MANAGEMENT OF REMOTELY SENSED IMAGERY BASED ON GEOGRAPHICAL INFORMATION SYSTEM

Li Yang & Jian-kang Wu
Image Processing Lab.
Univ. of Scie. & Tech. of China
HEFEI, CHINA
Commission: IV

ABSTRACT

With the rapid growth of geographic information system (GIS), and its wide application in remote sensing, how to organize the large amount of data efficiently has become a serious problem. The idea of integratly remotely sensed data in geographical information system for mutual enhancement appear quite obvious. In this paper, we present some utilizations that incorporate a combination of remotely sensed data and geographic information for environmental assessment. We propose an object-oriented run-length coding scheme to encode the entities in GIS. Image algebraic algorithms based on run-length coding and convertions with other data structures are fully discussed. Run-length coding has advantages such as quick display on raster formatted screen, easy output by plotter, suitable for image algebraic operations. It has shown to be a promising alternative to vector format and quadtree data structure.

Next, we discuss the automated interpretation of the remotely sensed data. The interpreter use the location network as the inference mechanism and the frame system as the knowledge representation method. The aim of research is to develop an spatial information inference and interprete system based on the GIS, which incorporates some of the geophysical and photogrammetric knowledge and which can perform a knowledge-based automated geophysical interpretation. The object-oriented run-length coding scheme and location network inference now are used in a Knowledge-based Spatial Database System (KSDS).

INTRODUCTION

In the last twenty years, pictorial database systems, especially geographic database systems, has been developped rapidly[1]. There is a large varity of systems based on different data structures, *e.g.* Freeman [2] described various coding methods of the line features including the famous Freeman chain code. Sammet *et al* [3] described a quadtree based graphical information system which seems very flexible. Peuquet [4] introduced a hybrid structure which integrate the chain-coded vectors into raster-formatted scan lines *e.g.* swaths. Although each of above data structures has its own advantages, we think it is neccesary to seek for another data structure, which can provide combination of compactness, computational efficiency, flexibility to satisfy the processing requirements of geographic information systems. Vector structure does have advantages for coding line object. It is convincient for geometric transformation, but inefficient for image algebraic operations on regions. Operations on quadtree coded objects are based on tree travesal. This leads to consideration of both the data handling and storage problems, and its operations can not be executed parallelly. Geometric transformations are especially difficult for quadtree coded objects.

The present paper will examine this problem and offer a third type of data structure, object-oriented runlength coding, which is designed to combine the advantage of raster and vector data formats within a unified structure. Object-oriented runlength coding is first proposed by Jungurt [5] as a spatial data structure and the run-length coding based map managemant system MASK (MAp Supervising and Knowledge base system) [6] was implemented. Its main aim is to meet the requirements of cartography. From the view point of knowledge-based spatial database design, we propose here a object-oriented run-length coding scheme with a set of associated algorithms, which together form a spatial data storage and manipulation sub-system. In what follows, we first give the definition and coding scheme for runlength codes, and then discuss image algebraic algorithms and conversions to other data structures.

The concept of location network was formally proposed by RUSSELL [7], It is a form of semantic network in computer vision, and is used mainly as a represention method of scene knowledges, it supports the geometrical inference mechanism that computes two-dimensional search areas in an image. As defined in [8], a location network is a network representation of geometric point sets related by set-theoretic and geometric operations such as set intersection and union, distance calculation, and so forth. The operations corresponed to restrictions, or rules, are dictated by cultural or physical facts. Every internal node contains a geometric operation, inference is performed by evaluating the net from the leaf nodes to the root. These geometric operations are fully supported by the image algebra or mathematic mophorlogy methods.

Location network has poor ability of knowledge representation, but it is very flexible when used as the evaluation framework of a specified problem. Thus, it can be served as the control strategy in the graphical information interpretation environment, it can supports the sophisticated spatial information processing, and provides the flexible user interface. In the system, location network is used as the evaluation mechanism to execute the geometrical operation.

OBJECTED-ORIENTED RUN-LENGTH CODING

We can give the informal description of run -length coding as follows:

Definition:

Consider a collection of boundary points of an object region, to display this object region on the raster display device, the boundary points should be first sorted according to the *y*- axis coordinate and then the points with the same *y*- axis are sorted according to *x*- axis. By parity checking or other methods, the region then can be filled line by line in a raster data structure manner.

$$Y_1 , X_{11} , X_{12} , \ldots$$

$$Y_2 , X_{21} , X_{22} , \ldots$$

$$\vdots$$

From the definition we can see that this coding method is actually the combination of the raster data structure and the vector data structure. Similar to vector structure, it only contains boundary points of the object. On the other hand, it has property of raster data structure because this method is based on image pixels and has the raster scanning manner.

Futhermore, the run-length code of a object can be represented as a list of run-length terms, each term in the list has the form of tri-turple, such as:

$$(Y_1 , X_{11} , X_{12})$$

where: Y_1 denotes the Y-coordinate of the term,
X_{11} denotes the start X-coordinate of the term,
X_{12} denotes the end X-coordinate of the term,
as illustrated in Fig.1 .

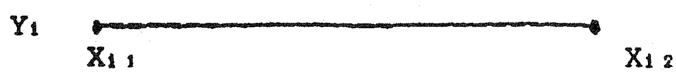

Fig. 1

So, the run-length codes can be viewed as a list of integer numbers, operations on coded image objects can be all implemented by algebraic operations. This reduces the computation complexity greatly.

If the codes were sorted on y coordinate with ascending order, or decending order otherwise, and the coded object is simply connected, based on this hypothises the run-length term can be further compassed into binary-tuple, such as follows:

$$(X_{11} , X_{12})$$

where:

1. if $X_{12} = 0$ then X_{11} represents the *y* coordinates of the next tuple. For a sequence of continuous ascending *y*, only the minimum *y* is to be coded.
2. if $X_{12} = 0$ then X_{11} , X_{12} represent the start and end *x* coordinates of the run length term, respectively.

So, the coding of a region can be represented as :

$\{ (X_{i1}, X_{i2}), i=0,\ldots,n \}$

The generalized form of run length coding of a connected region is as follows:

$\{ (Y, 0), (X_{1,1}, X_{1,2}), (X_{2,1}, X_{2,2}), \ldots\ldots, (X_{n,1}, X_{n,2}) \}$

Y denotes *y* coordinate of the first term $(X_{1,1}, X_{1,2})$, the mininum *y* value of this region.

Assume the *y* coordinate of the data is sorted by ascending order, then, the triple form of the coding can be recovered from bi-tuple form by using the following procedure :

```
  Input:  { (Xi,1 , Xi,2), i=0,..., n }
  Output: { (Yi , Xi,1, Xi,2), i=1,...,n }
  Procedure:
      for i = 1 to n do
              if Xi,2 equal 0 then          /* case 1 */
                      Yi = Xi,1
              else
                      if Xi,1 > Xi-1,2 then     /* case 2 */
                              Yi = Yi-1
                      else                      /* case 3 */
                              Yi = Yi-1 + 1
                      end if
              end if
      end do
```

Image object may consists of several individual parts, which appear as several connected regions. The object code will then be composed of codes of individual region, seperated each other using a particular symbol such as (0, 0).

To form a complete relational description of an object, an object name and several attributes can be inserted in front of this data record. Probably, the maximum and minimum of *x,y* coordinates can give quick estimation of object position. Other geometric properties, such as area, short axis, elongation, etc. may be also useful in some applications. One may then have

$\{$objname, centriod, $X_{max}, X_{min}, Y_{max}, Y_{min}, L_{axis}, S_{axis}$, run-length codes $\}$

The same coding scheme is also usefulm for point objects and line objects. In the case of point object, the data record is simplified as

$\{ (X_i, Y_i) \}$

IMAGE ALGEBRA BASED ON RUN-LENGTH STRUCTURE

As discussed above, the run-length coding of a object is based on the pixels in raster format. So, we can easily implement the image algebra on run-length coded objects either parallelly or sequentially.

In this section, algorithms for performing image algebra operations on objected-oriented run-length coded object regions are presented. Attentions are paid on some basic algorithms such as union, intersection, difference. Other operations, such as dilation, erosion, opening, etc. are based on these basic operations and coordinates shifting. These basic operations can be described as: Given the run-length codes of two region A and B, compute A U B, A ˆ B, A - B, and the results are also represented as the run-length codes. We can demonstrate here that computation of the algorithm is linear with respect to number of run-length terms in two input codes.

Each run-length code of the object is essentially an ordered sequence of integers. Hence, the basic set opeation on two objects resembles a merge operation on two sorted lists of numbers. The set operations based on the run-length coding can be reduced to the computing of the spatial relation of two appropriate run-length terms. We can easily design these algorithms and put it into practical use. These operations include intersection, union, difference, shifting, scaling, overlay, etc.

Computation of geometric properties of the objects encoded by run- length coding is a trivial process because it involves only point operations which are fully supported by the run-length coding. For example, to computer the area of a region, one need only sum up the length of all the run-length term :

$$\text{area} = \text{sum}_{i} \; (X_{i2} - X_{i1} + 1)$$

In KSDS, a general query is decomposed into relational query and graphical query through the query command parser. As illustrated is Fig. 2, relational query is only related to the relation in relational database and be implemented by relational algebra or relational calculus. The graphic part of the query must be implemented by the image algebra and the access to graphic database. Complicated image algebraic operations on the run-length coding are further decomposed into basic ones. By performing these basic operations, the query answer is achieved.

$$\text{General query} \xrightarrow{\text{parser}} \text{query atom} \begin{cases} \nearrow \text{ralational query (relational calculus)} \\ \searrow \text{graphic query (image algebra)} \end{cases}$$

Fig. 2 Query decomposition.

Image algebra is based on mathematic morphology. It has many advantages over traditional image operations. The mechanism of the algorithm based on morphology theory is parallel and based on raster data structure[9]. Each image object in image algebra is represented as point set. This coincides very well with run-length coding. Therefore run-length coding is a good candidate for image algebra data structures. Another facility of this run-length coding method is the possibility of browsing and the zooming operation [10]. This will make it easier to the user to find a desired region.

DATA CONVERSION WITH OTHER STRUCTURES

Data conversion between various data structures is often required in practical systems. Data conversion problems occur most frequently in the input/output module of the spatial database system. For instance, large amount of terrain data are in vector format due to the characteristics of terrain map digitizer. To facilitate query process and raster display, terrain data must be converted to and stored in runlength codes. Output of geographic information system is often produced using the vector formatted device such as plotter in cartographic applications. Runlength to vector conversion is then required. Other data format conversions may be neccesary for data sharing between various types of geographic information systems. With the rapid development of remote sensing imaging and processing technology, remote sensing images and their interpretation results become very important data resource of the geographic information system. This enables the dynamic monitoring of the earth resources and enviroment. Remote sensing data are inherently raster formatted, data conversion to vector format must be used for map productions. In what follows we briefly discuss data conversions between runlength code and the other data formats.

1) Generation of run-length code from raster data

Suppose, we are given a raster image which contains several objects in it, the runlength code generation procedure then takes the following steps:

a) label the connected regions, a label is assigned to each individual object;

b) code generation for each labelled object region;

c) save the results to data management system.

2) Generation of run-length code from vector data

By converting the vector format data to the raster image, the vector to run-length coding conversion can be performed as in Fig. 3.

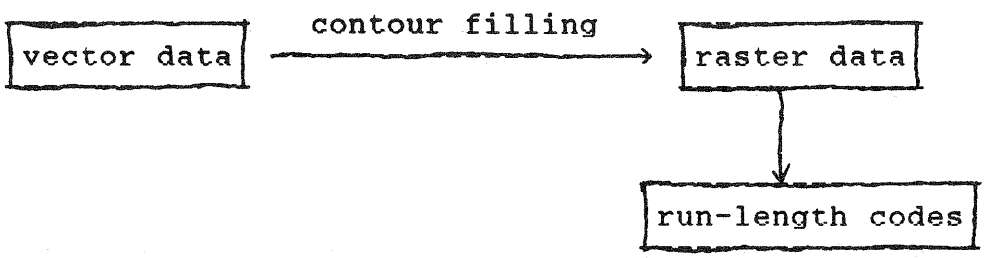

Fig.3 Flowchart of vector to run-length

3) Run-length code to vector conversion

Convert run-length codes to vector form means to generate the boundary chain from the run-length codes. Boundary tracing algorithm on run-length structure is firstly invoked with prediction features according to local imformation. As mentioned above, run-length structure is based on the image pixel, therefore algorithms on raster structure can work on run-length data structure without much modification.

A problem arises , however, for adjacent regions sharing pieces of boundaries and any objects having common points or line segments. Run-length code is a object-oriented coding, this results in coding redundancy for these common parts. For example, part of country boundaries may also be part of other geographical line entities such as roads, streams, shorelines,etc. In vector coding this common part can be only coded once. Therefore the conversion is designed to consist of following steps:

1. Assign different priorities to various type of image entities;
2. find the highest priority for the common part, eliminate this part from image entities having lower priority; for example, rivers are usually assigned higher priority than district boundary, when they are converted to vector form, their common parts can appear only in river data.
3. Sort the data points in clockwise/counter-clockwise sequence;
4. Perform piecewise linear process to the sorted points.

To implement this convertion, a queue of image objects is required. In the queue, object with high priority is placed in front of the object with lower priority. The plotting algorithm is described as follows:

```
  Input: object boundary sorted in priority descending order:
         { Queu(i), i= 1, ... ,n }
  Output: plotting result.
  Procedure:
      for i = 1 to n do
         for j = 1 to i-1 do
              if queu(i) and queu(j) have overlapped parts then
                put these parts to temp storage.
              end if
         end do
         plotting  queu(i) - temp
      end do.
```

LOCATION NETWORK INFERENCE

Image database system techniques have developed so quickly that the research may not avail against its development without the concept of artificial intelligence. By the combination of these two fields, the intelligent database system can be achieved. In the system, knowledge about the real world is represented by a commonly-used representation method, frame system. And the location network which has an explicit structure that its internal nodes means the operation on the decestors, is served as a evaluation mechanism of spatial reasoning.

We intend to discuss the practical implementation of spatial information interpretation by automatic mapping of frame to location network. That is to say, queries about the spatial information may be achieved by the evaluation of location net constructed by frame representation according to individuality of task. Fig. 4 shows the architecture of the system. We will show how location network when interpreted as so-called conceptual scheme can be successfully made to save as common denominator to two as yet disparate areas of artificial intelligence and the real-time graphical database management.

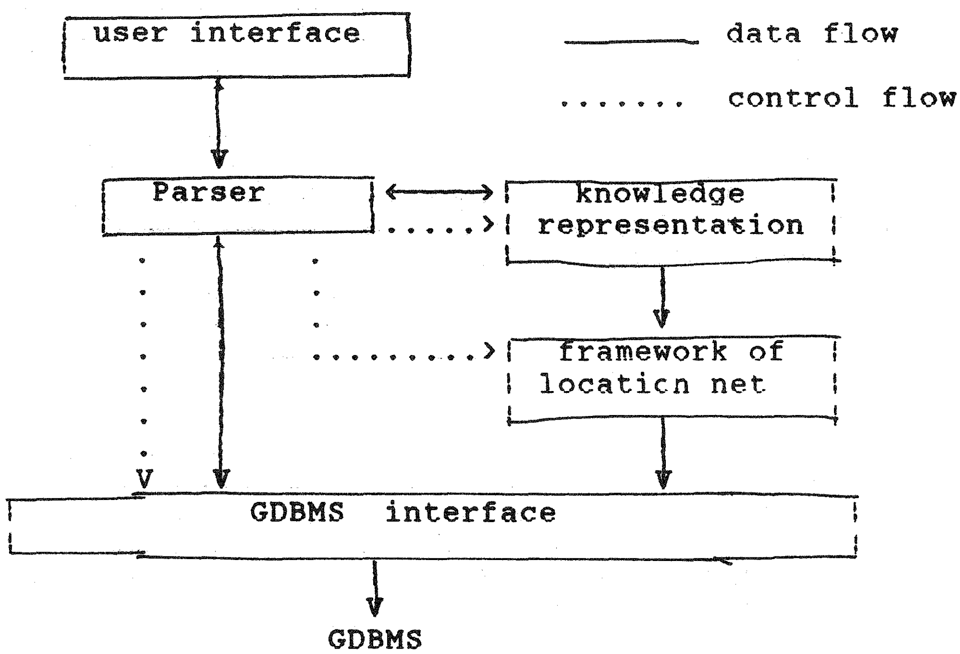

Fig. 4 Functional Diagram of the System.

Location network provides a method for associating geometric data with semantic net using the special evaluation rule. In location network, the expected relative locations of objects in the map are encoded into a network. The primitive sets of geometric relations between objects is made up of four different types of operations.

1). Directional operations (up, down, north, etc.)
2). Area operations (contains in, close to, etc.)
3). geometric operations (union, intersection)
4). Predicates which restrict the size of sets

The evaluation mechanism based on location network works in the top-down manner. During the evaluation process, every internal node in the network must have one of the following three states:

up-to-date, out-of-date or hypothesized. By start at the root node, process can use the data in the nodes which marks up-to-date or hypothesized, the data on out-of-date nodes are not available and its decestors must be evaluated until the up-to-date, hypothesized or the leaf nodes have been reached. When one leaf node is marked with out-of-date, the result may be not available, and this causes the reevaluation of the network.

Early uses of the location network mainly focused on image understanding, three-dimensional scene analysis. The prerequisite of using location network inference is that atoms of inference, e.g. the leaf nodes of the network, must be explicitly presented. This is hard to achieve in many applications such as above, bacause the pre-processing of scene is still an on-going research. But the problem is not exist in our database system. In the system, all the entities of the real world would be represented as clearly as possible.

CONCLUSION REMARKS

A object-oriented run-length data structure suitable for the geometrical data management and retrieval has been fully discussed. Algorithms for image algebraic operations on runlength coded objects, data format convertions and file organization in database system are presented. Runlength coding supports fast display of graphic data in raster monitor. The image algebra based on run-length structure can be performed parallelly. It has been shown that the computation of the operation is linear with respect to the number of terms in run-length codes. A prototype spatial database system KSDS (Knowledge-based Spatial Database System) has been developped in Image Processing Laboratory of the University of Science and Technology of China using runlength coding scheme. The experiment of KSDS has shown a good performance.

Objecte-oriented run-length coding may be expected to have a wide use in the field such as computer graphics, geographical imformation system, cartography, office automation, etc. It does not represent the ultimate answer to all spatial data storage and processing problems. It has, however, been demonstrated to possess advantages of both raster and vector data structures. Some investigation about theoretical issues and the logical inference scheme of the FSDS remain as the further research topics.

REFERENCES

1. H.Tamura & N.Yokoya, Image database systems: a survey, Pattern Recognition, Vol.17, No.1, pp29-43, 1984
2. H.Freeman, Computer processing of line-drawing images, Computing Surv. 6, 1974, p57-97.
3. Samet, H. A, A Quadtree-based Geographical Information System, in Auto-Carto, 5. Crystal city, 1983.
4. D.J.Peuquet, A Hybrid structure for the storage and manipulation of very large spatial data sets, CVGIP 24, p14-27, 1983.
5. Erland Jungert, Run length code as an object oriented spatial data structure, Proc. of IEEE workshop on Languages for Automation, Kent Ridge, Singapore, August 27-29, p66-70, 1986
6. Jan-Olof Eklundh, E. Jungert "An Approach to a Pictorial Information Management System". FOA Report C-30288-E1, Nov, 1982.
7. D. M. Russell, "Where do I look now?", Proc. PRIP, Aug 1979.
8. D. H. Ballard, C. M.Brown, Compouter vision, New Jersey: Prentice-Hall, 1982
9. J. Serra, Image Analysis and Mathematical Morphology, Academic Press, London, 1982.
10. Christopher F. Herot, Spatial Management of Data, ACM Trans. on Database Systems, Vol.5, No.4, Dec, 1980.