Thus, AG_PAD is asymptotically eight times faster than PathPAD. The data are sufficiently uncertain that this factor could be as low as three, but it seems unlikely. In the experiments, AG_PAD was two to four times faster, with higher multiples for larger N, but these values are distorted by the slow access to two-dimensional arrays in the implementation of ALGOL 68 used for the experiments.

One important optimization to the algorithms is to pack the bit matrices in words so the bit operations can be done in parallel. This does not change the asymptotic behavior, but it does reduce the high order coefficients by a factor equal to the number of bits used in each word. For N less than the word size—an important special case—the access count expressions are:

PathPAD: $5.5N + 5.5R + 13.0$
AG_PAD: $3.8N + 4.7R + 33.5$

(A column access has been counted as N bit operations, but there is the same column access in each algorithm.) Since the uncertainty of R is not as crucial to these figures, it seems likely that AG_PAD is superior for values of N greater than about ten.

Acknowledgment. I appreciate the comments by C.L. Liu on an earlier version.

Received November 1977; revised January 1979

References
1. Gray, J.N., Lorie, R.A., and Putzolu, G.R. Granularity of locks in a shared data base. Proc. Int. Conf. on Very Large Data Bases, Framingham, Mass., Sept. 1975, pp. 428–451 (ACM, 1975).
2. Hansen, W.J. User engineering principles for interactive systems. Proc. AFIPS 1971 FJCC, Vol. 39, AFIPS Press, Montvale, N.J., pp. 523-532.
3. Hansen, W.J. A predecessor algorithm for ordered lists. *Inform. Processing Letters 7*, 3 (April 1978), 137–138.
4. Knuth, D.E. *The Art of Computer Programming, Vol. 1: Fundamental Algorithms.* Addison-Wesley, Reading, Mass., 1973.
5. Lockemann, P.C., and Knutsen, W.D. Recovery of disk contents after system failure. *Comm. ACM 11*, 8 (Aug. 1968), 542.
6. Tanenbaum, A.S. A tutorial on ALGOL 68. *Comptng. Surveys 8*, 2 (June 1976), 155–190.
7. van Wijngaarden, A., et al. The revised report on the algorithmic language ALGOL 68. *Acta Informatica 5* (1975), 1–236.
8. Verhofstad, J.S.M. Recovery techniques for database systems. *Comptng. Surveys 10*, 2 (June 1978), 167–196.
9. Williams, R. On the application of graph theory to computer data structures. In *Advanced Computer Graphics*, R.D. Parslow and R.E. Green, Eds., Plenum Press, London, England, 1971.

Scientific Applications
F.N. Fritsch Editor

Approximation of Polygonal Maps by Cellular Maps

George Nagy and Sharad G. Wagle
The University of Nebraska–Lincoln

The approximation of polygonal thematic maps by cellular maps, an important operation in geographical data processing, is analyzed. The data organization used for representing the polygonal maps is a widely used segment-based data structure, where class labels identify the regions bordering each segment on either side. The approximation algorithm presented operates on such an organization, eliminating the need for the recognition of region boundaries. Each segment is examined only once. The versatility of the new organization is further illustrated by the outline of algorithms for area computation and point inclusion. The algorithm is applied to a set of soil maps converted to computer-readable form by means of a coordinate digitizer.

Key Words and Phrases: polygon maps, cellularization, gridding, geographic data structures, computational geometry, computer cartography, automated cartography

CR Categories: 3.14, 4.34, 8.2

This project was sponsored by the Nebraska Natural Resources Commission and supported by the Old West Commission and the University of Nebraska Research Council's Maude Hammond Fling Fellowship Fund.

Authors' addresses: G. Nagy, Department of Computer Science, Ferguson Hall, University of Nebraska, Lincoln, NE 68588; S.G. Wagle, Department of Computer Science, Illinois Institute of Technology, IIT Center, Chicago, IL 60616.

1. Introduction

The demand for computer access to the vast amounts of geographical information which was hitherto available only in the form of printed maps has spurred considerable activity in the area becoming known as geographical data processing (GDP). An introduction to this field and a comparative evaluation of several major systems for processing *thematic* maps (which depict the subdivision of an area into homogeneous regions) are given by Tomlinson [20]. Additional program packages are reviewed in Aldred [1], Rhind [13], and Taylor [19].

An examination of the many systems in current use or under development reveals that the methods used to represent thematic information can be readily divided into *polygonal* schemes and *cellular* schemes [2, 5, 11, 13, 18].

The objective of this paper is to present a new algorithm for converting geographic data from a polygonal representation based on line segments to a cellular representation based on a rectangular grid. The next section provides an informal definition of the two types of data representation and a rationale for the importance of the interconversion problem. The relationship between the conversion algorithms and the underlying data structures is also explored briefly.

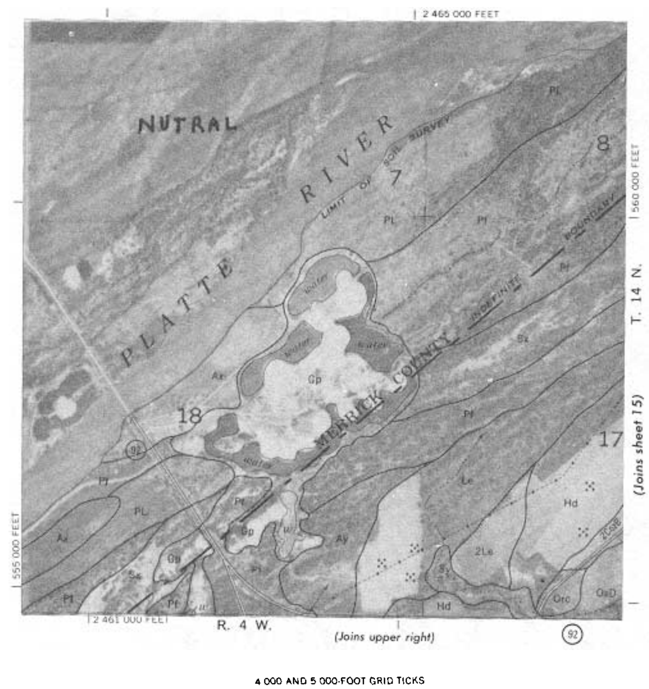
In the remainder of the paper certain properties of segment-based and cell-based data organizations for thematic maps are established and the conversion algorithm is derived. A related method of area computation is presented with a view of convincing the reader of the versatility of the proposed algorithm. The experience gained in implementing the algorithm is reported, and certain conclusions are drawn.

2. Alternative Methods of Data Organization

In *polygonal thematic maps*, the extent (i.e. the map area) is partitioned by *segments* into a set of *regions*, each of which is identified with a *class label*. The labels are drawn from a finite set of *class types*. The map is viewed as a planar graph consisting of nodes and segments. An example of a polygonal map for soil types is shown in Figure 1. Each segment is stored in the computer as a piecewise-linear curve consisting of a string of coordinates. The region boundaries will appear as curved lines because of the extreme shortness of the *links* between digitized neighboring points which make up the segments.

The difference between a *region-based* data structure for polygonal maps and the *segment-based* structure favored here stems from the manner in which the class labels are used. In the region-based organization the entire boundary and the class of each region are known [2, 4, 15]. In the segment-based organization the significant entities are the segments, for each of which the class types on either side are available [12, 6, 9, 10, 16, 17, 21].

Fig. 1. Soil map. This is an example of a polygonal thematic map. The classes or themes are the soil types denoted by the labels. This source map can be entered into a computer using a digitizer. For each segment in the map, the entry procedure involves tracing it and entering the soil type on its right and the soil type on its left (relative to the direction of tracing).



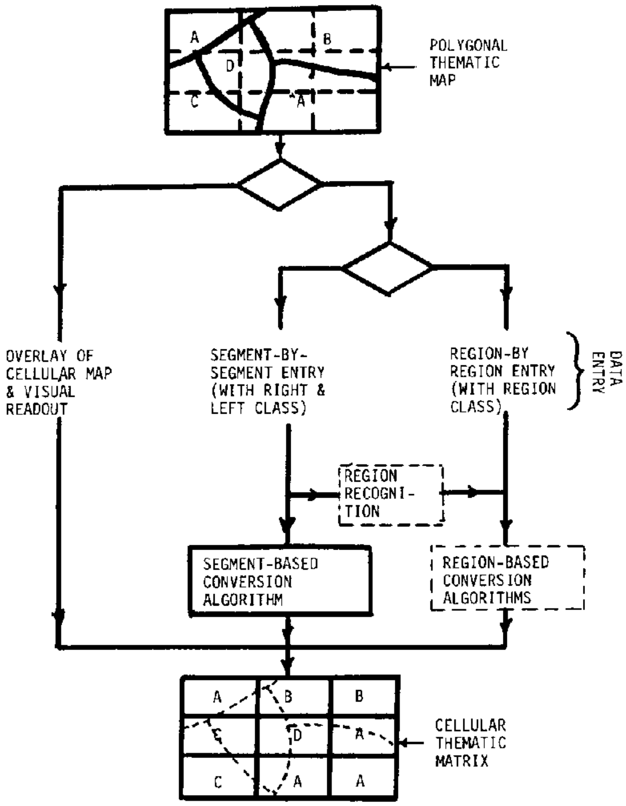
In *cellular maps*, the extent is divided into cells by means of a *uniform rectangular grid* (certain complications arise in large-area maps because of the sphericity of the earth, but we shall not concern ourselves with these here). A class label is then stored with each cell, which is regarded as an indivisible unit. Alternative methods of converting a polygonal thematic map into the corresponding cellular map are shown in Figure 2.

The leftmost path in the figure, representing the manual method in current use at, among others, the Nebraska Natural Resources Commission, consists of overlaying a transparent grid on the map and labeling each cell on the grid according to the class of the underlying region. The labels are subsequently keypunched and used to produce interpretive line-printer maps for various applications requiring different combinations of the labels.

The rightmost path requires tracing the boundary of each region by means of a coordinate digitizer or line-following optical scanner and then assigning labels to each cell depending on the class of the region which includes it [5, 15, 18]. Tracing every region boundary as a closed curve entails traversing each segment twice [3].

The center path, which represents the course advocated in this paper, allows tracing the segments constituting the map in arbitrary order, entering the labels of the regions on either side of each segment, and computing the cell assignments directly from this information. Alternatively (dotted box), the region boundaries may be

Fig. 2. Conversion of polygonal map into cellular map. The diagram illustrates the options for conversion. The new segment-based algorithm provides the most efficient conversion procedure, eliminating the duplicate tracing of segments inherent in region-based methods.



assembled from the individual segments and the cell assignment performed by means of a region-based algorithm as above.

The two principal types of geographical data representation, polygonal and cellular, have relative advantages and disadvantages with regard to various information retrieval applications. Each has its strong adherents, but our own interest in the matter was prompted mainly by the following input and output considerations. Entering the soil maps into a computer by means of a coordinate digitizer produces information in a form directly suitable for polygonal representation. On the other hand, producing maps by means of a line-printer or teletype terminal clearly requires the information in a cellular form, with the cell-size corresponding to the character spacing (which is generally different in the x and y directions). Incremental plotters operate much more efficiently with the information in polygonal form. Interconversion between the two methods of representation is thus a frequently needed operation and, when involving large databases, it requires efficient implementation [11, 18].

The conversion from cellular to polygonal form is fairly straightforward since it requires only the determination of boundaries between cells with different class labels. These boundaries then form the required polygonal representation, and the transformation entails no loss of information [3, 5].

We shall be concerned here only with the conversion of polygonal maps to cellular maps based on a prespecified grid. This conversion is an approximation inasmuch as a cell overlapped by several regions can be assigned only a single class. The algorithm to be developed requires as input only the coordinate strings corresponding to each segment, and the labels of classes on either side.

The segment-based organization is advantageous for data entry because each segment needs to be traced only once. If the conversion is, as suggested here, carried out directly on the segment structure, then the region-recognition phase (Figure 2) is also eliminated. Furthermore, the segment-based algorithm presented below requires only a single pass over the set of segments; any region-based algorithm will require a minimum of two passes if the region boundaries are processed independently.

It is possible to devise more efficient algorithms if it is assumed that the cell size is relatively small compared to the average size of a region and that consequently the vast majority of the cells falls completely inside region boundaries. For example, in SYMAP and other programs developed at the Harvard University Laboratory for Computer Graphics and Spatial Analysis, it is assumed that the correct labeling of the cells which straddle multiple region boundaries is not critical, and that a few misclassified cells among these can be tolerated [16]. In the derivation below we shall make no assumptions of this sort and will therefore obtain the correct classification of each cell (in the sense described in the next section) regardless of the number of regions it intersects.

3. Cell Assignment

A particular cell may lie completely inside a region. Since the information necessary to classify such a cell must be found elsewhere, we will defer consideration of such *interior* cells for the next section and consider for now only the *cut cells*. A *cut cell* is partially overlapped by two or more regions and has segments crossing some of its walls or has segments that lie completely within its area. Figure 3 shows a typical cut cell.

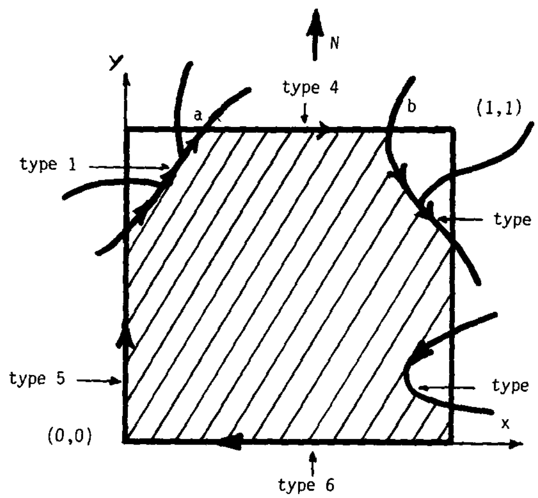
The obvious rule for assigning a class to a cut cell is to compute the fraction of the cell area covered by each class and then determine the class with the largest fraction. The necessary information is contained in an *overlap table* such as Table I. In this section we develop a technique for computing the rows of the overlap table corresponding to cut cells from the segment information.

The coverage $A(k, s)$ of cell s by class k is the cumulative sum of the areas $A'(j, s)$ over all regions j associated with class k:

$$A(k, s) = \sum_{j \ni \text{class}(j) = k} A'(j, s) \tag{1}$$

It is instructive to investigate the computation of the area of overlap $A'(j, s)$ between a region j of class k and

Fig. 3. Simple segments. Simple segments are a result of the intersection of segments with cell walls. Simple segments include the original segments in part or in full (type 1, 2, and 3) as well as parts of cell walls (type 4 and 5) and complete cell walls (type 6). The arrows in the diagram show the path of integration for computing the area of overlap between the class of the shaded region (not shown completely) and the cell.



cell s. Let us assume that region j does overlap cell s. The part of region j which overlaps cell s is bounded entirely by *simple segments.* A *simple segment* is of one of the following types:

(1) a complete segment terminating in junction points with other segments (i.e. a segment that lies completely within a cell);
(2) part of a segment terminating at a junction point at one end and at a cell wall at the other;
(3) part of a segment terminating on cell walls at both ends;
(4) part of a cell wall terminating on intersections with crossing segments at both ends;
(5) part of a cell wall extending from a corner of the cell to an intersection with a segment crossing it; and
(6) a cell wall extending from corner to corner.

The various types of simple segments are illustrated in Figure 3.

The overlap $A'(j, s)$ may be obtained by a clockwise integration around the entire boundary (the path of integration is also shown in Figure 3). For convenience, the integration is performed with respect to the x-axis. The southwest corner of the cell is chosen arbitrarily as the origin of the cartesian coordinate system, and the northeast corner is assigned, without loss of generality, the coordinates (1, 1). The area contributed by region j within cell s is:

$$A'(j, s) = \oint Y(x)dx = \sum_{i=1}^{n} \int_{x_i^-}^{x_i^+} Y_i(x)dx \qquad (2)$$

Table I. Overlap table for map in Figure 2. Each value in the table below represents the area of overlap between a class and a cell. For convenience the area of the cell is assumed to be unity. According to the "dominant" class rule, the class assigned to a cell, as shown on right, is the class with the largest fraction in the row for that cell.

Cell	Class A	B	C	D	Class assigned
1, 1	0.8	0.0	0.0	0.2	A
1, 2	0.15	0.65	0.0	0.2	B
1, 3	0.0	1.0	0.0	0.0	B
2, 1	0.15	0.0	0.45	0.4	C
2, 2	0.35	0.25	0.0	0.4	D
2, 3	0.6	0.4	0.0	0.0	A
3, 1	0.0	0.0	0.9	0.1	C
3, 2	0.7	0.0	0.15	0.15	A
3, 3	1.0	0.0	0.0	0.0	A

$Y(x)$ represents the entire boundary of the part of region j within the cell; n is the number of simple segments in this boundary; $Y_i(x)$ corresponds to the i'th simple segment; and x_i^- and x_i^+ correspond to the beginning and termination, respectively, of the i'th simple segment. Because the integration is performed around a closed curve, $x_n^+ = x_1^-$.

With the convention of clockwise integration, the region of interest is always to the right of each simple segment traversed from beginning to termination. If a segment is encountered in the opposite direction (terminus first), its contribution to the integral is negated. Consequently, simple segments which straddle a cell yield complementary contributions to the overlapping regions on either side of the segment.

Simple segments Y_i of type 4, 5, and 6 which are part of the west, south, and east cell walls contribute nothing to the area (since their integral is zero). Hence we need concern ourselves only with the contributions of such simple segments along the north wall.

Superficially, it might appear that in order to integrate on the north wall along a type 4 simple segment (i.e. one which intersects segments at both ends), one would have to have the x-coordinates of both segment-crossings simultaneously available. This would render it impossible to make the cell assignment on a strictly segment-based organization. Fortunately, such is not the case: The contribution of north-wall simple segments can in fact be determined by considering the segment intersections sequentially rather than at the same time.

Consider the segment $\overline{ab}$ in Figure 3. Its contribution to the integral is (b-a) modulo 1.0, which represents an increment corresponding to the x-coordinate of the intersection if the simple segment is to the left of the intersection, and an increment of one minus the x-coordinate if it is to the right. Furthermore, these increments are individually computable when the simple segment corresponding to the intersection, such as the one crossing the boundary at point "a" in Figure 3, is integrated. The increment can be added (using equation (1)) to the class overlap $A(k, s)$ of the cell along with the contribution of the intersecting segment itself. In conformity with the relation just cited, these additions are performed modulo 1.0 (the normalized cell area).

The type 5 and type 6 simple segments along the north wall are a special case of the type 4 segments discussed above—in this case, either $a = 0$ or $b = 1$ (or both), depending on the corner involved. The above rule with the modulo addition still holds, eliminating any need to discriminate between the three types. This too, if required, would have foiled our intention to process the segments sequentially (since we cannot determine to which type a cell boundary belongs by considering only its intersection with a single segment).

Summarizing the discussion so far, we see that the overlap $A(k, s)$ of a class k with a cell s is the sum of contributions of the type 1, 2, and 3 simple segments in the cell s which have class k either on their right or to their left. The contribution of a simple segment in turn involves integration along its path and, in the event the segment intersects the north wall of the cell, an increment depending *only* on the location of the intersection and the direction of the segment.

Since the contributions are additive, the *order* in which these are computed and added is *immaterial.* All that remains, therefore, is to obtain a sequence of these simple segments identified with the cells in which they lie. The following iterative structure shows how this can be done.

```
do i = 1 to # of segments in the polygonal map;
  do while (simple segment exists);
    get next simple segment for segment i and identify the
    corresponding cell s;
    compute contribution t of current simple segment to class k
    on its right;
    add this contribution to A(k, s) modulo 1.0;
    add complement of this contribution (i.e. 1.0 − t) to
    A(m, s), corresponding to class m on the left;
  end;
end;
```

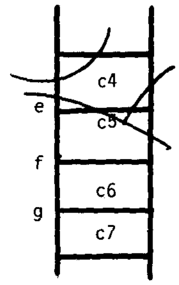
If we were to restrict ourselves to only the cut cells, we could complete the algorithm by following the above segment processing with a cell-by-cell scan over the overlap table. For each cell s the largest overlapping class is determined and the corresponding element of the thematic matrix is set to this class.

The sum of areas corresponding to the nondominant classes accumulated over all cut cells represents the degree of approximation involved in the conversion of the polygonal map to a cellular map. (No such loss will be entailed in the assignment of classes to the interior cells.)

4. Propagation

In the class-cell overlap computation method just described, only cut-cell class assignments are definitely identified. We will now show that the properties of the cut cells also characterize completely the class assignments of the interior cells.

Fig. 4. Columnar propagation. Cell $c4$ is a propagating cut cell. It propagates the class of its SW corner southward. Cell $c5$ is a nonpropagating cut cell and allows the propagation to pass through itself. The propagation results in the assignment of this class to the interior cells $c5$ and $c6$ and possibly beyond. The propagation class and the dominant class of a cell need not be same, cell $c4$ being such an example.



Consider any two points on a polygonal thematic map. They belong to the same region and hence to the same class if the straight line joining them is not intersected by a segment of the polygonal map. In other words, the class of a point can be propagated along a line to another point provided that line is intersection-free. Class assignment by propagation is a well-known systematic application of this simple observation [18].

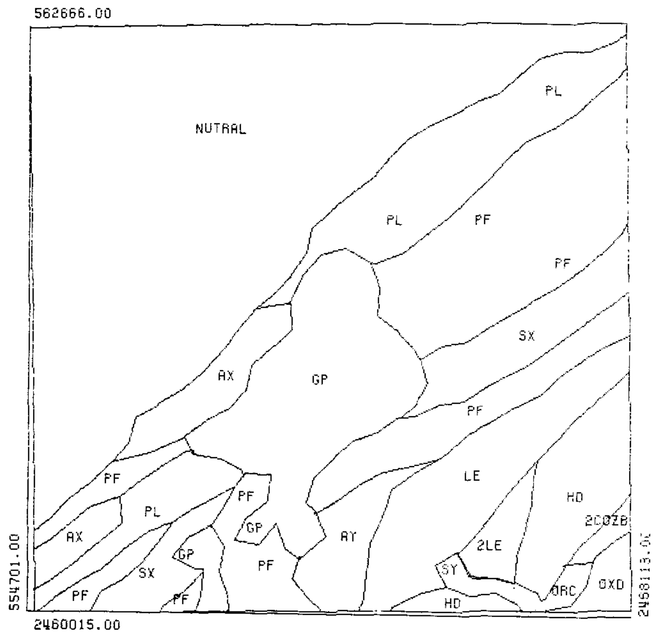
From a picture of a cell map superposed on a polygonal map, only a part of a column of cells is shown in Figure 4. Cells $c4$ and $c5$ in the figure are cut cells while the other two are interior cells. Let the class of point e, the SW corner of cell $c4$, be k. Since no segment crosses the west side of cell $c5$, the class of point e can be propagated south to point f, and further to point g. This in turn allows us to assign class k to cells $c6$ and $c7$.

Cut cells such as $c4$ are called *propagating* cells and have the property that their west walls are intersected by one or more segments. The distinction between propagating cut cells, nonpropagating cut cells, and interior cells is simple and can be determined during the segment processing outlined in the previous section. A cell is initially labeled an interior cell and changes to one of the other two types as one or more simple segments are identified with that cell.

What remains is the determination of the class of the SW corner of the propagation cell (called the *propagation class*). Generalizing observations from Figure 4, we see that this is the class to the *south* of a segment that crosses the west wall of the cell and additionally has the lowest y-coordinate ("minimum intersection coordinate") among all such intersections. Since the necessary information is contained in the simple segments, determination of propagation class can be performed during the segment processing in parallel with overlap computation. Therefore, the propagation class, as well as the minimum intersection coordinate of a cell s, is updated (if necessary) when a simple segment identified with the cell is recognized as terminating on its west wall.

To resolve possible ties between the current ordinate and the previous minimum ordinate, the *angle of incidence* on the west wall can be used. In this case, three

Fig. 5a. Incremental plotter output, used for verification, of segment database for the soil map shown in Figure 1. The digitizer operator enters the class labels by pointing to the label on the original map and then to a "software keyboard" or list of soil types. Hence the labels on the plotted map can be positioned according to their placement on the original.



variables—the current propagation class, the current minimum ordinate, and the current minimum angle of incidence—are required for each cell. The propagation type of a cell, as well as its propagation class, if any, are indeterminate until segment processing is completed.

The actual class propagation from propagating cells to the interior cells is performed following the segment processing phase in a single pass through the entire cell matrix, proceeding from north to south and column by column. It results in the assignment of classes to the interior cells. This iteration over the cells can be shared with the assignment of classes using the overlap table as described in the previous section.

5. Area Computation and the Inclusion Relation

The measurement of areas (planimetry) is an important and frequently used map operation in geographical data processing. What is typically required is the computation for each class of the sum of the areas of all the regions associated with that class (the *coverage* of each class). Another important GDP operation is the determination of the class of a given point (the *inclusion* relation between the point and its class). In this section we show how to perform these operations by treating them as special cases of the conversion problem.

The reduction of the conversion problem to that of area computation requires consideration of the entire polygonal map as a single-cell cellular map. Recalling the discussion on overlap computation, we see that only type 1 simple segments, which are the original segments themselves, can occur in this case. The elements of the conversion algorithm related to cutting the segments into simple segments, keeping track of propagation, and the final class assignment, are no longer needed. The modified algorithm for area computation is a simple sequential scan through all the segments and integration along each of these segments. Each segment yields complementary contributions to the area of classes on either side.

The idea of a cellular map with a single cell is also useful in establishing the inclusion relation between a given "search" point and its class. The cell in this case, however, has its SW corner at the search point and it has infinite extent otherwise. Since the propagation class of a cell is the class of the SW corner itself, it is sufficient to determine the propagation class of this cell. After discarding from the conversion algorithm all the superfluous components, a modified algorithm for point inclusion is obtained. This algorithm also requires a sequential scan through the segments during which the propagation class of the "cell" is updated, if necessary, whenever its west side is crossed by a segment.

6. Notes on Implementation

The conversion algorithm described above has been programmed in PL/1 and is now part of an ongoing soil mapping project [7, 8]. A cellular thematic map obtained using this program from the soil map shown in Figures 1 and 5a is reproduced in Figure 5b.

The application of the algorithm is preceded by sectioning the map segments, entered by means of a coordinate digitizer, into their component simple segments. This operation is performed by a segment preprocessing module which uses linear interpolation to insert additional points which serve as simple-segment end-markers at the intersection of the segments with the grid formed by the cellular array. The address of the cell to which each simple segment belongs is also added to the sequence of points constituting each simple segment.

The recognition of simple segments is essentially a lexical analysis problem. Consequently, the storage required is only that necessary for the simple segment with the largest number of points (the current version, however, processes an entire segment at a time). The design of this module was complicated only by the need to avoid overlooking special cases such as that of segments directly superimposed on a cell wall.

The conversion algorithm, however, requires random access to entries in the overlap and propagation tables. The number of rows in both tables is proportional to the total number of cells in the map. The number of columns in the propagation table is, as mentioned earlier, fixed, but the number of columns in the overlap table is proportional to the number of classes. As pointed out by one of the referees of a draft of this paper, the storage requirement is a potential bottleneck in implementing the algorithm.

Fig. 5b. Printed Output or Cell Map Corresponding to Figures 1 and 5a. The symbols for each class are assigned by the user; it is also possible to combine several classes under one label. The scale of the map, or cell size, is specified as an input parameter to the conversion program.

```
                  CELLULAR SOIL MAP FOR POLK COUNTY(20)      CELL WIDTH = 250 FT

REMARKS : AX = '$'  AY = '@'  GP = '%'  HD = '+'  LE = '*'  NUTRAL = ' '

          ORC = '<'  OXD = '>'  PF = '#'  PL = 'Z'  SX = '='  SY = '&'

          UNKNOWN = '?'  WRONG = 'X'  2COZB = ';'  2LE = 'V'


 27
 26                                  Z
 25                              ZZZZZ
 24                            ZZZZZZ#
 23                          ZZZZZZ###
 22                        ZZZZZZ#####
 21                      ZZZZZZZ######
 20                     ZZZZZZ#########
 19                     ZZZZZZ#########
 18                   ZZZZZZZ#########
 17                   ZZZZZ###########=
 16                 Z%%%###########===
 15                Z%%%%#########=====
 14               $$%%%%%######======##
 13              $$%%%%%%%###======###*
 12             $$%%%%%%%%======###***
 11           $$$$%%%%%%%%===####****+
 10          $$$$%%%%%%%%=#########****++
  9          $$$%%%%%%%%%#####*******++++
  8         ###ZZ%%%%%%%######******+++++
  7        ##ZZZZZ###%###@@*******V+++++
  6      ##$ZZZZ==##%%#@@@*******VV+++++
  5     #$$$ZZ===#%%%%%%@@*****VVV++++;
  4     $$ZZ##=%%####%@@@*****VVV+++;;>
  3    $$ZZ#====%####@@@@****&&V&++<<>>
  2    ZZ##====#%%####@@@**+++++++&?<>>>
  1

           .    |    .    |    .    |
               10        20        30
```

```
SORTED LISTING OF PRINT SYBOLS


 1      NUTRAL
 2      ORC        <
 3      HD         +
 4      SY         &
 5      AX         $
 6      LE         *
 7      2COZB      ;
 8      GP         %
 9      OXD        >
10      WRONG      ?
11      UNKNOW     ?
12      PF         #
13      AY         @
14      SX         =
15      2LE        V
16      PL         Z
```

In the current implementation, two strategies are used to keep the storage requirements within manageable bounds. First, we limit the number of classes which can overlap a single cut cell to 5. A warning is issued if this limitation is exceeded, but to date the alarm has not been sounded. Second, we fold the map into a matrix of submaps, and process each submap separately.

A more sophisticated approach would take advantage of the fact that only a small fraction of the cells are cut cells which require the storage of multiple-class information. Sparse matrix techniques could then be used to accommodate this information. Another alternative suggested by the referee mentioned above is to presort the segments by geometric locality in such a manner that only a portion of the overlap table need be accessed for each localized group of segments. This approach is conceptually similar to the "sweep" method used in the Harvard POLYVRT program [12].

The task of determining the optimum size of the submaps is avoided in our application by limiting the width of the maps to be printed to the standard 132-character line-printer specification. The storage required for a page-sized map can be readily accommodated in one partition of the IBM 360/65 memory. The method of computation is, however, completely general, and the program can tailor the size of the submaps to whatever amount of storage is made available to it.

A potentially more serious problem is that the input map is not necessarily "ideal" in the sense that segments would never intersect and that the ends of segments which meet would always coincide perfectly. The program takes care of such situations by examining the immediate vicinity of segment ends and merging them into a single node. This node-merging process is, it must be admitted, a miniaturized and simplified version of the region-recognition process which we claimed earlier to have eliminated.

7. Summary and Extensions

An important geographic data processing problem, that of approximating polygonal thematic maps by cellular maps, was formulated and analyzed. A solution utilizing a segment-based data organization was presented. It is claimed that the resulting algorithm is computationally more efficient than an algorithm which requires partitioning the map area into closed regions. Simple modifications of the conversion algorithm to accomplish other common operations on polygonal maps, such as area determination and point inclusion, were also demonstrated.

Cell maps are special instances of polygonal maps. Therefore a more general formulation of our algorithm would allow the approximation of a given polygonal thematic map by another polygonal map. The modified algorithm would yield the optimal class assignment for the second map through segment-by-segment integration, propagation, and a scan through the overlap table.

A related problem is the determination of the degree of similarity between two polygonal thematic maps. The total area where the class labels on the two maps do not correspond is a good measure of the deviation between the two maps. Therefore a generalized algorithm yielding an overlap table would also allow calculation of the area of mismatch between such maps. The application of the mismatch calculation to quality control is of direct interest to us in soil-map data-entry.

Shamos [14] and others in the field of computational geometry have studied the computational complexity of inclusion and other GDP operations. Those studies, however, are for the case in which the map segments are straight lines and are not made up of multiple, connected, straight-line links as in our case. The extension of that work to maps with curvilinear (or piecewise linear) segments is worth exploring.

Acknowledgments. We are grateful for a number of useful suggestions made by the referees of the ACM and by R. Burton, who read an earlier draft of this paper. We also acknowledge several helpful discussions with S. Seth during the development of the algorithm, and the help of S. Tam in programming.

Received March 1978; revised January 1979

References

1. Aldred, B.K., Ed. Proc. IBM UK Scientific Ctr. Seminar on Geographical Data Processing. Report UKSC 0073, IBM (UK) Scientific Ctr., Peterlee, 1975.
2. Baxter, R.S. Some methodological issues in computer drawn maps. *The Cartographic J.* (Dec. 1976), 145–155.
3. Brassel, K., Hanson, P., Jarvis, R., and Marble, D.F. The handling of space and time dependent data. Report R-77/1, Geographic Inform. Systs. Lab., State U. of New York at Buffalo, 1977, p. 28.
4. Burton, W. Representation of many-sided polygons and polygonal lines for rapid processing. *Comm. ACM 20,* 3 (March 1977), 166–177.
5. Calkins, H.W., and Tomlinson, R.F. Geographic information systems, methods, and equipment for land use planning. IGU Comm. on Geographical Data Sensing and Processing, Ottawa, Ont., Aug. 1977, pp. 273–278 (see ORMIS, Oak Ridge Regional Modelling Information System).
6. Loomis, R.G. Boundary networks. *Comm. ACM 8,* 1 (Jan. 1965), 44–48.
7. Nagy, G., Tam, S., and Wagle, S.G. SMAPS digitization guide. Dept. of Comptr. Sci. Rep., U. of Nebraska, Lincoln, Oct. 1978.
8. Nagy, G., Tam, S., and Wagle, S.G. SMAPS soil map processing. Dept. of Comptr. Sci. Rep., U. of Nebraska, Lincoln, July 1978.
9. Nordbeck, S., and Rystedt, B. *Computer Cartography.* Studentlitteratur, Lund, Sweden, 1972.
10. Peucker, T.K., and Chrisman, N. Cartographic data structures. *The Amer. Cartographer 2,* 1 (April 1975), 55–69.
11. Peuquet, D. Raster handling in geographic information systems. Rep. R-77/7, Geographic Inform. Syst. Lab., State U. of New York at Buffalo, 1977.
12. POLYVRT Manual. Harvard U., Lab. for Computer Graphics and Spatial Analysis, Cambridge, Mass., 1974.
13. Rhind, D.W. Computer aided cartography. *Trans. Inst. of British Geographers, New Series 2,* 1 (1977), 71–96.
14. Shamos, M.I. Geometric complexity. Proc. 7th Ann. Symp. on Theory of Comptng., ACM, New York, 1975, pp. 224–233.
15. Russell, R.M., Sharpneck, D.A., and Amidon, E.L. WRIS: A resource information system for wildlife management. Res. Paper PSW-107, USDA Forest Service, 1975.
16. Schmidt, A.H., and Zafft, W.A. Programs of the Harvard University Laboratory for Computer Graphics and Spatial Analysis. In *Display and Analysis of Spatial Data,* J.C. Davis and M.J. McCullagh, Eds., John Wiley and Sons, Inc., New York, 1975.
17. Silver, J. The GBF/DIME system: Development, design and use. U.S. Bureau of the Census, Washington, D.C., 1975.
18. Sinton, D. Spatial data manipulation and analysis techniques. In *Geographical Data Handling,* Tomlinson, Ed., IGU, Ottawa, Ont., 1972.
19. Taylor, D.R.F., Ed. Proc. of Symp. on Geographic Inform. Processing, Dept. of Geography, Carleton U., 1976.
20. Tomlinson, R.F., Ed. *Computer Handling of Geographic Data.* UNESCO Press, Paris, 1976.
21. Waugh, T.C. GIMMS—An example of a user oriented, integrated system with special reference to locational description and mapping capabilities. Proc. Fourth European Symp. on Urban Data Management, London PTRC, 1974.