A DATA STRUCTURE FOR A RASTER MAP DATA BASE

Roger L.T. Cederberg
National Defence Research Institute
Box 1165, S-581 11 Linköping, Sweden.

Introduction

The increasing requirement for geographically referenced data in the public, private and military sectors, has led to technological advances in automated data collection, storage, manipulation and display of spatial data. A large fraction of the costs in Geographical Information Systems (GIS) can be associated with digitization, data encoding and input processing. The data encoding scheme also influences the storage efficiency and effectiveness of programs that analyse and manipulate the geographic data.

Each spatial data type (spatial data layer) has three possible types of geographic entities that must be encoded: points, lines and regions. The geographical entities are generally coded as either raster data or x, y-coordinate structures.

For raster data the entities are quantized and stored by the use of grid cells as location identifier (2). This implies that neighbourhood relationships are implicit, which facilitates analysis and manipulation. A disadvantage with raster data is the storage inefficiency due to redundancy in data.

The coordinate structures describe a point entity by its coordinate, a line by a chain of uniformly or non-uniformly spaced coordinates and a region by its boundary (lines). The coordinate oriented systems are

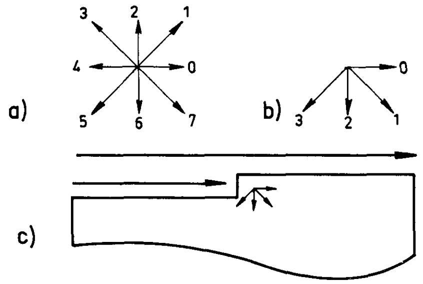

FIG.1. (a) Chain-code. (b) RC-code. (c) A contour can only be traced in four directions with the raster scan technique.

suitable for region and contour maps and have high storage efficiency.

This paper presents a data structure or encoding scheme which is intended for raster scan techniques (3). The code can be used for encoding and decoding of binary entities with high storage efficiency. The code can also be stored as a data structure, where the entities can both be accessed as raster data and as regions, boundaries and lines.

Raster-scan Chain-code

The conventional chain-code lists each object as a *x*, *y*-coordinate and a sequence of chain-links that defines the border of the object. The code contains eight directions and the entire object must be accessible during the tracing of the border. If we trace the contour stepwise in a raster-scan fashion, then only four directions can be registered for an 8-connected object (FIG.1). But more than one coordinate must normally be listed for each object. These coordinates are the first pixels that the raster-scan procedure hits in some part of the contour. Each listed coordinate, called a max-point, is connected to two R-chains (FIG.2). A min-point is the terminal of two R-chains. A number of max-points with associated R-chains define an object.

The following sections describe encoding and decoding with Raster-scan Chain-code (RC-code).

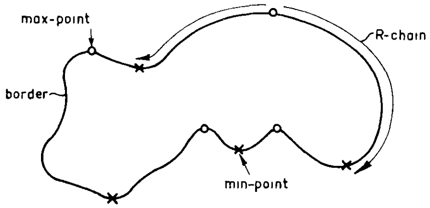

FIG. 2. A border with indicated max-points and min-points.

Coding of Regions

According to the sampling theorem the sampling interval has to be half the size of the smallest features to be encoded. Therefore the regions should not have peninsulas of one pixel width. (The preprocessing eliminates· quantization noise).

The max-points can be detected either with a set of 3 by 3 templates (3) or by the use of pointers to all the R-chains of the current line. In the latter case, a max-point is registered when a border is encountered that is not marked by a chain-pointer. The attached chain-links are detected with ordinary border-following techniques. This procedure is easier to implement on conventional computers.

There are basically two methods to store the RC-code. While the extraction is strictly ordered, all the links can be stored on a stack in the same order as they are found. The max-points are stored on a separate stack. In rewriting mode, the max-point stack is poped when the coordinate given by the max-point is reached. The max-point is replaced in the image and the two links at the top of the chain-link stack is attached to the max-point. For each new generated line, the chain-link at the top of the stack is attached to the next R-chain on the line, the stack is poped and so on. We can keep track of the R-chains from line to line with pointers. Two R-chains that are meeting indicate a min-point.

To be able to access the coded image both as raster data and as regions, each R-chain must be stored as a unit. Several chains are built concurrently, so the endings of these chains must be accessible and there

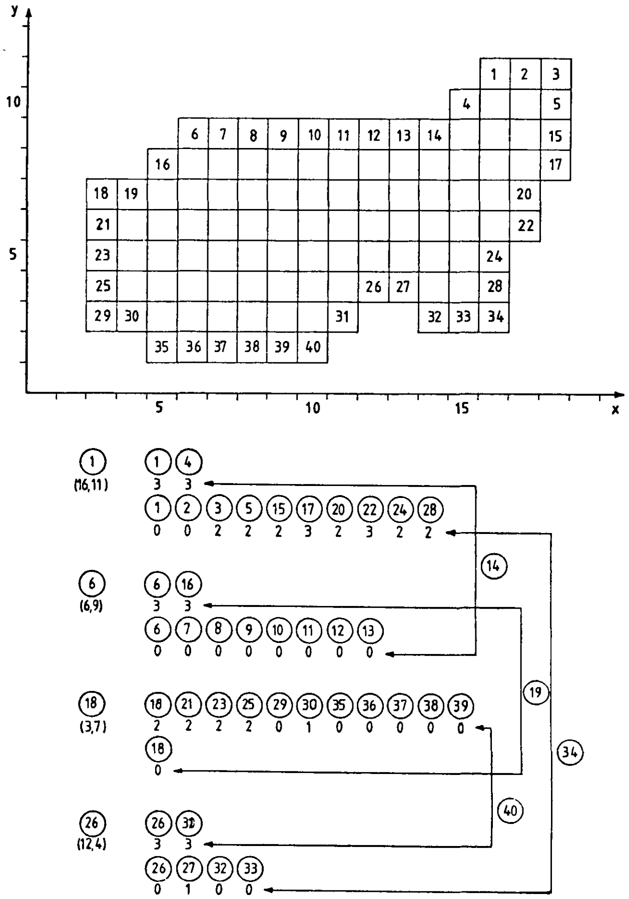

FIG. 3. Encoding of the border step by step. The numbers within rings correspond to the numbers on the border.

has to be space to append more links. Therefore, the chains are built in a buffer storage and depending on the size of the buffer, completed regions, chain-pairs or if a chain becomes extremely long, a part of a chain are transferred to secondary storage. The max-points with pointers to the R-chains are stored separately in a dictionary. In rewriting mode, the max-points are replaced as above. The chain-links must be retrieved from the list of chain-links at positions indicated by the pointers associated with the max-points. The encoding using this code is exemplified in figure 3.

The nesting of objects and holes, objects within holes and so on, can be revealed concurrently with the encoding. The principle is basically the same as for component labelling with the run-tracking method (4). The max-points introduce new possible interpretations, and min-points solve the ambiguities (3).

Coding of lines

The digitized lines are thinned to one pixel width and then encoded. The lines can be non-intersecting closed contour lines (elevation data) or a general graph with nodes and branches (region map or road map). As before, the code can be stored only for retrieval and display or stored to be accessible both as raster data and line entities. The non-intersecting closed contour lines do not change anything from the previous section except that the interior of a border is not filled with *1's* in the rewriting mode.

The general graphs are based on single lines (branches) and nodes,so we must introduce single chain max-point and min-point, called respectively start-point and end-point (FIG. 4). For the simple raster format encoding, four stacks must be stored: start-point, max-point, end-point and the stack of chain-links. The min-points are, as before, detected automatically during reconstruction. The start-point indicates the beginning of a single chain or a fork. An end-point indicates the end of a single chain or when two chains join to one.

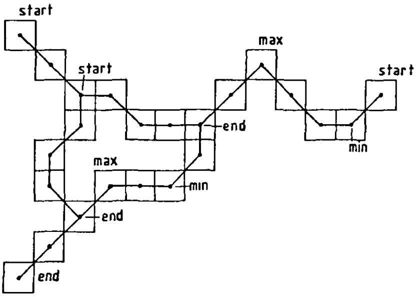

FIG.4.A unit width graph with start-points and end-points

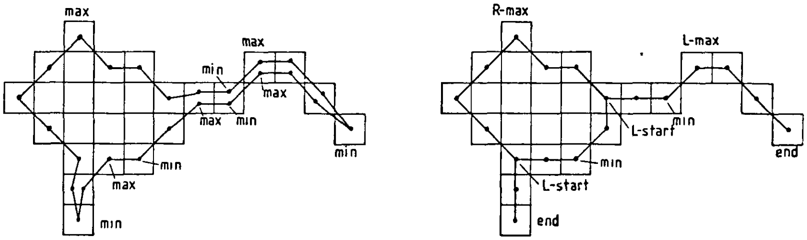

FIG. 5. The extended region encoding and the combination of region and line encoding.

To be able to access the image both as a graph and line by line in raster mode, the max-points and nodes are stored and the associated R-chains are added link by link. As before, this implies that several chains must be accessible concurrently and special routines for transfer of chains to secondary storage are needed.When a chain terminates, in a min-point or node, interconnection pointers are established.

Coding of Regions and Lines

A quantized image without preprocessing and an image with areas and thinned lines cannot be coded with the methods given above. To allow both regions and lines of one pixel width, the region encoding must be extended or combined with the line encoding. The extended region encoding assigns two R-chains to a line as if it was a region. If it is infrequent it can be accepted, but in an image composed of areas and several lines of unit width, it is better to combine the two methods. This implies that two types of max-points and start-points are involved, region max-point (start-point) and line max-point (start-point). The difference between the methods are exemplified by figure 5.

RC-code for Display

One of the encodings discussed so far can be retrieved as regions and lines or line-by-line for processing and display. The display can only be rolled in the scan direction, which makes it impossible to move the display window freely in the data base. To be able to roll the map both up, down, left and right, the RC-code can be extended. In this case, max- and min-points have to be stored for both the x and y-directions, so four co-

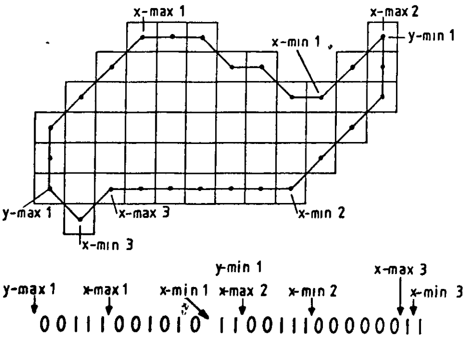

FIG. 6. RC-code for display.

ordinate lists including pointers and the chain-links are stored. This reduces the storage efficiency considerably but the situation can be improved by using relative coordinates within the x-max, x-min, y-max and y-min list for an object and because only one bit is needed for storage of the chain-links (FIG. 6). We can also transform the usual chain-code or RC-code to the display version of RC-code.

Conclusions

There are two major approaches to internal data organization in geographic information systems: raster data and x, y-cordinate/linked structures. Raster data has the advantage of direct correspondence to the format of a raster-scanner input and matrix plotter output. Processing related to distance and retrieval based on location can easily be accomplished. Software development is easier for many applications. The major disadvantage with raster data is storage inefficiency. The linked structures, on the other hand, are storage efficient and correspond directly to the format of coordinate digitizers and incremental plotters. It is suitable for processing polygonal maps and contour maps which have the property that lines do not meet. The disadvantages are difficulties of performing distance-related operations and set algebra.

The RC-code is an attempt to combine the advantages of the two schemes. RC-code corresponds to both raster and incremental input/output and it is storage efficient. The code is suitable for processing of polygonal and contour maps. One disadvantage is that, due to the data compression, processing related to distances cannot be computed directly from the storage addresses. Another minor problem is the buffer management during storage and retrieval of the code.

We are currently investigating the usefulness of the RC-code in a project where overlays from the Swedish topographical map are digitized and encoded. Encoding of region entities like lakes and forests have been implemented and we are now working with line entities.

References

1. T.K. Peucker and N. Chrisman. "Cartographic Data Structures", The American Cartographer, Vol. 2., No. 1., April -75.

2. H. Freeman, "Computer Processing of Line-Drawing Images", Comp. Surveys, Vol. 6, No. 1, March -74.

3. R. Cederberg, "Chain-Link Coding and Segmentation for Raster Scan Devices", CGIP, 10, 1979.

4. A. Rosenfeld, "Connectivity in Digital Pictures", J Assoc. Comput., March 17, 1970.

5. G. Nagy and S. Wagle, "Geographic Data Processing", Comp. Surveys, Vol. 11, No. 2, June 1979.

6. D.L. Milgram. "Constructing Trees for Region Description", TR-541, Univ. of Maryland, June -77.