# AREA CALCULATIONS USING PICK'S THEOREM ON FREEMAN-ENCODED POLYGONS IN CARTOGRAPHIC SYSTEMS

Mark R. Anstaett
and
Harold Moellering
Department of Geography
The Ohio State University
Columbus, Ohio 43210

## ABSTRACT

Pick's Theorem provides a means for calculating areas of closed figures formed by connecting points of a regular grid (Coxeter, 1961). The basic equation is

$$AREA = GI + 1/2*GB - 1$$

where GI is the count of grid points inside of the figure and GB is the count of points on the figure's perimeter. The application of Pick's Theorem on polygons represented by various chain codes is the primary topic of this paper. Two grid structures, those formed by corner points of rectangular and of hexagonal cells, and four Freeman-type encodings, 4 and 8-way on the rectangular grid structure and 6 and 12-way on the hexagonal grid structure, are those to which Pick's Theorem is specifically applied. A geometric proof of the proper area calculation through the original equation for 4, 6, and 8-way encodings is given. A revised Pick's Theorem for the grid formed by all 6 corner points of hexagonal cells allows application to 12-way encoded polygons. A proof is also given for this new equation. Through sequentially processing the Freeman codes, an algorithm produces the grid point values which are used in the area calculation. This algorithm is shown to correctly handle polygons with holes and pinched perimeter.

## INTRODUCTION

The computation of polygonal areas is a common operation in cartographic systems. The method of area calculation employed is dependent to some extent on the data format. A formula which was proven long before the days of computer-assisted cartography, Pick's Theorem, calculates areas of polygons whose vertices are points in a regular grid. The basic equation is

$$AREA = GI + 1/2*GB - 1 \tag{1}$$

where GI is the count of grid points inside of the polygon and GB is the count of grid points in the polygon's perimeter (Coxeter, 1961). Cartographic data originating from a variety of vector or raster formats, when quantized into a regular grid, can be efficiently stored as chains of neighboring grid points. Lines and polygonal boundaries are then represented as sets of integer codes which describe the line direction between points (Freeman, 1961). This representation has been called chain or starburst codes, but is commonly referred to as Freeman encoding. A particular Freeman encoding format is derived from the grid geometry and chosen number of angular directions from a grid point. Storage and processing efficiencies and computational and positional errors with use of Freeman encoding versus other data formats must be studied in building a case for or against use of this method with cartographic data. The purpose of this paper, however, is to show how Pick's Theorem can be applied for area calculation of polygons represented by four chain code formats. The four chosen formats are 4 and 8-way encoding in a rectangular grid and 6 and 12-way encoding in a hexagonal grid.

## THEORY

Although Pick's Theorem is a simple equation for area calculation, its proof and application seem to be relatively obscure. The proof of the theorem given in (Coxeter, 1961) involves decomposing polygons into simpler ones. The area equation for a single polygon has been extended and proven to apply to a polygon with n holes (Sankar and Krishnamurthy, 1978). Applications of Pick's Theorem to 4 and 8-way encoded polygons have been shown without proofs (Rosen, 1980 and Kulpa, 1977). A simple proof of Equation 1 based on the geometry of Freeman encoding can be stated informally. Examples of 4, 8, and 6-way encoded polygons and their encoding schema appear in Figure 1. The 4 and 8-way encodings use a grid whose points are corners of rectangular cells, while grid points for 6-way encoding are 3 of the 6 points of hexagonal cells. Each grid point "owns" a Thiessen polygon which is identical in shape and area to the cells of the grid structure. To the right of each example is a grid point with its Thiessen polygon (dashed lines) and n-way chain codes. For grid points located within the polygon boundary, counted in GI of Pick's Theorem, an area contribution of one is made for its Thiessen cell. Considering points on the polygonal boundary, those counted by GB of Equation 1, a straight line of two equal chain codes crossing a grid point bisects that point's Thiessen cell. The area expectation for a GB point is then (1/2) cell. Where the chain code changes at a grid point, there is deviation from this expected area. As seen in the right half of each Figure 1 example, the n chain codes at a grid point cut its Thiessen cell into sectors with equal areas of (1/n) cell. For each unit change in chain code at a point, an area deviation of ($\pm$1/n) cell from the (1/2) cell expectation is produced. Assuming that Freeman chains circle polygons in a clockwise direction, polygon closure requires a net total of n clockwise code changes within the chain. The result of this is a subtracted area of n*(1/n) cells from the straight-line expectation of (1/2)*GB cells. This corner-correction value becomes the (-1) term of Equation 1, completing the informal proof.

The 12-way chain code is shown in Figure 2(a) in a polygon example, using the code scheme from (Scholten and Wilson, 1983) with X and Y grid orientations reversed. Figure 2(b)

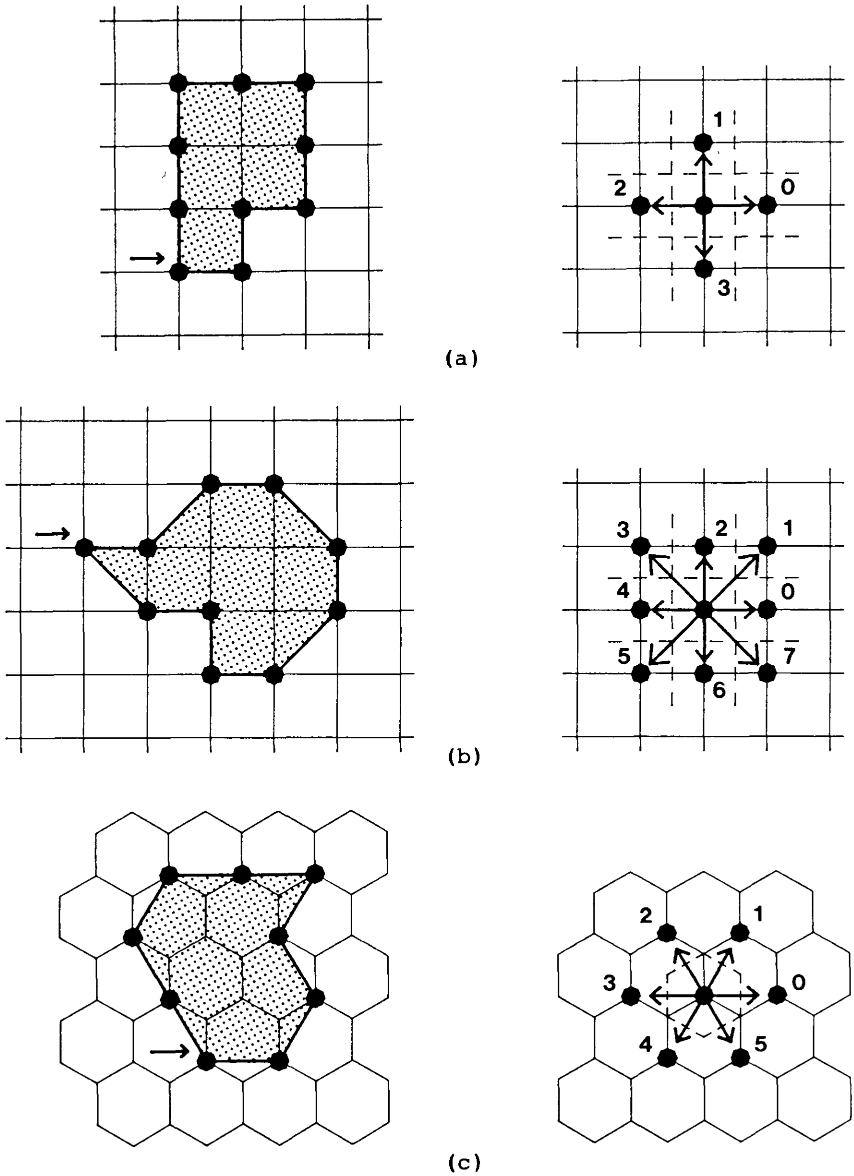



Figure 1. Examples of 4, 8, and 6-way encoded polygons.
Chain coding key is in right half of each figure.
(a) 4-way encoding:
polygon code chain=(1,1,1,0,0,3,3,2,3,2)
(b) 8-way encoding:
polygon code chain=(0,1,0,7,6,5,4,2,4,3)
(c) 6-way encoding:
polygon code chain=(2,2,1,0,0,4,5,4,3)

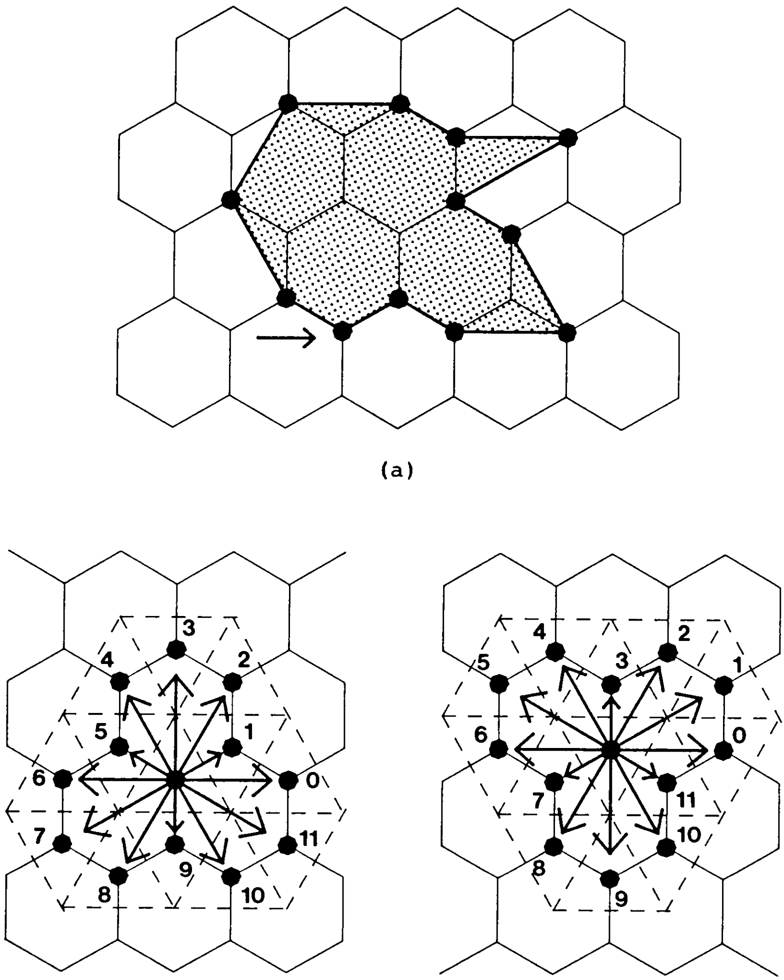



Figure 2. 12-way encoding.
(a) 12-way encoded polygon:
code chain=(5,4,2,0,11,0,7,11,10,6,5,7)
(b) Chain code directions from upper-tier grid point. Thiessen triangles are shown in dotted lines.
(c) Chain code directions from lower-tier grid point. Thiessen triangles are shown in dotted lines.

illustrates the chain codes at a grid point which is at the top of a vertical hex cell edge (upper-tier point), while Figure 2(c) shows the 12 codes at a lower-tier grid point. Thiessen polygons created through connecting perpendicular bisectors of a point with its 12 neighbors make overlapping tri-hexagonal shapes, then bisecting the area of overlap leaves the triangular Thiessen cells shown in dotted lines for each of Fig. 2(b) and (c). It is evident that a straight line of two equal chain codes crossing a grid point does not in all cases bisect the Thiessen triangle, and that the sector areas of deviation where codes change are not equal for all codes. For these reasons, 12-way encoding requires a different formulation of Pick's Theorem. The Pick's Theorem equation for 12-way Freeman encoding is

$$AREA=1/2*(GI+1/2*GB_{ODD}+2/3*GB_{EVEN,UPPER}+1/3*GB_{EVEN,LOWER}-1) \quad (2)$$

where GI is the count of grid points inside of the polygon, GB(ODD) is the count of odd-valued 12-way codes in the chain, GB(EVEN,UPPER) is the count of even-valued codes in which the centroid of the hexagonal cell crossed by that code lies to the right of the code vector (inside of the polygon), and GB(EVEN,LOWER) is the count of even-valued codes in which the centroid of the hexagonal cell crossed by that code lies to the left of the code vector (outside of the polygon). For the proof of this equation, first consider that the equation counts Thiessen triangles, then gives area in terms of hex cells of the grid. The (1/2) factor scales the count of triangles to hex cells, as this is their area ratio. Each grid point located within the polygonal boundary, counted in GI, contributes one to the area. A straight line of two odd codes crossing a point bisects its Thiessen triangle, thus the 1/2*GB(ODD) term. Assuming polygon interior to the right of the chain vectors, straight-line GB(EVEN,UPPER) codes contribute 2/3 of a Thiessen triangle (codes 0,4,8 in Figure 2(b) and 2,6,10 in Figure 2(c)), while the remaining straight-line even code situations, counted in GB(EVEN,LOWER), contribute an area of 1/3. As in the proof of Equation 1, polygon closure requires a net total of 12 code changes in a clockwise direction, and each code change produces an area deviation from the expected straight-line GB area for the first of two codes. To prove that the total area deviation for a polygon equals one Thiessen triangle, one must consider an algorithm which marks sectors of the Thiessen triangle when that sector is the area deviation corresponding to a particular change of code. The starting grid point of a polygon sets the base triangle in either the Fig. 2(b) upper-tier or Fig. 2(c) lower-tier grid point situation. As the polygon chain is cycled, Thiessen triangle sectors are marked for clockwise code changes (GB area is less than straight-line expectation for the code) or unmarked for counterclockwise code change (GB area is more than the straight-line expectation). The grid point tier changes for each odd code, but the chain ends at the same tier as it begins. Area deviation sectors for a given odd code:odd code change are the same total area in either tier. Thus all code changes mark sectors in the base triangle or mark sectors between pairs of odd codes in the other tier's triangle, which equal the area of those same sectors in the base triangle. The result is that all sectors of the base triangle become marked, which is a total area deviation of (-1) from the straight-line area expectations for GB points. This is the term (-1) in Equation 2, completing the proof.

## DESCRIPTION OF ALGORITHM

An algorithm has been written and implemented in a FORTRAN 77 program which computes areas of 4, 6, 8, or 12-way Freeman-encoded polygons by use of the Pick's Theorem Equations 1 and 2. Constraints on the input polygons are that they be simple closed figures, cycled clockwise (polygon interior to right), and that holes are linked to the perimeter by chain codes. Chain pre-processing removes zero-area peninsulas (detected by 180° change of direction in consecutive codes) and rotates the chain so that the last code has a nonzero X-component. A loop processes each chain code by breaking the code into X and Y grid coordinate components, updating the (X,Y) of the code vector destination, and incrementing GI and GB point counts. The function NC(Y) provides a count of grid points in a vertical column below current coordinate Y to a base level Y=0, and NCOFF(Y), used in the hexagonal grids, counts the column below (X$\pm$1,Y) for current grid point (X,Y). The actions taken for specific codes are shown in Table 1. After the last chain code has been processed, GB is set to the chain length, and GI is the chain length plus the number of internal grid points minus a correction factor for areal overcount by GB. This areal overcount may occur when GB counts every code while more than one code reaches a given grid point. The original Pick's Theorem works for some of these cases because such revisited grid points are visually counted only once, while in sequential processing of the codes no memory is kept of which grid points were reached. The equation

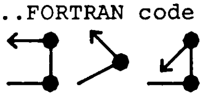
$$GI' = GI - GB \tag{3}$$

produces the GI value for insertion into area Equations 1 and 2 which corrects for areal overcount in GB. The reader may wish to confirm from the situations given in Table 1 that when the chain revisits a point in such a way that the polygon area lies between the code vectors which meet the point, the algorithm only increments GI once while GB will count two, giving a net area correction of (-1) at this grid point by Equation 3. Holes in polygons are correctly handled by being cycled counterclockwise and linked to the polygon perimeter by chain codes; they are then made up of points counted in GB, and interior grid points within the hole become points external to the polygon, which are subtracted out in processing the top edge of the hole. For 12-way chains, the algorithm also maintains the three separate GB counts of Equation 2 by keeping track of upper/lower tier of grid points in the chain. The result of each polygonal calculation is the area in terms of grid cells, which can be directly converted to square user coordinate values by applying cell dimension information.

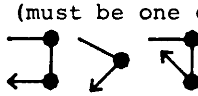
```
(situation and update of GI)          (description)

IF(DX.GT.0) THEN
  GI=GI+NC(Y)+1                         add column below and
                                          including current point
  IF(DX.EQ.2) GI=GI+NCOFF(Y)            add column passed over at
                                          (X-1) in hex grid
  IF(.NOT.XFLAG) THEN                   change of X-direction found
    IF(...FORTRAN code to test for
```

```
                                  cases)
                                        convex curve:
      GI=GI+NC(LY)+1                      add column below and
                                          including last point
    ELSE (must be one of
```

```
                                      cases)
                                        concave curve:
      GI=GI+NC(LY)                        add column below
    ENDIF                                 last point
  ENDIF
ELSE IF(DX.LT.0) THEN
  GI=GI-NC(Y)                           subtract column below
                                          current point
  IF(DX.EQ.-2) GI=GI-NCOFF(Y)           subtract column passed over
                                          at (X+1) in hex grid
  IF(XFLAG) THEN                        change of X-direction found
    IF(...FORTRAN code to test for
```



```
                                  cases)
                                        concave curve:
      GI=GI-(NC(LY)+1)                    subtract column below and
                                          including last point
    ELSE (must be one of
```



```
                                      cases)
                                        convex curve:
      GI=GI-NC(LY)                        subtract column below
    ENDIF                                 last point
  ENDIF

ELSE IF(XFLAG.AND.DY.GT.0) THEN
```

```
                                        (   ) DX=0,
  GI=GI+1                                     add in current point

ELSE IF(.NOT.XFLAG.AND.DY.LT.0) THEN
```

```
  GI=GI+1                               (   ) DX=0,
                                              add in current point
ENDIF
```

Table 1.
Update of GI for each code situation, in which
(DX,DY) are X and Y grid components of current chain code,
(X,Y) are grid coordinates at end of current code vector,
LY is Y coordinate of origin of current code vector,
XFLAG=.TRUE. if last nonzero DX was greater than zero,
     =.FALSE. if last nonzero DX was less than zero,
NC(Y) is count of grid points in column below (X,Y),
NCOFF(Y) is count of grid points in column below X±1,Y).
Both NC and NCOFF use vertical column to and including
base level Y=0.

## EXAMPLES

A set of four examples illustrate the algorithm's proper handling of 4, 6, 8, and 12-way encoded polygons. Figure 3 is a simple 4-way encoded polygon. Table 2 lists specific values computed during the sequential processing of the chain (see Table 1 for details) with the resulting values inserted into Equations 3 and 1 for the area in terms of cells. Figure 4 shows a 6-way encoded polygon containing a concavity, and Table 3 traces the area computation for it. An 8-way encoded polygon with a concavity and a pinched perimeter, that situation which Equation 3's modified GI value corrects for, is given in Figure 5 with the corresponding trace of area computation in Table 4. The last example, Figure 6, is a 12-way encoded polygon with a pinched perimeter and a hole. Table 5 lists the important values of the algorithm's execution, including the specific GB values which go into Equation 2.

## SUMMARY AND CONCLUSIONS

The practical application of Pick's Theorem, an area calculation method for regular point grids in general, to specific grids used with Freeman encoding has been illustrated by this paper. The theorem was proven geometrically for 4, 6, and 8-way encodings. The modified area equation was introduced which allows a similar calculation on the irregular point grid of 12-way encoding, and this equation was also informally proven. The algorithm sketch describes an implementation of these area calculations on properly-encoded cartographic polygons, and the utility of the algorithm on holes and pinched perimeters has been illustrated.

While Freeman encoding may not be in widespread use with cartographic data, it does offer several advantages. The grid point-based structure allows easy integration of point, line, and area data into a single coordinate scheme. The chain codes themselves represent a vector format within a cellular structure, which permits data conversion between Freeman encoding and both vector and raster data. This quality of Freeman encoding, combined with its capacity for data volume compression, support its use in the new hybrid vaster data structure (Peuquet, 1983). Due to the versatility in dealing with various grid formats and the sequential nature and resolution-independence of the processing, the application of Pick's Theorem to Freeman-encoded polygons enhances the use of these data formats for cartographic data.

## REFERENCES

Coxeter, H.S.M. 1961, Introduction to Geometry, pp. 208-210, John Wiley and Sons, Inc., New York

Freeman, H. 1961, On the Encoding of Arbitrary Geometric Configurations: IRE Transactions on Electronic Computers, June 1981, pp. 260-268

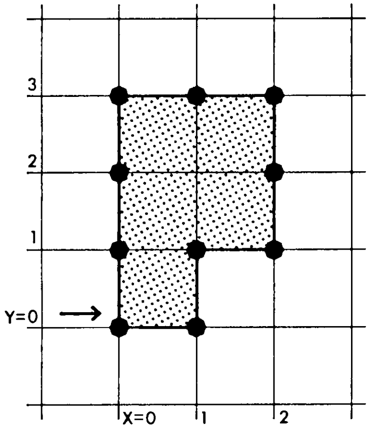



Figure 3.
4-way encoded polygon:
code chain=(1,1,1,0,0,3,3,
2,3,2), GB=10.

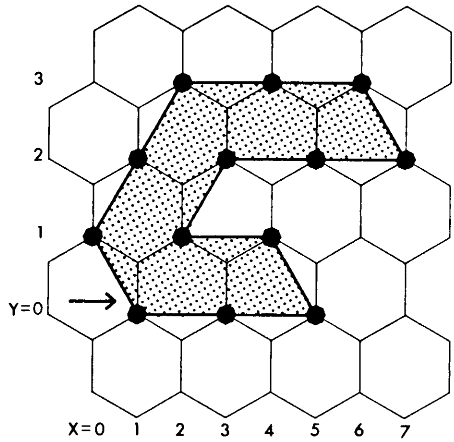



Figure 4.
6-way encoded polygon:
code chain=(2,1,1,0,0,5,3,
3,4,0,5,3,3), GB=13.

Table 2. Trace of algorithm execution for Figure 3 polygon.

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| CODE | 1 | 1 | 1 | 0 | 0 | 3 | 3 | 2 | 3 | 2 |
| DX | 0 | 0 | 0 | 1 | 1 | 0 | 0 | -1 | 0 | -1 |
| DY | 1 | 1 | 1 | 0 | 0 | -1 | -1 | 0 | -1 | 0 |
| GI | 0 | 0 | 0 | 8 | 12 | 12 | 12 | 10 | 11 | 11 |

GI' = (11) - (10) = 1 (Eq. 3)
AREA = (1) + 1/2*(10) - 1 = 5 (Eq. 1)

Table 3. Trace of algorithm execution for Figure 4 polygon.

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CODE | 2 | 1 | 1 | 0 | 0 | 5 | 3 | 3 | 4 | 0 | 5 | 3 | 3 |
| DX | -1 | 1 | 1 | 2 | 2 | 1 | -2 | -2 | -1 | 2 | 1 | -2 | -2 |
| DY | 1 | 1 | 1 | 0 | 0 | -1 | 0 | 0 | -1 | 0 | -1 | 0 | 0 |
| GI | 0 | 3 | 5 | 9 | 13 | 15 | 12 | 10 | 10 | 12 | 13 | 13 | 13 |

GI' = (13) - (13) = 0 (Eq. 3)
AREA = (0) + 1/2*(13) - 1 = 11/2 (Eq. 1)

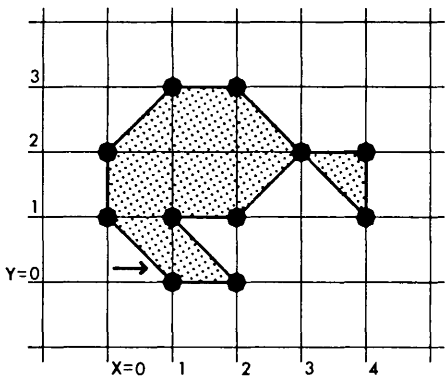



Figure 5.
8-way encoded polygon:
code chain=(3,2,1,0,7,0,6,3,5,4,7,4), GB=12.

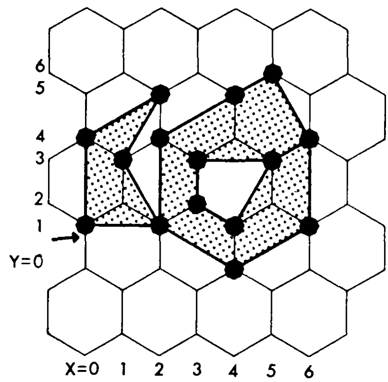



Figure 6.
12-way encoded polygon:
code chain=(3,1,8,10,3,1,1,10,7,6,9,11,2,1,9,7,5,6),
GB=18.

Table 4. Trace of algorithm execution for Figure 5 polygon.

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CODE | 3 | 2 | 1 | 0 | 7 | 0 | 6 | 3 | 5 | 4 | 7 | 4 |
| DX | -1 | 0 | 1 | 1 | 1 | 1 | 0 | -1 | -1 | -1 | 1 | -1 |
| DY | 1 | 1 | 1 | 0 | -1 | 0 | -1 | 1 | -1 | 0 | -1 | 0 |
| GI | -1 | -1 | 6 | 10 | 13 | 16 | 16 | 13 | 12 | 11 | 13 | 13 |

GI' = (13) - (12) = 1 (Eq. 3)
AREA = (1) + 1/2*(12) - 1 = 6 (Eq. 1)

Table 5. Trace of algorithm execution for Figure 6 polygon.

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CODE | 3 | 1 | 8 | 10 | 3 | 1 | 1 | 10 | 7 | 6 | 9 | 11 | 2 | 1 | 9 | 7 | 5 | 6 |
| DX | 0 | 2 | -1 | 1 | 0 | 2 | 1 | 1 | -1 | -2 | 0 | 1 | 1 | 1 | 0 | -2 | -2 | -2 |
| DY | 3 | 1 | -2 | -2 | 3 | 1 | 1 | -2 | -1 | 0 | -1 | -1 | 2 | 1 | -3 | -1 | 1 | 0 |
| GI | 0 | 9 | 5 | 8 | 9 | 15 | 18 | 21 | 18 | 15 | 16 | 18 | 20 | 23 | 23 | 22 | 21 | 20 |
| $GB_{ODD}$ | 1 | 2 | 2 | 2 | 3 | 4 | 5 | 5 | 6 | 6 | 7 | 8 | 8 | 9 | 10 | 11 | 12 | 12 |
| $GB_{EVEN,UPPER}$ | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| $GB_{EVEN,LOWER}$ | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 3 | 3 | 3 | 3 | 3 | 4 |

GI' = (20) - (18) = 2 (Eq. 3)
AREA = 1/2*[(2) + 1/2*(12) + 2/3*(2) + 1/3*(4) - 1] = 29/6 (Eq. 2)

Kulpa, Z. 1977, Area and Perimeter Measurement of Blobs in Discrete Binary Pictures: Computer Graphics and Image Processing, Vol. 6, pp. 434-451

Peuquet, D.J. 1983, A Hybrid Structure for the Storage and Manipulation of Very Large Spatial Data Sets: Computer Vision, Graphics, and Image Processing, Vol. 24, pp. 14-27

Rosen, D. 1980, On the Areas and Boundaries of Quantized Objects: Computer Graphics and Image Processing, Vol. 13, pp. 94-98

Sankar, P.V. and Krishnamurthy, E.V. 1978, NOTE- On the Compactness of Subsets of Digital Pictures: Computer Graphics and Image Processing, Vol. 8, pp. 136-143

Scholten, D.K. and Wilson, S.G. 1983, Chain Coding with a Hexagonal Lattice: IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-5, No. 5, pp. 526-533

Mark R. Anstaett

BIOGRAPHICAL SKETCH

December, 1984

Mark R. Anstaett received his B.A. in Computer and Information Science from The Ohio State University, where he is currently working on a master's degree in the Department of Geography with specialization in computer cartographics. He is employed as a programmer/analyst with the Ohio Department of Natural Resources, working on the Ohio Capability Analysis Program GIS. In the past he has worked as a research assistant for Dr. Harold Moellering and for the Department of Geography on various software and hardware projects. He is a member of ACSM and his research work currently focuses on error analysis of vector/ raster data conversions.

Harold Moellering

BIOGRAPHICAL SKETCH

October, 1984

Harold Moellering received his Ph.D. from the University of Michigan and is currently Professor of Geography at Ohio State University. He is also Director of the Department's Numerical Cartography Laboratory. He is past Chairman of the ACSM Committee on Automation in Cartography and Surveying. He is Chairman of the National Committee for Digital Cartographic Data Standards in cooperation with ACSM, the U.S. Geological Survey and the U.S. National Bureau of Standards. He is a member of the U.S. National Committee for the ICA and travel chairman for the recent meetings in Australia. He has presented papers at the ICA Congresses in Maryland in 1978, Tokyo, 1980, Perth, 1984, and at the IGU Regional Congress in Rio de Janiero, in 1982. His research specialities include numerical, analytical and dynamic cartography. Professor Moellering is currently a member of the Committee on Cartography of the U.S. National Academy of Sciences/National Research Council.