UC Santa Barbara

Core Curriculum-Geographic Information Systems (1990)

Title
Unit 31 - Efficient Storage of Lines - Chain Codes

Permalink
https://escholarship.org/uc/item/0j4408cz

Authors
Unit 31, CC in GIS
Douglas, David H.

Publication Date
1990

Peer reviewed

UNIT 31 - EFFICIENT STORAGE OF LINES - CHAIN CODES

UNIT 31 - EFFICIENT STORAGE OF LINES - CHAIN CODES

Compiled with assistance from David H. Douglas, University of Ottawa

UNIT 31 - EFFICIENT STORAGE OF LINES - CHAIN CODES

Compiled with assistance from David H. Douglas, University of Ottawa

A. INTRODUCTION

- the previous unit looked at structures which encode certain types of spatial relationships
  - in this unit we look at the coding of the geometry of lines and areas

  overheads - Examples of geographic lines
  - note the different characteristics of each of these lines

- each of these lines suggests different geographic features - can you identify them?

- most systems store lines and areas as sequences of points connected by straight lines
    - is simple, but best for representing lines with sharp breaks of direction, not smooth curves
        - for example, a meandering river or a railroad could be represented very much more efficiently as a few smooth curves than as a large number of straight lines
        - this could greatly reduce the effort of digitizing

- there are many methods of discretizing a line
    - this unit looks at some of them, and the applications where they are useful

Terms

- because area objects are mostly represented by straight line segments they are often referred to as polygons

- a variety of terms are used to describe an irregular line coded as a sequence of straight line segments, particularly when the line is the common boundary between two area objects
    - these terms include arc, segment, edge, and chain
    - of these, chain has been established as the standard terminology for US digital cartography when the line is a common boundary, but arc is probably used most often

B. TECHNIQUES FOR REPRESENTING IRREGULAR LINES

1. Straight line segments

- if the line on the ground is curved, e.g. a river bank, then straight line segments can only approximate the truth
    - the degree of approximation depends on the length of straight segment used, but the segments would have to be infinitely short to represent a true curve

- thus, straight line segments are a way of representing a continuous line using a finite amount of information, often the coordinates of the segment endpoints

2. Arcs of circles

- some systems oriented toward survey data allow both straight lines and arcs of circles between points

- for curves, the system must store the radius of curvature as well as the start and end points
    - it must also correctly identify those segments which are straight lines and those which are arcs

- this approach is used in Prime's System9 GIS

- is useful for engineered features, like highways and railroads, which are designed as straight lines and curves

3. Splines

- can be used for describing smoothly curving features like meandering rivers
    - the curve is modelled as a spline function, a mathematical function that passes through specified points but has minimal curvature

- splines are often used to smooth the curves drawn by plotters as part of the contouring operation

- are more commonly used in CAD than in GIS

C. STORING CHAINS (ARCS)

- there is often significant redundancy when a line is stored as a sequence of coordinate pairs
    - for example, a curved street represented by four points in Columbia, MO might have coordinates as follows:

        (38.9519, 92.3503)

        (38.9519, 92.3510) (38.9522, 92.3511) (38.9522, 92.3527)

    - note that the first four digits of each coordinate (latitude,longitude) are the same for every point

- can economize greatly on storage by storing the offset from the previous point, in units of 0.0001 degrees:

    (38.9519,92.3503) (+00,+07) (+03,+01) (+00,+16)

    - this would allow every subsequent point to be stored in 4 decimal digits instead of 12, or roughly 12 bits instead of 36
        - also need one bit each to store the signs of the change in longitude and latitude,
        - for a total of 14
    - however this will limit the maximum change between any pair of points to .0099 degrees latitude or longitude

Advantages and disadvantages

- the advantage is reduction in storage volume

- the disadvantage is loss of generality:
    - we now have a fixed spatial resolution (.0001 degrees) and maximum distance between points (.0099 degrees latitude or longitude)
    - this creates problems when using global references and converting coordinates

- variations on this technique are particularly applicable for data obtained from scanners and digitizers
  - these devices have built-in resolution
    - scanners have a resolution equal to the pixel size of the scanner, and all displacements between points are actually whole numbers of pixels
    - digitizers also work in discrete units

Freeman chain code

- instead of strings of coordinate pairs, systems which rely heavily on scanner input often code lines as lists of incremental movements
  - a fixed number of move directions is established, usually 8, and assigned the integers 0 through 7 (the "Queen's case" move set):

  diagram

  - a series of four moves up would be coded as four 0s

- the string "01012" would be used to code the following curve:

  diagram

- each step along the line requires a digit between 0 and 7, or three binary digits between 000 and 111

- this technique of incremental coding of lines or chain code is usually associated with Herbert Freeman
  - however, it was described much earlier in the 19th Century by Francis Galton as a means of transmitting information on fingerprints over a telegraph

- see Freeman (1961) for description of a number of algorithms based on chain codes

Compressing code

- certain pairs of codes are much more likely to occur in chain codes than others
  - lines generally turn slowly, and sharp bends are unusual in much geographic data
    - the next move is most likely the same as the previous one
    - next most likely is 45 degrees right or left
    - 180 degrees is the least likely next move
  - thus, a code 4 is most likely to be followed by another 4, or a 3 or a 5, and least likely to be followed by a 0, 1 or 7

- this can be exploited by using a code whose length depends on the sharpness of the turn angle:

  1 ahead 01 45 degrees right 001 45 degrees left 0001 90 degrees right 00001 90 degrees left 000001 135 degrees right 0000001 135 degrees left

- in this way, the total number of binary digits needed to represent a line can be reduced

overhead - Compressing code

Repeating sequences

- if the line being coded has long straight stretches, sequences of chain codes will be repeated

  overhead - Repeating sequences

- for example, to go from one point to the other requires 6 moves to the east and 3 to the northeast (3 1s and 6 2s).
    - a sequence such as 222222111 would be a serious distortion of the straight line
    - the straight line is best approximated by the sequence 221221221
        - repeats the 221 pattern 3 times

- straight is always best approximated by a homogeneous mix of codes

- to code lines with long straight sections, a way of representing runs of patterns is needed
    - for example 3(221) might indicate four repeats of the 221 sequence

Summary

handout - Comparison of different encoding schemes

D. APPLICATIONS OF CHAIN CODES

- chain codes are useful in two particular applications:

Vectorizing raster data

- output from raster data along the boundary separating zones A and B would appear as follows:

  A A A B A A B B A B B B B B B

- a line can be assumed along the pixel edges which separate the A's from the B's
    - a vector representation of the line could be coded in chain code as 202020

Vectorizing scanner output

- when a line is read by a scanner it looks something like this:

  0 0 0 1 1 1 0 0 1 1 1 0 0 1 1 1 0 0 1 1 1 0 0 0

    - 1's indicate where the scanner "saw" the line and 0's where it "saw" the blank spaces around the line

- the first step in deriving a vector representation of the line is to "thin" the pixels using a standard algorithm

- are many different algorithms available for this
- a good source on these is Pavlidis (1982)

- the result of thinning will be:

  0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0

- the line can then be derived by following the remaining skeletal 1s
  - represented in chain code as 111 (three moves northeast)

  - usually stored with the x,y coordinates of the line followed with the chain code

- chain code is most useful in applications where vector data originated in raster form, from a scanner or an image
  - since it discretizes both distance and direction (both can take a limited set of possible values), it is not particularly good for more general applications

REFERENCES

Freeman, H., 1961. "On the encoding of arbitrary geometric configurations," Institute of Radio Engineers, Transactions on Electronic Computers, EC10:260-8.

Pavlidis, T., 1982. Algorithms for Graphics and Image Processing, Springer-Verlag, Berlin.

DISCUSSION AND EXAM QUESTIONS

1. Using a sample line drawn on a sheet of paper, write out (a) the Freeman chain code representation of the line using single digits ranging from 0 to 7, (b) its representation using the binary code based on turn angles, and (c) its representation using run encoded chains. Which option gives greater data compression? (express the length of each code in bits, and remember that a number between 0 and 7 requires 3 such bits) (need to include a drawn example).

2. The answer to the previous question depends on the type of line being coded. Discuss the characteristics which would give greatest data compression for each of the three options, and describe examples of lines with these characteristics.

3. The accuracy with which a line is represented as a series of straight line segments depends on the process used to select points during digitizing. Discuss the criteria you would use to select points in order to obtain an accurate representation of a line. How would these criteria change depending on the nature of the line?

4. Using a simple classified raster, write out a vectorized representation using arcs and Freeman chain code. Include the left and right polygons and pointers to adjacent arcs as attributes of each arc, and the classes as attributes of each polygon. Show each line as a coordinate pair for the

beginning point, followed by a chain code using digits in the range 0 through 7.

5. In preparation for the Unit 32, figure out how you would set up a program to determine where two lines, defined by their end points, intersect. Start with a diagram. Under what conditions will your program fail?

Last Updated: August 30, 1997.

UNIT 31 IMAGES

Left
00001

0000001.   .001

Back 00000001 —————————————— 1 Forward

000001   01

0001
Right

— ╱ ╱ ╱ ╱ — — ╲ ╲ ╲
2 1 1 1 1 2 2 3 3 3
1 001 1 1 1 01 1 01 1 1

Line 1

A●   ●B

Line 2

A●   ●B

Line 3

A●   ●B

Line 4

A●   ●B