SOLVING SPATIAL QUERIES

BY RELATIONAL ALGEBRA

Robert LAURINI, Françoise MILLERET
Laboratoire Informatique Appliquée
Institut National des Sciences Appliquées de LYON
69621 VILLEURBANNE Cedex, FRANCE

Abstract :

In conventional Geographic Information Systems, computational geometry is used to solve spatial queries such as point-in-polygon, region and vacant place queries. In this paper, we propose a formalism (Peano relations) based on linear quadtrees and Peano space-filling curves which allows the solving of the previous queries by a tuple algebra. In essence, it is a relational algebra taking into account the extensional/intensional approach of spatial data. Some examples are taken from urban planning and we conclude this paper by emphasizing several aspects of geomatic reasoning.

Keywords :

GIS, spatial data modeling, quadtree, Peano keys, algebra, spatial query, geomatics, spatial reasoning.

I - INTRODUCTION

Spatial database management systems address to application dealing with geometric and topological data especially in geomatics. One important issue in their design is how to handle queries against geometric information. Some off-the-shelf systems propose two kinds of data:

- attribute data to which relational algebra can be applied,
- and geometric or graphic data for which computational geometry is used to solve spatial queries.

However, when one has to solve a query combining criteria with alpha numeric data and geometric data, he has to make a mixture of relational algebra and computational geometry.

Due to Peano relations, we will show that spatial queries can also be solved by a tuple relational algebra. So, the goal of this paper will be to present a new methodology for answering queries.

Let us examine a small example. Should we ask to retrieve the number of trees in a zone or in a lot of zones, no problem will arise and the answer is obtained by ordinary relational algebra. But if we are interested in the number of trees in a region defined by its boundary, we need to use computational geometry (see Figure 1). In this paper, we will show that Peano relations can allow the solving of this query by relational algebra.

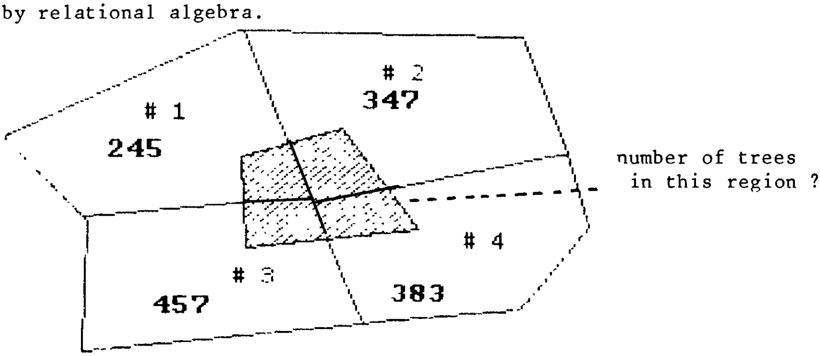

Figure 1 : Example of query not solvable by relational algebra : what is the number of trees in the hatched region

In this paper, we will first present the Peano tuple algebra, then the typology of spatial queries in order to answer a multimedia spatial query example. And we will conclude by some aspects of geomatic reasoning.

II - PEANO TUPLE ALGEBRA

Let us first present the Peano Relations model and second its algebra.

2.1 Peano relations

In several papers (LAURINI, 85, 87 and LAURINI-MILLERET, 87) we have defined a spatial database model whose characteristics are :

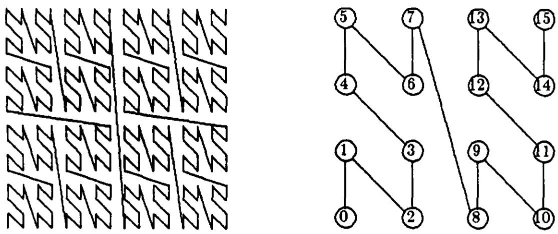

Figure 2 : Excerpts of the Space-filling Peano N-curve

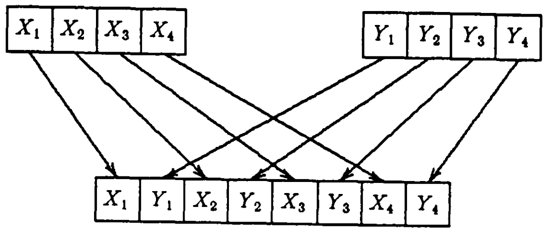

Figure 3 : Obtaining Peano keys by bit interleaving

Peano Key	size	color
0	2	black
4	1	black
5	1	white
6	1	white
7	1	white
8	1	white
9	1	white
10	1	black
11	1	white
12	2	white

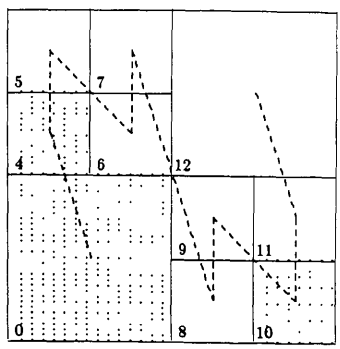

Figure 4 : Examples of a quadtree and its description by Peano relations

- area orientation avoiding the infinite number of tuples
- based on Peano space-filling curves
- based on linear quadtrees and octrees
- use of tuple algebra to solve spatial queries.

In this model, a squared 2D space is described by a recursive splitting into homogenous quadrants (Quadtree), (See SAMET, 1986) and these quadrants are sorted by their Peano Key. In the 3D space we deal with octants. Peano keys p derives from fractal space-filling curves (Figure 2) and the more practical way to obtain them is by the bit interleaving of the x and y coordinates (Figure 3). Peano key based quadtrees are also named linear quadtrees (GARGANTINI, 1983) and Peano keys are also called Morton sequence (MORTON, 1966), z-value by ORENSTEIN (1986) and tesseral arithmetic by DIAZ-BELL (1986)

Let note a Peano relation : R (p, a, A)

in which

- p stands for a Peano key
- a the size of the square/cube
- A a set of domain attributes.

An example of an object described by a Peano relation is given in Figure 9. Often, to shorten, we can exclude white or void squares giving :

R (# object, p, a)

Bearing in mind that a tuple describes a square/cube, it is easy to see that it can be split into 4 (respectively 8) other tuples. So it is an intensional/extensional way of describing space and the rule is

"One tuple can always be split into 4 (8) tuples"

In order to deal with consistent and compact objects, 3 conformance levels are necessary:

- well positionned squares/cubes
- overlaping elimination
- maximum compaction.

2.2 Manipulation

In (LAURINI, 1987) the Peano tuple algebra for manipulating object is given. Beside geometric and boolean operations, relational operations are very useful for manipulations. For instance the Peano join can be used to solve point-in-polygon and regions queries.

Let us consider an example. A scene consisting of three objects, A, B and C is described by means of a Peano relation SCENE (p, # object, a) and we want to test a region REGION (p, a) in order to know what are the objects within it. See Figure 5. The result is given, first by a Peano join between SCENE and REGION and second by a projection of this result. In the example, the tuple SCENE (B, 52, 2) can be disaggregated into four tuples SCENE (B, 52, 1), SCENE (B, 53, 1), SCENE (B, 54, 1) and SCENE (B, 55, 1)).

Scene		
object	Peanokey	size
A	16	2
A	24	2
A	28	1
A	30	1
B	29	1
B	31	1
B	48	2
B	52	2
C	0	4

Region	
Peanokey	size
30	1
31	1
52	1
53	1
55	1
62	1

Result of the join		
object	Peanokey	size
A	30	1
B	31	1
B	52	1
B	53	1
B	55	1

Project result
object
A
B

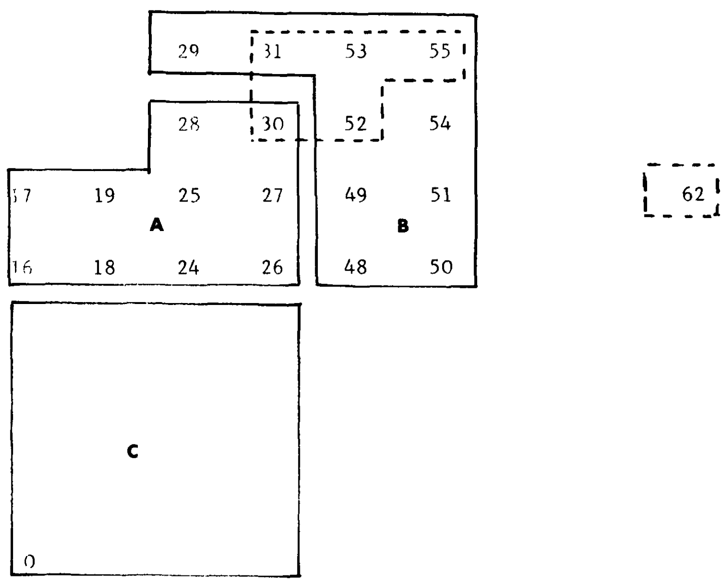

Figure 5 : Examples of spatial object (A,B) and a region query

III - SOLVING SPATIAL QUERIES

The role of spatial queries is to provide algorithms to solve questions. See LAURINI-MILLERET (1988). Among them, the more important seem to be :

- point-in-polygon query,
- region query,
- vacant places.

We have recently established that Peano relations give faster algorithms to solve spatial queries than wireframe representation.

3.1 Point-in-polygon (Fig. 6a)

Starting from a x_0, y_0 point, the problem is to find what objects it belongs to. An example is to determine who is the landowner of a point in a cadaster. Let us suppose we have n plots of land.

With a wireframe representation, the solution is given by the half-line algorithm whose complexity is O (n) (PREPARATA-SHAMOS, 1986). With the cell-oriented representation based on Peano relations, the complexity becomes O (log n) (LAURINI, 1987).

3.2 Region query (Fig. 6b)

Here, starting from a zone called region, one has to determine what are the objects belonging to it. It is the same problem as the point-in-polygon query except that the point is replaced by a zone. As an example in town planning, we can try to retrieve the landowners affected by the creation of a new freeway, or the list of urban objects in a zone defined by its boundary.

With the wireframe representation, one has to perform an algorithm based on an intersection which is very complex to design. With Peano relations, this query is solved by a Peano join.

3.3 Vacant places (Fig. 6c)

Here, the problem is to retrieve vacant places within a predefined zone. Wireframe representation leads to a geometric difference algorithm more difficult to be written than a relational difference algorithm with Peano relations.

In a same way, we have shown that Peano relations allow the easy solving of spatial query by means of a join operator taking into account the extensional/intensional aspects (Peano join). See LAURINI (1987).

3.4 Other spatial queries

Among other spatial queries, let us mention distance query. For instance, we want to retrieve all parcels within a distance of 3 km from a precise one.

With the wireframe representation, in the first step, one has to determine a region built from the given parcel and the distance, and to apply the region query. The computation of the shape of this region is not very simple to perform, especially in the case of holes and concavities.

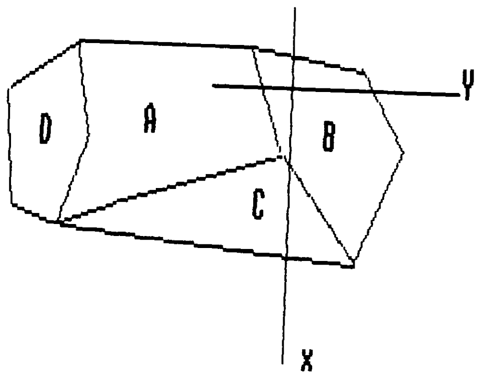

Fig. 6a: Point-in-polygon query

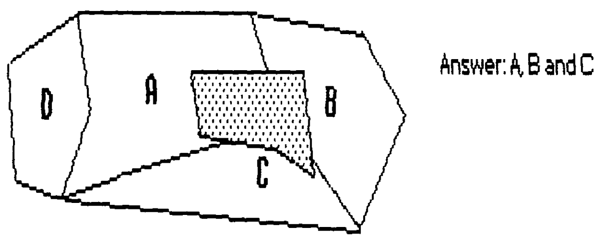

Fig. 6b: Region query

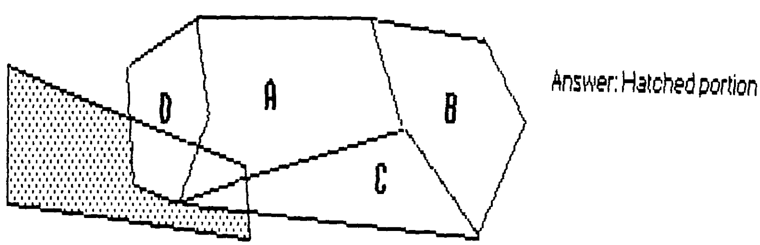

Fig. 6c: Vacant place query

Figure 6: Examples of spatial queries

However, with quadtrees, we can determine the region by constructing the union of all quadrants and their neighbors within the distance. After this step, the result is given by a Peano join.

3.5 Example of a multimedia spatial query

Suppose, after a flooding, we want to retrieve all farmers affected by this flooding in order to indemnify them. For that, let us start from the cadaster and aerial photographs of flooded fields. After having given the structure of the land data, and pixel-based photographs, we present the solving process.

a) Land data structure

Let us have a relation giving a parcel and its landowner and three other relations for parcel boundary description :

R_1 (# Parcel, # Farmer)

R_2 (# Parcel, # Segment)

R_3 (# Segment, # Point 1, # Point 2)

R_4 (# Point, x, y)

b) Aerial photographs

Suppose an aircraft has taken digital photographs of flooded area with several gray levels. Moreover, suppose the exact position of each photo in term of coordinates and orientation is known.

P_1 (# Image, x pixel, y pixel, gray level)

P_2 (# Image, x pos, y pos, length, width, orientation)

c) Query solving (Figure 13)

To solve this spatial query implying geometric objects described with various geometric representation, its seems interesting to map into the linear quadtree representation (Peano relations) whose main advantage is to use relational algebra to solve spatial query. See LAURINI 1987, or LAURINI-MILLERET 1987 and 1988 for more details. First, let us deal with aerial photos for which a relational restriction must be applied to cancel all pixels not corresponding to the water. By examining gray levels of pixels, this operation will be performed so giving A_1 corresponding to P_1 reduced to water.

A_1 (# Image, x pixel, y pixel)

Second, all these pixels have to be positioned in the coordinate system and geometrically corrected due to photographic distorsions transforming some pixels into rectangles :

A_2 (# Image, x corrected, y corrected, length, width)

The next step will be to regroup all these relations in order to cover the whole territory by quadtrees governed by Peano keys :

FLOODING (Peano Key, size).

In the same manner, we have to transform land information into Peano relations : so, starting from R_2, R_3 and R_4, we can construct the following relation :

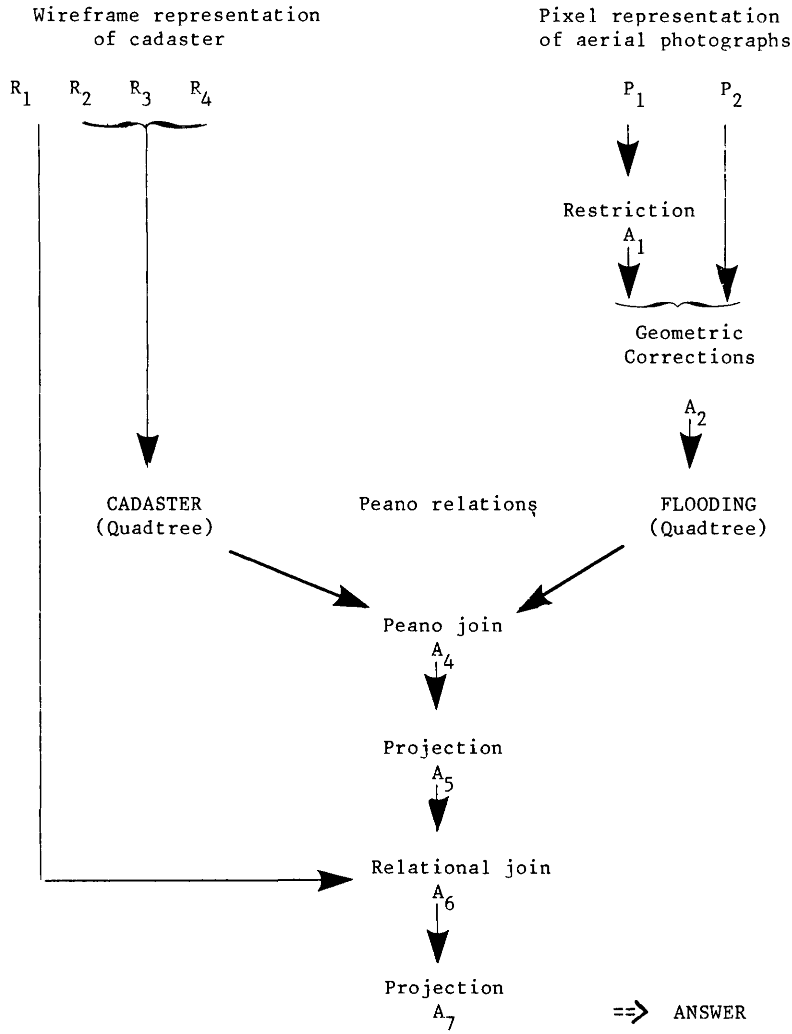

Figure 13 : Solving algorithm for the multimedia query

CADASTER (# Parcel, Peano Key, Size)

To obtain the result, a Peano join has to be performed between FLOODING and CADASTER to give A_4 corresponding to only flooded plots :

A_4 (# Parcel, Peano Key, Size)

Now, to know the list of flooded farmers, in a first step, we will perform a projection on A_4 to give only the parcels (A_5) ; then a relational join of A_5 with R_1 to give A_6 which will be followed by a projection to get A_7 which is the answer.

A_5 (# Parcel)

A_6 (# Parcel, # Farmer)

A_7 (# Farmer)

So, the result is obtained through an amalgamation of computational geometry and relational and Peano algebras.

IV - CONCLUSIONS

The scope of this paper was to show that some spatial queries can be easily answered by tuple algebra when solid objects are described by Peano relation. A multimedia example, taken from urban planning has illustrated this fact.

For the design of a spatial DBMS, we do think that the representation must be chosen not only from storage criteria but also from the facility to solve spatial queries. Peano relation and cell enumeration methods are very good candidates for the foreground of this design.

BIBLIOGRAPHY

DIAZ BM, BELL SBM (1986) Spatial data processing using Tesseral Methods. Publ. by Natural Envir. Research Council (UK), Sept 1986.

GARGANTINI I (1983) Translation, Rotation and Superposition of Linear Quadtrees. International Journal of Man-Machine Studies. Vol 18, 3, March 1983, pp 253-263.

LAURINI R (1985) Graphics Data Bases Built on Peano Space-Filling Curves. EUROGRAPHICS'85, Nice, Sept.8-13/1985. pp 327-338. Ed. by C.E. VANDONI, NHPC.

LAURINI R (1987) Manipulation of Spatial Objects with a Peano Tuple Algebra. University of Maryland CfAR, Technical Report. CAR TR 311.

LAURINI R, MILLERET F (1987) Peano Relations in CADCAM Databases. Int. Conf. IEEE "Data and Knowledge Systems for Manufacturing and Engineering", Hartford, Connecticut October 19-20/1987.

LAURINI R, MILLERET F (1988) Spatial data base queries: relational algebra versus computational geometry. "IVth International Conference on Statistical Scientific Data Base Management", Rome, Italy, 21-23 June 1988, Vol 2, pp 23-44. Edited By M. RAFANELLI.

MORTON GM (1966) A Computer Oriented Geodetic Database and a New Technique in File Sequencing. IBM Canada-Ontario report, March 1966.

ORENSTEIN J (1986) Spatial Query Processing in an Object-Oriented Database System. Proceedings of ACM/SIGMOD'86, Washington DC. pp 326-336. Edited by C. ZANIOLO.

PREPARATA F, SHAMOS M (1986) Computational Geometry: an Introduction. Springer-Verlag.

SAMET H (1984) The Quadtree and Related Hierarchical Data Structures. Computing Surveys. Vol 16, June 1984, pp 187-260.