f_1, \ldots, f_n$. Let $t_1, \ldots, t_{N-1}$ be the time instances corresponding to the intersections sorted in increasing order. These time instances can be found precisely for $m < 5$, and approximately to any desired precision for larger m. Let $t_0 = -\infty$ and $t_N = +\infty$.

Then the corresponding curves of the n functions can intersect in none of the $N \leq \frac{m}{2}(n^2 - n) = O(mn^2)$ open intervals (t_{k-1}, t_k) where $1 \leq k \leq N$. Therefore within $[t_{k-1},\ t_k]$, either $f_i << f_j$ or $f_j << f_i$. ■

Lemma 13. The intersection of two d-dimensional m-degree ($m < 5$) non-periodic parametric rectangle relations is a parametric rectangle relation which can be evaluated in PTIME in the size of the relations.

Proof: In this proof, we assume for simplicity that tuples have only spatial and temporal attributes. Let $r1 = (X_{1,1}, \ldots, X_{1,d}, T1)$ and $r2 = (X_{2,1}, \ldots, X_{2,d},\ T2)$ be two d-dimensional m-degree non-periodic parametric rectangles. Let $T = T1\ \cap\ T2$.

By Lemma 12, for each spatial dimension x_i, there is a set S_i of $O(m) = O(1)$ time instances that partition T into $O(1)$ intervals such that within each interval there is a "<<" order of the functions of the lower and upper bounds of $r1$ and $r2$ in x_i dimension.

Let $S = \bigcup_{i=1}^{d} S_i$. S partitions T into $N = O(d)$ intervals such that within each interval $(t_{j-1}, t_j]$ there is a "<<" order of the functions of the lower and upper bounds of $r1$ and $r2$ in any dimension.

For each time interval $(t_{j-1}, t_j]$, let

$$X_i^{[} = \begin{cases} X_{1,i}^{[} \text{ if } X_{2,i}^{[} << X_{1,i}^{[} \text{ in } [t_{j-1},\ t_j] \\ \\ X_{2,i}^{[} \text{ otherwise} \end{cases}$$

$$X_i^{]} = \begin{cases} X_{1,i}^{[} \text{ if } X_{1,i}^{]} << X_{2,i}^{]} \text{ in } [t_{j-1},\ t_j] \\ \\ X_{2,i}^{[} \text{ otherwise} \end{cases}$$

Then the intersection of $r1$ and $r2$ over $(t_{j-1}, t_j]$ is

$$r'_j = \begin{cases} \{(X_1, \ldots, X_d,\ (t_{j-1}, t_j])\} \text{ if } \forall i,\ X_i^{[} << X_i^{]} \text{ in } (t_{j-1},\ t_j] \\ \\ \emptyset \qquad\qquad\qquad\qquad \text{otherwise} \end{cases}$$

The intersection of $r1$ and $r2$ is :

$$r1 \cap r2 = \bigcup_{j=1}^{N} r'_j$$

Since S can be computed and sorted in $O(d + d \log d) = O(d \log d)$ time and computing each r'_j takes $O(d)$ time, thus $r1 \cap r2$ can be computed in $O(d \log d) + O(N \cdot d) = O(d^2)$ time.

Let $R1$ and $R2$ be two parametric rectangle relations with at most n parametric rectangles in each relation, then

$$R1 \hat{\cap} R2 = \bigcup_{r1 \in R1,\ r2 \in R2} (r1 \cap r2)$$

which can be computed in $O(d^2 n^2)$. ■

Lemma 14. The complement of a relation R of d-dimensional m-degree ($m < 5$) non-periodic parametric rectangles can be evaluated in PTIME in the size of the relation.

Proof: Let n be the number of tuples in R. In each spatial dimension, the tuples contain $2n$ bound functions of time. By Lemma 12, we can partition the t-axis into $O(n^2)$ intervals such that within each interval there is a "$<<$" order of the functions for the lower and the upper bounds. Therefore, for d dimensions, we have a partition of $O(d \cdot n^2)$ intervals, which can be computed in $O(d \cdot n^2 \log(d \cdot n))$ time.

For each interval $[t_{i-1},\ t_i]$, let $f_{1,i},\ \ldots,\ f_{2n,i}$ be the set of bound functions in x_i dimension sorted in an order that $f_{j,i} << f_{k,i}$ if $1 \leq j < k \leq 2n$. These functions partition the x_i dimension into $2n + 1$ parametric intervals

$$(-\infty,\ f_{1,i}],\ [f_{1,i},\ f_{2,i}], \ldots, [f_{2n-1,i},\ f_{2n,i}],\ [f_{2n},\ +\infty)$$

For two dimensions, Figure 8 shows that the space is partitioned into $O(n^2)$ number of small 2-dimensional parametric rectangles, and the shaded regions in the figure are tuples in R. For d-dimensions, the space is partitioned into a set S of $O(n^d)$ small d-dimensional parametric rectangles. For each of the small d-dimensional parametric rectangle in S, we can check in $O(d \cdot n)$ time whether it is contained in any of the parametric rectangles in R, and if it is then remove it from S. The remaining set is the complement of R over the temporal interval

$[t_{i-1},\ t_i]$. This step can be done in $O(d \cdot n \cdot n^d) = O(d \cdot n^{d+1})$. For $O(d \cdot n^2)$ intervals, it takes $O(d \cdot n^2) \times O(d \cdot n^{d+1}) = O(d^2 n^{d+3})$ time.

Therefore, the total complexity is $O(d \cdot n^2 \log(d \cdot n)) + O(d^2 n^{d+3}) = O(d^2 n^{d+3})$. ■

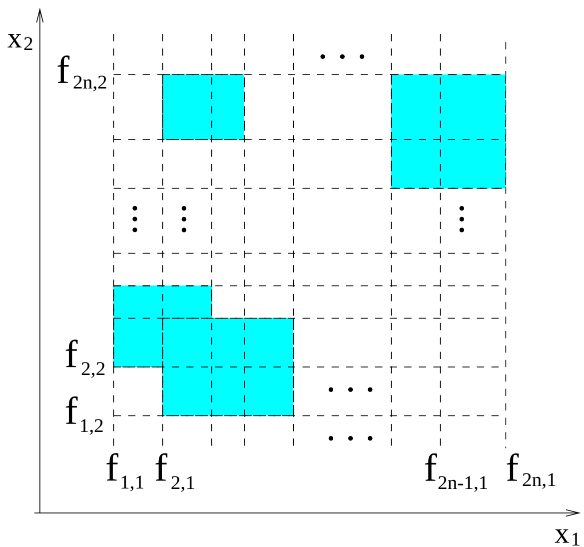


Figure 8. Complement of 2-dimensional parametric rectangle relation

Theorem 15. For any fixed d dimensions, any relational algebra expression can be evaluated in PTIME in the size of the input m-degree ($m < 5$) non-periodic parametric rectangle database.

Proof: By Lemmas 13 and 14, intersection and complement can be evaluated in PTIME. It is easy to see that union can also be evaluated in PTIME. Since

$$R1 \hat{-} R2 \equiv R1 \hat{\cap}\, \hat{\neg} R2$$

the difference can be evaluated in PTIME. By induction on the number of the operators in any relational algebra expression we prove this theorem. ■

Now let us consider the case of periodic parametric rectangles. In this case the main idea is to partition the time dimension into finite intervals and then reduce the periodic case to non-periodic case for each interval.

Let **K** be any finite set of natural numbers such that for any $k1, k2 \in \mathbf{K}$, the least common multiple of $k1$ and $k2$, denoted by $lcm(k1, k2)$, is also in **K**. Similarly to Toman and Chomicki [33] we assume that the periods of all the PPRs in a database are in **K**. We also restrict the slopes of both the bound start lines and the bounds of the intervals in each period as follows: the difference of any two different slopes is at least m and at most M, where m and M are two constants.

Lemma 16. In any dimension, the "width" of any bound belt of a periodic spatio-temporal object r is bounded by a constant.

Proof: Given $r = (X_1, \ldots, X_d,\ T)$, without loss of generality, let us suppose that $T = [from,\ to]_{p,end}$ and $X_i^{[} = a((t - from) mod p) + b + s \cdot p \cdot \lfloor \frac{t}{p} \rfloor$ Then by Definition 4, the functions of the lower bound start and end lines of X_i are:

$$st + b_1 \text{ and } st + b_2$$

where $b_1 = (a - s) \cdot from + b$ and $b_2 = (a - s) \cdot to + b$. Because $(to - from) \leq period \leq \max(K)$ and $|a - s| \leq M$, the absolute value $|b_1 - b_2| = |(a - s)(to - from)|$ is bounded by some constant. Similarly, we can prove that the "width" of the upper bound belt is also at most some constant. ■

Lemma 17. In any dimension, if the bound belts of two PPRs intersect, then the time duration (intersection of the temporal projection) of the crossing part is bounded by some constant.

Proof: Suppose we are given two PPRs $r1$ and $r2$ and the functions of the lower bound start and end lines of $r1$ in the x_i dimension are: (1) $s_1t + b_1$ and (2) $s_1t + b_2$. Similarly, the functions of the lower bound start and end lines of $r2$ in the x_i dimension are: (3) $s_2t + b_3$ and (4) $s_2t + b_4$, where $s_1 \neq s_2$.

Let $t_{i,j}$ be the intersection time of line (i) and line (j) where i and j do not belong to the same PPR. We have:

$$t_{i,j} = \frac{b_i - b_j}{s_2 - s_1}$$

Hence

$$\max(t_{i,j}) - \min(t_{i,j}) = \max\left(\left|\frac{b_4 - b_3}{s_2 - s_1}\right|,\ \left|\frac{b_1 - b_2}{s_2 - s_1}\right|,\ \left|\frac{(b_1 - b_2) + (b_4 - b_3)}{s_2 - s_1}\right|,\ \left|\frac{(b_1 - b_2) + (b_3 - b_4)}{s_2 - s_1}\right|\right)$$

Since $|s_2 - s_1|$ is at least some constant $m > 0$ and by Lemma 16 the numerators are less than some constant, the time duration of the crossing part is bounded by some constant. ■

Lemma 18. The union of two linear PPR relations can be evaluated in PTIME in the size of the relations.

Proof: By Definition 7, the union of two linear PPR relations can be computed as the regular set-union operation, which can be done in PTIME in the size of the relation. ■

Lemma 19. The intersection of two linear PPR relations can be evaluated in PTIME in the size of the relations.

Proof: Suppose $r1$ and $r2$ are two PPRs. For each dimension, there are at most four crossings between the lower and the upper bound belts of $r1$ and those of $r2$, as shown in Figure 9, where solid lines are the bound start and end lines of $r1$ and dashed lines are those of $r2$. There are $O(d)$ bound belt crossings for d dimensions. Projecting the crossings to the t-axis produces $O(d)$ time intervals. By Lemma 17, the size of each interval is bounded by some constant, hence the sum of the sizes of these intervals is $O(d)$. Since the period of $r1$ is at least some constant $min(K)$, where K is the set of all periods in the database, there are $O(d)$ periods of $r1$ falling in these intervals, which can be represented by a set of $O(d)$ non-periodic moving rectangles. Similarly, we represent $r2$ over these intervals by a set of $O(d)$ non-periodic parametric rectangles. Hence the intersection of $r1$ and $r2$ over these intervals can be computed in $O(d^2)$ using the non-periodic parametric rectangle intersection algorithm (Theorem 15).

Now let us consider the $O(d)$ gap intervals between two adjacent intervals discussed above. It is easy to see that within each gap interval $[t1,\ t2]$ and in each dimension x_i the lower and the upper bound start lines of $r1$ and $r2$ do not intersect. Let $f1$ be the "greater" function of the lower bound start lines of $r1$ and $r2$ over $[t1,\ t2]$, and $f2$ the "smaller" function of the upper bound start line of $r1$ and $r2$. If $f1 << f2$, then the intersection of $r1$ and $r2$ can be represented by a set of PPRs whose lower and upper bound start lines are $f1$ and $f2$ respectively, and whose period p is the least common multiple of the periods of $r1$ and $r2$.

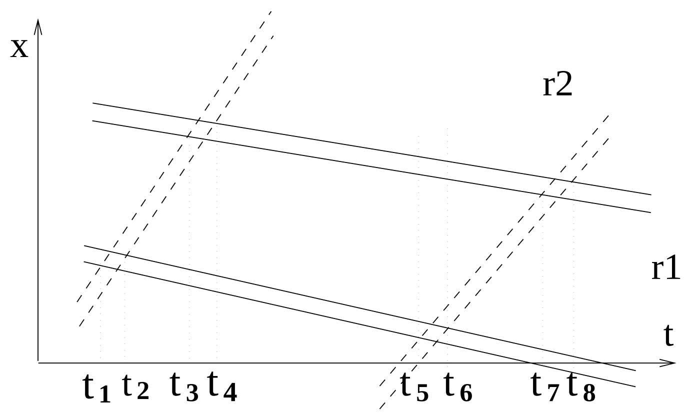


Figure 9. Projection of a periodic parametric rectangle on (x_i, t) space

Consider the first period $[t1,\ t1+p]$. There are at most $\frac{p}{p1}$ periods of $r1$ and $\frac{p}{p2}$ periods of $r2$ falling in $[t1,\ t1+p]$. Since $\frac{p}{p1}$ and $\frac{p}{p2}$ are at most $\frac{\max(K)}{\min(K)}$, we can reduce $R' = r1 \hat{\cap} r2$ over $[t1,\ t1+p]$ to the intersection of two set of $O(1)$ non-periodic parametric rectangles which can be computed in $O(d)$ time (Theorem 15). Each non-periodic PPR r in R' can be translated in $O(d)$ time to a PPR $(X_1, \ldots, X_d, T, t2)$ as follows. For each dimension x_i, let $a_1 t + b_1$ and $a_2 t + b_2$ be the bound functions of r,

$$X_i^{[} = s_1 p \lfloor \frac{t}{p} \rfloor + a_1 (t\ mod\ p) + b_1, \quad X_i^{]} = s_2 p \lfloor \frac{t}{p} \rfloor + a_2 (t\ mod\ p) + b_2$$

where s_1 and s_2 are the slopes of $f1$ and $f2$, respectively, and

$$T = [from + kp,\ to + kp]$$

where $from$ and to are the lower and upper bounds of the time interval of r.

Since there are $O(d)$ gap intervals, $r1 \hat{\cap} r2$ over gap intervals can be computed $O(d^2)$ time. The intersection of $r1$ and $r2$ is the union of their intersections in all the intervals, which can be computed in $O(d)$ time. Therefore the intersection of $r1$ and $r2$ can be computed in $O(d + d^2) = O(d^2)$ time. Let $R1$ and $R2$ be two sets of at most n PPRs. Since

$$R1 \cap R2 = \bigcap_{r1 \in R1,\ r2 \in R2} r1 \cap r2$$

the intersection can be evaluated in $O(d^2 n^2)$ time. ■

Lemma 20. The selection operation can be evaluated in PTIME in the size of the relations.

Proof: Let us assume that the selection has the form $\hat{\sigma}_C R$, where R is a PPR relation with n tuples over $x_1, \ldots, x_d$ dimensions and $A_1, \ldots, A_k$ are attributes, and C is a selection condition. We first write C as $C1$ and $C2$ where $C1$ is the conjunction of all the atomic constraints in C that contain one spatial dimension variable x_i, and $C2$ is the conjunction of those constraints that do not contain any spatial dimension. It is easy to see that $R' = \hat{\sigma}_{C2} R$ can be evaluated in PTIME by checking whether the tuples in R satisfy $C2$.

For each tuple r_j in R', we substitute the attributes in $C1$ with their values in r_j. For each dimension x_i, let $b1$ be the maximum of the set $B1$ of constants in all the constraints of the form $x_i \geq c$ or $x_i = c$ in $C1$ if $B1$ is not empty, otherwise let $b1 = -\infty$. Also let $b2$ be the minimum of the set $B2$ of constants in all the constraints of the form $x_i \leq c$ or $x_i = c$ in $C1$ if $B2$ is not empty, otherwise let $b2 = +\infty$. Let $r'_j = (X_1, \ldots, X_d, T, +\infty, A_1, \ldots, A_k)$ be a PPR with period $+\infty$ such that $X_i^{[} = b1$ and $X_i^{]} = b2$, $T = [-\infty, +\infty]$ and $A_1, \ldots, A_k$ are the same as r_j. It is easy to see that

$$\hat{\sigma}_C R = \bigcup_{r_j \in R'} r_j \mathbin{\hat{\cap}} r'_j,$$

and by Lemma 19, it can be computed in PTIME. ■

Lemma 21. The complement of a relation of PPRs can be evaluated in PTIME in the size of the relation.

Proof: Suppose that R is a relation of n PPRs. Let $R1$ be the set of PPRs in R whose period is $+\infty$, $R2$ the set of PPRs in R whose period is not $+\infty$. Clearly,

$$\hat{\neg} R = (\hat{\neg}\ R1) \mathbin{\hat{\cap}} (\hat{\neg}\ R2).$$

Since by Theorem 15 $\hat{\neg} R1$ can be computed in PTIME, we only have to show that $\hat{\neg} R2$ can be computed in PTIME.

For each dimension x_i, the intersection of the bound belts of the PPRs in $R2$ results in at most $O(n^2)$ crossings. So we have $O(dn^2)$ crossings for all the dimensions, by which we can partition t into $O(dn^2)$ intervals. One type of intervals corresponds to the crossings. The second type of intervals corresponds to the gaps between the first type of intervals.

For the first type of the intervals, by Lemma 17, the time duration of each crossing is bounded by some constant, the total size of the first type intervals is $O(dn^2)$. Hence we can reduce each PPR in R to $O(dn^2)$ non-periodic parametric rectangles similar to the proof of Lemma 19. Also, $\hat{\neg} R2$ over the first type of intervals can be reduced to the complement of a non-periodic moving rectangle relation of size $O(dn^2)$, which can be computed in PTIME (Theorem 15).

Let $[t1,\ t2]$ be an interval of the second type. We first compute the least common multiple p of the periods of the n PPRs in $R2$. The complement of $R2$ can be represented by a set of PPRs whose periods are p. Similar to the proof of Lemma 19, if $t2 - t1 < p$, $R2$ over $[t1,\ t2]$ can be translated to at most some constant number of non-periodic parametric rectangles. Otherwise, we compute the complement of $R2$ over the first period $[t1,\ t1 + p]$ by reducing it to the complement of non-periodic parametric rectangles and then translating the result to PPRs. Since each PPR has at most $p/\min(K)$ periods falling in $[t1,\ t1 + p]$, $R2$ can be translated to a set $R2'$ of $O(n)$ non-periodic moving rectangles. The complement of $R2'$ can be computed in PTIME in the size of $R2'$ (Theorem 15).

Now let us translate each non-periodic parametric rectangle r in $\hat{\neg} R2'$ into a PPR r' as follows: if $r.X_i^[= -\infty$, then $r'.X_i^[= -\infty$; and if $r.X_i^[$ is some function $a \cdot t + b$, then $r.X_i^[$ must be the lower or the upper bound of some o' in R', where o' is translated from some $o \in R2$. Without loss of generality, let us suppose $o'.X_i^[= a \cdot t + b$. Also let s be the slope of the lower bound start line of o in the x_i dimension. Then:

$$X_i^[= sp\lfloor \frac{t}{p} \rfloor + a(t\ mod\ p) + b$$

Similarly we can compute $X_i^]$ from the upper bound of r'. It is easy to see that the translation of $\hat{\neg} R2'$ can be done in PTIME in the size of $\hat{\neg} R2'$. Hence the complement of R over one gap interval $[t1, t2]$ can be computed in PTIME. Since there are $O(n^2)$ gap intervals, the complement of R over the second type of intervals can still be computed in PTIME. Therefore, the complement of R can be computed in PTIME in the size of R. ■

Lemma 22. The difference of two PPR relations can be evaluated in PTIME in the size of the input PPR database.

Proof: The proof follows from Lemmas 19 and 21. ■

Theorem 23. For any fixed d dimensions, any fixed relational algebra expression can be evaluated in PTIME in the size of the input PPR database.

Proof: The proof follows from the lemmas by induction on the number of the operators in the relational algebra expression. ■

4. Implementation

We implemented the query language and animation algorithm in PReSTO (short for *Parametric Rectangle Spatio-Temporal Objects*) using Microsoft Visual C++. The PReSTO system ran in Windows NT, on a 266 MHz Pentium II PC with 64 MB RAM.

Example	Number of Tuples	Running Time (milliseconds)
shuttle bus (Ex. 8)	35	200
solar system (Ex. 9 in 2D)	513	7500
photon (Ex. 11)	4 (photon 1), 50 (photon 2)	741
photon (Ex. 11)	4 (photon 1), 100 (photon 2)	1493
photon (Ex. 11)	50 (photon 1), 50 (photon 2)	4516

Table 4
Query evaluation times

Table 4 shows the execution times for the evaluation of three examples from Section 3. The solar system example was projected into 2D. For the photon example in Section 3, we considered three different input databases.

We implemented in PReSTO a graphical user interface through which the user can specify the following parameters: the name of the parametric rectangle relation, the initial time, the time period, the number of time-steps, and the minimum delay time, which controls the speed of the animation. Each snapshot of the relation is displayed as soon as the animation algorithm returns the corner vertices of the rectangles to be displayed.

In order to display a relation at a time instant t_i, we need to first check whether each parametric rectangle tuple r exists at time t_i, that is , we need to check whether $t_i \in T_r$, where T_r is the time interval of r. If r exists at t_i, we substitute the variable t with t_i in the functions and obtain the snapshot of r at t_i. In this way we obtain a set of rectangles corresponding to the relation at time t_i. These rectangles can be displayed using standard graphics routines. Since the animation is extremely fast, we did not include animation time as a part of the running time.

5. Related work

Constraint databases, introduced by Kanellakis et al. [21,28], may be used to represent spatio-temporal objects. Benedikt et al. [6] study spatio-temporal queries that can be expressed using relational algebra over real polynomial constraint databases.

The parametric 2-spaghetti data model of Chomicki and Revesz [14] generalizes the 2-spaghetti data model [26] by allowing the corner vertices to be represented as linear functions of time. The parametric 2-spaghetti data model cannot represent polynomial and periodic parametric rectangles and cannot be queried by relational algebra, because it is not closed under intersection [14]. However, this model can represent linear constraint databases over two spatial and one temporal dimension [14] and can be used to animate such databases [13].

Chomicki and Revesz [15] propose to represent spatio-temporal objects by a composition of a reference spatial extent at some reference time and various types of transformation functions. In this model, it is easy to obtain any snapshot of a spatio-temporal object, making animation straightforward. However, in some cases this model also may not be closed under intersection [15].

Erwig et al. [16,18] also define spatio-temporal objects that are moving points or regions and an extended SQL query language on such objects. However, in [16,18] they do not consider periodic movement, animation, and the computational complexity of the proposed query language.

Sistla et al. [30] present a model for moving objects along with a query language. This model represents each object's position as a continuous function of time $f(t)$ and updates the database when the parameters of the motion, like speed or direction, change. However, the model captures just the *current* part of the motions that are composed of several parts described by different functions. It does not describe complete trajectories and the spatial extents of moving objects.

Worboys's model [35] can only represent spatio-temporal objects with discrete change. This model can be queried by an extended relational algebra. Grumbach et al. [17] propose another spatio-temporal data model based on constraints in which, like in [35], only discrete change can be modeled. An SQL-based query language is also presented. There are temporal GIS models [2,5,27,25,37] representing spatio-temporal information as a sequence of raster or vector snapshots. None of them can represent continuous change.

Much work in temporal databases aims to represent periodic time points, for example the fact that a class meets every Monday at 10:00 am. Proposals include periodicity constraints over the integers by Bertino et al. [7] and Toman and Chomicki [33], linear repeating points by Baudinet et al. [3] and Kabanza et al. [20], and an extension of Datalog with the successor function by Chomicki and Imielinski [12]. These data models, however, are not able to represent spatial dimensions or continuous movements of objects.

Some other purely temporal or spatial data models are reviewed in [1,26, 31,36]. We also did not deal with issues like visual query languages [8], indefinite information [24], query processing [9], nearest neighbor [23] and approximate queries [34]. We are currently investigating extensions in these directions.

6. Conclusion and Future Work

The periodic spatio-temporal database data model has great application potential in modeling physical processes, which we are currently exploring in more detail. We are also developing additional operators to enhance the expressive power of the query language. Indexing is an important issue when we have a huge number of periodic spatio-temporal objects to be queried. Traditional stationary spatial object index structures like R-trees [19] and R*-trees [4] are inefficient for parametric rectangles. Existing index structures for moving objects by Kollios et al. [22], Saltenis et al. [29], and Tayeb et al. [32] are not efficient either because they are based on the moving points model and do not take periodicity into account. Cai and Revesz [11] investigated the index structure for non-periodic parametric rectangles. We are developing appropriate index structure for periodic spatio-temporal databases.

References

[1] N. R. Adam and A. Gangopadhyay. *Database Issues in Geographic Information Systems.* Kluwer, 1997.

[2] M. P. Armstrong. Temporality in spatial databases. In *Proc. URISA Conference on Geographic and Land Information Systems*, volume 2, pages 880–89, 1988.

[3] M. Baudinet, M. Niézette, and P. Wolper. On the representation of infinite temporal data and queries. In *Proc. ACM Symposium on Principles of Database Systems*, 1991.

[4] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger. The R*-tree: An efficient and robust access method for points and rectangles. In *Proc. ACM SIGMOD International Conference on Management of Data*, 1990.

[5] A. Beller, T. Giblin, K. V. Le, S. Litz, T. Kittel, and D. Schimel. A temporal GIS prototype for global change research. In *Proc. URISA Conference on Geographic and Land Information Systems*, volume 2, pages 752–65, 1991.

[6] M. Benedikt, G. Dong, L. Libkin, and L. Wong. Relational expressive power of constraint query languages. *Journal of the ACM*, 45(1):1–34, 1998.

[7] E. Bertino, C. Bettini, E. Ferrari, and P. Samarati. An access control model supporting periodicity constraints and temporal reasoning. *ACM Transactions on Database Systems*, 23(3):231–85, 1998.

[8] C. Bonhomme, C. Trépied, M-A. Aufaure, and R. Laurini. A visual language for querying spatio-temporal databases. In *Proc. 7th ACM Symposium on Geographic Information Systems*, pages 34–39, 1999.

[9] A. Brodsky, J. Jaffar, and M. Maher. Towards practical query evaluation for constraint databases. *Constraints*, 2(3–4):279–304, 1997.

[10] M. Cai, D. Keshwani, and P. Revesz. Parametric rectangles: A model for querying and animating spatiotemporal databases. In *Proc. 7th International Conference on Extending Database Technology*, volume 1777 of *Lecture Notes in Computer Science*, pages 430–44. Springer-Verlag, 2000.

[11] M. Cai and P. Revesz. Parametric R-tree: An index structure for moving objects. In *Proc. 10th COMAD International Conference on Management of Data*, pages 57–64. Tata McGraw-Hill, 2000.

[12] J. Chomicki and T. Imielinski. Finite representation of infinite query answers. *ACM Transactions on Database Systems*, 18(2):181–223, 1993.

[13] J. Chomicki, Y. Liu, and P. Revesz. Animating spatiotemporal constraint databases. In *Proc. Workshop on Spatio-Temporal Database Management*, volume 1678 of *Lecture Notes in Computer Science*, pages 224–41. Springer-Verlag, 1999.

[14] J. Chomicki and P. Revesz. Constraint-based interoperability of spatiotemporal databases. In *Proc. 5th International Symposium on Spatial Databases*, volume 1262 of *Lecture Notes in Computer Science*, pages 142–61. Springer-Verlag, 1997.

[15] J. Chomicki and P. Revesz. A geometric framework for specifying spatiotemporal objects. In *Proc. International Workshop on Time Representation and Reasoning*, pages 41–6, 1999.

[16] M. Erwig, R. H. Güting, M. M. Schneider, and M. Vazirgiannis. Abstract and discrete modeling of spatio-temporal data types. *Geoinformatica*, 3(3):269–96, 1999.

[17] S. Grumbach, P. Rigaux, and L. Segoufin. Spatio-temporal data handling with constraints. In *ACM Symposium on Geographic Information Systems*, pages 106–11, 1998.

[18] R.H. Güting, M.H. Böhlen, M. Erwig, C.C. Jenssen, N.A. Lorentzos, M. Schneider, and M. Vazirgiannis. A foundation for representing and querying moving objects. *ACM Transactions on Database Systems*, 25, 2000.

[19] A. Guttman. R-trees: A dynamic index structure for spatial searching. In *Proc. ACM-SIGMOD Conference on Management of Data*, pages 47–57, 1984.

[20] F. Kabanza, J.-M. Stevenne, and P. Wolper. Handling infinite temporal data. *Journal of Computer and System Sciences*, 51(1):1–25, 1995.

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

[22] G. Kollios, D. Gunopulos, and V. J. Tsotras. On indexing mobile objects. In *Proc. ACM Symposium on Principles of Database Systems*, pages 261–72, 1999.

[23] G. Kollios, D. Gunopulos, and V.J. Tsotras. Nearest neighbor queries in a mobile environment. In *Proc. Workshop on Spatio-Temporal Database Management*, pages 119–134. Springer-Verlag LNCS 1678, 1999.

[24] M. Koubarakis and S. Skiadopoulos. Querying temporal and spatial constraint networks in PTIME. *Artificial Intelligence*, 123(1–2):223–63, 2000.

[25] G. Langran and R. N Chrisman. A framework for temporal geographic information. *Cartographica*, 25(3):1–14, 1988.

[26] R. Laurini and D. Thompson. *Fundamentals of Spatial Information Systems*. Academic Press, 1992.

[27] D. J. Peuquet and N. Duan. An event-based spatiotemporal data model (ESTDM) for temporal analysis of geographical data. *International Journal of Geographical Information Systems*, 9(1):7–24, 1995.

[28] P. Revesz. *Introduction to Constraint Databases*. Springer-Verlag, 2001.

[29] S. Saltenis, C. S. Jensen, S. T. Leutenegger, and M. A. Lopez. Indexing the positions of continuously moving objects. In *Proc. ACM SIGMOD International Conference on Management of Data*, pages 331–42, 2000.

[30] A. P. Sistla, O. Wolfson, S. Chamberlain, and S. Dao. Modeling and querying moving objects. In *Proc. 13th IEEE International Conference on Data Engineering*, pages 422–32, 1997.

[31] A. Tansel, J. Clifford, S. Gadia, S. Jajodia, A. Segev, and R. T. Snodgrass, editors. *Temporal Databases: Theory, Design, and Implementation*. Benjamin/Cummings, 1993.

[32] J. Tayeb, O. Ulusoy, and O. Wolfson. A quadtree-based dynamic attribute indexing method. *The Computer Journal*, 41(3):185–200, 1998.

[33] D. Toman and J. Chomicki. Datalog with integer periodicity constraints. *Journal of Logic Programming*, 35(3):263–90, 1998.

[34] D. Vasilis, M. Christos, and S. Spiros. A provably efficient computational model for approximate spatiotemporal retrieval. In *Proc. 7th ACM Symposium on Geographic Information*

Systems, pages 140–46, 1999.

[35] M. F. Worboys. A unified model for spatial and temporal information. *Computer Journal*, 37(1):26–34, 1994.

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

[37] M. Yuan. Modeling semantical, temporal and spatial information in geographic information systems. In *Geographic Information Research: Bridging the Atlantic*, pages 334–7. Taylor & Francis, 1996.