RESEARCH INTO AN INTERACTIVE SPATIAL INFORMATION SYSTEM

Greg Blair

Collins & Moon Ltd.
435 Stone Rd. W., Suite 215
Guelph, Ontario N1G 2X6 CANADA

ABSTRACT

Data base research is being conducted on the design and implementation of a spatial information system. The data base research consists of extending existing data base technology to handle spatial and semantic information simultaneously. The data base design supports interactive insertion and extraction of spatial and semantic information.

This paper describes progress to date in this research. Implementation details on data structures that support generalized resolution storage of spatial data, resolution dependent retrievals, topological relation retrievals and the analysis of topological properties for spatial data are given.

KEYWORDS: relational data models, spatial data, topological data, variable length attributes, generalized spatial data structures.

1. INTRODUCTION

Spatial information is topological information which has an embedded coordinate schema. This coordinate schema may be quantitative or qualitative. If quantitative, then numeric coordinates are embedded in the information. Metrics can be defined that measure distance between coordinates. If qualitative, either relationships, such as direction or proximity, are embedded in the information or, a reference, such as a name, to a spatial feature is embedded in the information. From quantitative coordinates, qualitative relations can be deduced. The semantics define the topology and the topology carries the coordiantes.

Semantic models can be stored in relational data model either as semantic graphs or semantic relations (Borkin, 80). The mapping from spatial semantic graphs to a relational data model has been done before (Chang, 81).

This research was supported by the Defense and Civil Institute for Environmental Medicine of the Dept. of National Defense of the Government of Canada under Department of Supplies and Services contract number 2379990/8SE80-99952/1450AA-3270009-02-D679.

The current state of spatial information systems is that of a one-off design per application, usually incorporating a file structure to hold the data. This has been done in computer aided cartographic, drafting, imaging, animation, simulation, design, and manufacturing systems. The common requirement between these systems is a database mechanism that can handle vector, matrix, and semantic data, usually variable length data.

2. RSDB

A relational model can handle semantic relations with ease. If the set of data types is rich enough, spatial data can be handled with equal ease. The inclusion of spatial data requires spatial operators for manipulation. A new database management system was built because none of the commercially available relational database management systems supported variable length attributes, or matrix attributes. This is not surprising, since commercial packages evolved to support business applications. A relational spatial data base (RSDB) management system has been developed.

RSDB supports variable length attributes in a relation. Integer, floating point and byte atomic data types are supported. The length of the atomic data type for an attribute is user defined. An attribute consists of a matrix of an atomic data type. The attribute matrix may be static or dynamic. A static matrix is a fixed dimensioned matrix. A dynamic matrix is a variable dimensioned matrix. An attribute matrix may have an arbitrary number of dimensions.

Coordinate data is usually two or more numeric quantities. These values - ie. (x,y,z) or (x,y) - can be stored as a vector. A vector is a matrix with either only one row or column, depending on whether it is a row vector or volumn vector. Most automatically scanned data is either matrix data or a time series of matrix data. Some examples are radar, multi-spectral scanner, sonar, and interferometer data.

3. RSDB COMPONENTS

There are several componenets of RSDB.

3.1. FILES

There are three classes of files to be managed.

(1) System files which describe to the RSDB software the layout of relations and attributes.

(2) Data files containing relation tuples.

(3) Index files which are used to optimize retrievals, updates, and deletions. There are two types of index files. The first type is the usual balanced tree index found in relational databases. This type allows smart joins between relations and range searches on a domain used as a key to be done quickly. The second type is a spatial index. The spatial index is a descriptor that permits efficient operations to be done on spatial domains. These descriptors can be a minimal area enclosing box for an object or a can be supplied by a custom application supplied subroutine.

3.2. RSDB CODE

Code exists that:

(1) creates/destroys relations
(2) inserts/deletes tuples into/from a relation
(3) joins two relations
(4) handles nested joins
(5) projects relations or projects the result of a join/nested join

To delete or selectively retrieve a tuple, that tuple must be specified. A where clause specifies a condition for a set of tuples in a relation. A where clause is a tree of boolean nodes. Each node requires an action to be taken to see if a tuple satisfies a condition. Each action is either a straight forward comparison or a spatial comparison. A spatial comparison must be done between a numeric attribute and either another numeric attribute or a numeric constant. There aresix types of nodes

(1) CONSTANT PREDICATE NODE This node consists of a relation + attribute name pair, a compare operator, and a constant. The attribute of the named relation is compared to the constant by the compare operator. The compare operators supplied are: equality; less than; greater than; less than or equal; greater than or equal; and a regular expression string match. The spatial compare operators supplied are proximity, adjacency, intersection, and containment.

(2) VARIABLE PREDICATE NODE This node consists of a relation + attribute name pair, a compare operator, and a second relation + attribute name pair. The first relation + name pair is compared to the second relation + attribute name pair by the compare operator. The compare operators supplied are: equality; less than; greater than; less than or equal; greater than or equal; and a regular expression string match. The spatial compare operators supplied are proximity, adjacency, intersection, and containment.

(3) COMPOUND PREDICATE NODE This node consists of a relation + attribute name pair, a compare operator, and a where clause. This node controls nested joins. The compare operators suppled are: equality; less than; greater than; less than or equal; greater than or equal; and a regualr expression string match. The spatial compare operators supplied are proximity, adjacency, intersection, and containment.

(4) BOOLEAN NOT NODE This node consists of a boolean not operator and a where clause. The where clause is evaluated against a tuple and the result inverted.

(5) BOOLEAN OR NODE This node consists of the boolean or operator and two where clauses. The first where clause is evaluated. If the result was true, this node has a value of true. If not, the second where clause is evaluated, and this node has the value of the second where clause.

(6) BOOLEAN AND NODE This node consists of the boolean and operator and two where clauses. Both where clauses are evaluated, and this node is true if both where clauses were true.

4. APPLICATIONS

4.1. LINE DRAWING

A graphics editor that allows line drawings to be digitized, stored, retrieved, edited, and plotted has been completed. Four relations were created:

(1) - an area relation - each tuple contained several attributes: a variable length name; fill codes; colour; and a varying number of (x,y) coordinates that define the outline of the area

(2) - a line relation - each tuple contained several attributes: a variable length name; line type codes; colour; and a varying number of (x,y) coordinates that define the line

(3) - a symbol relation - each tuple contained several attributes: a variable length name; symbol name; scale; size; colour; and a (x,y) coordinate

(4) - a text relation - each tuple contained several attributes: variable length text; font; size; colour; and a (x,y) coordinate

This work was commissioned by a psychologist studying the human computer map interface for the armed forces.

4.2. GENERALIZED DATA STRUCTURE LINE DRAWING

One of the problems in manipulating spatial data is that data is required at different resolutions at different times. An example is zooming in on an object by windowing to change the scale for display. A small scale image requires relatively few points to be retrieved compared to a large scale image of the same object.

Strip trees are a mechanism for storing lines that enable the coordinates to be retrieved at coarser resolution than that at which the coordinates are stored at. (Ballard, 78) The idea is straight forward: The data points are stored in a binary tree. Consider a monotone increasing function $y = f(x)$. The root node for the tree contains the end points (x1,y1), (xn,yn) and the maximum deviation from the straight line connecting the two end points. The end points and this maximum deviation define a strip that contain the line segment (x1,y1), ...,(xn,yn). The root node contains two son nodes. The first son is a strip tree node for (x1,y1),...,(xi,yi) and the second son is a strip tree node for (xi+1,yi+1),...,(xn,yn). This subdivision process is continued until the thickness of the strip is below some threshold. This results in a variable number, depending on the shape of the data, of tree nodes. The monotonic restriction can be dispensed with by considering the bisecting point for a strip subdivision, to be the upper right coordinate for the enclosing rectangle. Strip tree nodes can be stored in a variable length attribute.

Similarly, a quad tree or an oct tree generalized data structure (Doctor, 81) can be stored in a variable length attribute. A quad tree is an area generalized storage structure. An oct tree is an volume generalized storage structure. The RSDB storage for these type of tree nodes is the variable length matrix attribute.

An interactive strip tree/quad tree line drawing system, similar to the line drawing system mentioned above, has been completed. All line coordinate strings are stored in quad trees. This work was an experiment in rapid display techniques for maps.

5. CONCLUSION

RSDB has proven itself a powerful toll in supporting interactive computer graphic and cartographic applications. RSDB is applicable to any relational data base application requiring variable length or fixed length attributes in a relation.

Business applications exist for a relational data base supporting variable length domains. RSDB would have to be enhanced to support:

(1) - a "natural language" data base query interface. Several designs are under consideration.

(2) - a report generator

(3) - a forms editor

(4) - transaction journaling. Journaling can be added to the file access sub-system.

(5) - concurrency control. A record locking concurrency control approach is more palatible than a simplistic file lock out mechanism.

(6) - data relation views. The projection list mechanism provides a rudimentary view.

(7) - security controls. These have to be added to police who is permitted to create, destroy, and update relations, and their attributes.

(8) - attribute triggers - updating an attribute, triggers something to happen somewhere else, such as incrementing an attribute, or running a program.

The restriction of spatial operators to explicitly typed spatial attributes is under consideration. The listing of spatial domain types such as: points; lines; polygons; surfaces; solid objects; polyhydron; etc. has been influenced by the line drawing (mapping) and CAD/CAM fields. How many representations for surfaces should be supported? What additional spatial operators should be included in RSDB? What algorithms should be used for calculating these additional spatial operators?

The package is written in 'C'. A FORTRAN programming interface language exists. The file sub-system uses open, close, create, read, write, and lseek system calls. It should be simple to port RSDB to any machine with a 'C' compiler. Currently, this list includes IBM 370's/Amdahl's, pdp11's, VAX11's, Perkin Elmer 3200's, 68000's, Z80's, 8080's, Honeywell and others.

6. REFERENCES

Ballard, Dana H., Hierarchical Representation for Map Features, Report TR32, Defense Advanced Research Projects Agency, Arlington, Virginia, 1978.

Borkin, Sheldon A., Data Models: A Semantic Approach For Database Systems, The MIT Press, Cambridge, Massachusetts, 1980.

Chang, Nig-San, Image Analysis and Image Database Management, UMI, Research Press, Ann Arbor, Michigan, 1981.

Doctor, Luis and Torborg, John G., Display Techniques for Octree-Encoded Objects, IEEE Computer Graphics and Applications, V1, #3, 1981, pp 29-38.