AUTOMATED NAMES PLACEMENT IN A NON-INTERACTIVE ENVIRONMENT

Lee R. Ebinger and Ann M. Goulette
Geography Division
U.S. Bureau of the Census
Washington, DC 20233

ABSTRACT

Accurate and aesthetic placement of text is an important component of a well-designed map. Census Bureau cartographers and computer programmers have worked together to incorporate cartographic names placement conventions into mapping software. The result is an effective names placement system that is capable of positioning text and resolving text placement conflict without the need for human intervention. The names placement software is part of the automated mapping system developed to provide paper maps for numerous 1990 Decennial Census activities. This paper examines the current automated names placement system and describes algorithms for labeling point, line and area features. Several approaches for handling text conflict and text storage are also discussed.

INTRODUCTION

Poorly placed names are often the identifying factor of a computer-generated map. Even with sophisticated algorithms, computers have yet to produce maps with text placement that matches manual placement by trained cartographers. Computer-generated maps have been noted for overlapping text, upside-down text and text placed at awkward angles -- all properties that cartographers strive to avoid. As a result, some computer-generated maps have used predetermined coordinates, interactive techniques or a manually produced type stickup as an overlay to the computer-produced base map for placement of labels. These solutions lead to maps that are more accurately called "computer-assisted" rather than "computer-generated." Despite great improvements in other areas of computer cartography, automated names placement remains a problem.

The rules for manual label placement can be found in any introductory cartography textbook (e. g., Robinson, et al., 1978) and have been documented by Imhof (1975). Imhof's fundamental rules are that names: 1) should be legible; 2) should be easily associated with the features they describe; 3) should not overlap other map contents; 4) should be placed so as to show the extent of the object; 5) should reflect the hierarchy of objects by the use of different font styles; and 6) should not be densely clustered nor evenly dispersed. Imhof asserts that these rules are often violated because satisfying one rule may break another.

Research in automated names placement has acknowledged text placement conventions and the difficulty involved in fulfilling the established rules. Monmonier (1982) states that the goal of automated names placement is to optimize the number of placements that follow cartographic conventions and guidelines. Zoraster (1986) allows for deletion of labels that would overlap if placed. Other researchers (Ahn & Freeman, 1983; Cromley, 1983; and Pfefferkorn, et al., 1985) attempt to place names that have the smallest degree of freedom first and those that are less constrained in position afterward. All the aforementioned research use a recursive process that repositions as many labels as necessary to find appropriate locations for all text and encourages human intervention as the final step to improve label positions.

THE PROBLEM

Recursive computer processing and human intervention are not feasible solutions for the automated names placement problem at the U.S. Bureau of the Census due to the large volume of maps produced. For the collection of data for the 1990 Decennial Census, the Census Bureau will produce through automated means over ten different map types with nationwide coverage. This effort will produce an estimated one million different map sheets on monochromatic electrostatic plotters. All of the maps are highly dependent on accurate names placement and use guidelines compiled from accepted cartographic conventions and census traditions for the placement of labels. All maps must be plotted under rigid deadlines using limited computer and staff resources. Given these constraints, inefficient use of computing time and dependence on human map inspection for improved names placement are not possible.

There are many effective ways to improve label positioning without resorting to manual intervention or recursive processing. For instance, the map scale can be increased or larger scale insets can be made in areas of feature density. Text size can be decreased. Text may be repositioned within or along the feature or object. A text placement priority by feature type may be established to minimize names conflict. Text may be angled, hyphenated, stacked or interletter spaced. Arrows may be used in difficult cases to associate text with its object in a congested area. Fishhook symbols (see Block 124 in Figure 1) may eliminate the placement of duplicate names of adjacent areas with the same identifier. Text may be replaced by key numbers and a key listing. As a last resort, text may be suppressed altogether.

The Census Bureau employs some of these techniques in its completely automated names placement algorithms. Maps are preprocessed to determine adequate scale and to identify rectangular windows for insets of larger scale. However, most map types have practical constraints on the maximum number of sheets and sheet sizes. On some map types, text size may be decreased as long as legibility is preserved. All maps utilize a predetermined order to place feature names in terms of their importance to the map. The software provides for multiple alternative placement positions, as well as for stacking text, and in some cases, angling text to improve placement. The use of arrows and fishhooks, elements of Census Bureau mapping convention, has been incorporated into the names placement routines. Because naming features is crucial to census maps, suppression of labels is used only as a final alternative. Interletter spacing, hyphenating words and key numbering are currently not a part of the software.

The automated names placement algorithms used to produce the maps required for 1990 Decennial Census data collection represent a real-world application of a non-interactive, non-recursive system. The algorithms place labels for points, lines and areas. Overlapping text is avoided, although some overlap is allowed through the use of "see-through" screened fonts. Alternative placements are attempted before more radical procedures (arrows, fishhooks and suppression) are implemented. Text is not always placed as a manual cartographer would position it; however, the algorithms have resulted in readable, effective maps for 1990 Decennial Census operations.

THE ALGORITHMS

An integrated cartographic text placement system must incorporate algorithms for positioning names of point, linear and areal features. The problem of names placement differs significantly for the three types of data. The Census Bureau approach to point names placement is rather simple; the approach to linear and areal names placement is more sophisticated. In addition to these algorithms, the system must provide a method for the detection of text overlap. All of the names placement algorithms use an overlap detection routine for determining the final position of labels. Figure 1 shows examples of many features of the names placement algorithms.

Point Names Placement

The point names placement algorithm is used to identify point features depicted by pictorial symbols. The point symbol is centered on a single coordinate and the name is offset from the symbol. The algorithm begins by testing the position of the symbol for overlap with other text and symbols. If conflict is detected, neither the symbol nor the label is plotted. Otherwise, one of four ranked positions for the label is tested for overlap until a non-overlapping location is found. These positions are: 1) above and to the right of the symbol; 2) below and to the right; 3) above and to the left; and 4) below and to the left. If the name cannot be placed at any of the four positions, only the point symbol is placed. The name is always placed parallel to the horizontal axis.

Linear Names Placement

The linear names placement algorithm is used to label linear features such as roads or streams. Most often, linear features are labeled with a name or code, but on some map types, street address ranges are plotted also. Address ranges present a more difficult problem: they must be placed on the correct side of the street and reflect the relative direction in which the range increases along the street segment. In implementing the algorithm, the following guidelines are used: 1) text is placed right-reading and follows the curvature of the feature; 2) text is not allowed to overprint the linear feature; 3) text is not permitted to overlap previously placed text and, in some instances, point symbols; and 4) text placement maintains a true ground-to-map positional and directional orientation for address ranges. As in the point-oriented algorithm, when text conflict is detected, alternative positions are examined. If no suitable placement is found, the algorithm provides a user-defined option to force the text to plot despite overlap or to suppress the label.

The first procedure of the algorithm checks whether the feature consists of one or more segments. Different methods are used for single and multiple segment features. For single segment features, the segment length and text length are compared. If the segment length is not sufficient, multiple word names may be stacked and placed above and below the segment. Alternatively, the name may be placed on one line above the segment. Address ranges may be stacked or centered along the segment. As a last resort to placement, text is allowed to extend beyond the end of the line segment.

If the feature consists of more than one segment, the algorithm first attempts to place the text on the longest segment, provided the segment length exceeds the text string length. If the length constraint is met but overlap with other text is encountered, the name is moved incrementally along the segment until a non-overlapping position that meets the length requirement is found. If moving the text is unsuccessful, the procedure is repeated for the next longest segment. When the procedure has exhausted all possible single segment placements, multiple segments are tried. In these cases, the algorithm selects the two longest consecutive segments and

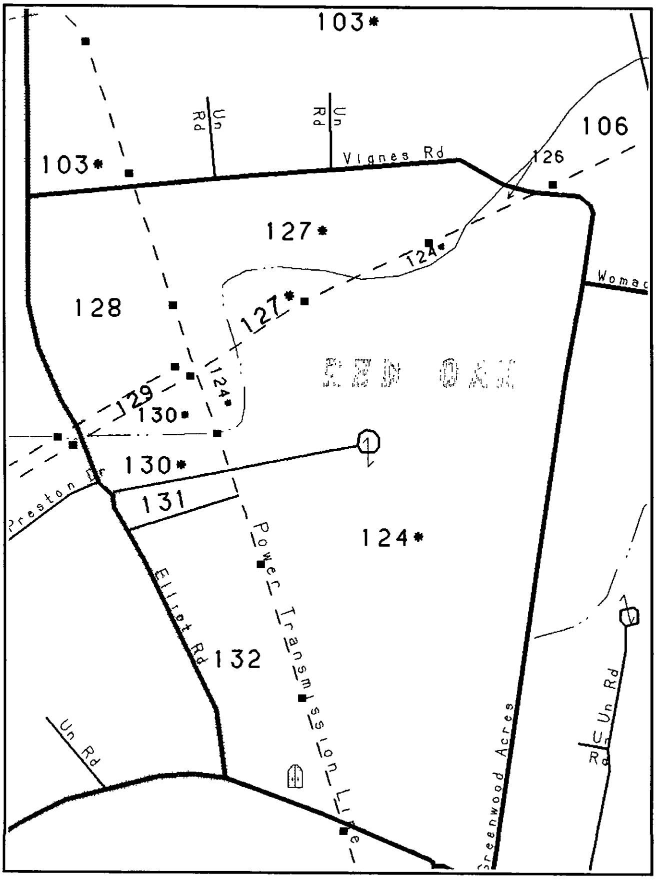

Figure 1. Portion of a data collection map with examples of linear and areal names placement using the non-interactive placement algorithms.

tries to place the name if the length of the combined segments exceeds the text length. Text is broken into substrings for each segment. The substrings are placed at angles corresponding to the angles of the segments. If the name cannot be placed, the process is repeated using the three longest consecutive segments and so on. If all segments have failed the procedure, the name is placed along and beyond the feature using all segments of the feature. Address ranges are restricted to a single segment of the multiple segment feature. The address range may be stacked or centered along the line segment.

After the number of segments is determined, a position for the text is calculated and checked for overlap each time text placement is attempted. If the position is unsuitable, another position is selected and tested. The placement, orientation, and test for suitability of placement are performed in three steps: finding the starting node of the selected line segment(s), calculating the lower-left corner coordinate of the text rectangle and detecting text overlap.

First, it is necessary to locate the starting node of the portion of the feature where the name is to be placed, given the constraint that text must be right-reading. The segment(s) used for placement has both a starting and ending node. The leftmost of these nodes is selected as the starting point for text placement. For vertical lines, the topmost node is the starting point. Although successful for most cases, this method may occasionally fail for sinuous line segment chains.

The lower-left corner of the label is determined next. The text is always offset diagonally from the starting point (above or below the line segment and toward the end of the segment) to avoid obscuring information. For vertical lines, the offset is always positive.

Finally, the coordinates of the text rectangle are calculated and the rectangle is checked for overlap with previously placed text and point symbols. If the text is a substring, it is also checked against other members of the text string. If no text overlap is detected, the text rectangle position is stored and the text string is plotted.

The aesthetic placement and legibility of linear feature names are affected by the map scale. On small-scale maps, the curvature of linear features is more extreme than on large-scale maps. As a result, text placement on large-scale maps is more cohesive. Although aesthetic placement could be improved by smoothing the coordinates of the underlying feature, this is not implemented in the current algorithm. Small-scale maps also have higher feature density. This causes small-scale maps to have a higher ratio of unlabeled-to-labeled features because of increased text conflict. In the current algorithm, conflict between text and most features is not detected. Conflict detection is performed only against previously placed text and point symbols. Therefore, text is easily obscured by a dense feature network.

Areal Names Placement

The areal names placement algorithm is used to label polygons such as water bodies and political or statistical areas. The algorithm finds a location within a polygon where a name can be placed parallel to the horizontal axis. Should text conflict occur, the algorithm provides several options for manipulating the text so that other alternatives are available.

The areal names placement algorithm performs two major functions. The first function heuristically selects points within a polygon to be considered as a center for the text. The second function determines the suitability of the point locations and provides placement options should the original point location be unsatisfactory for text placement.

Providing multiple potential coordinates for text placement within a polygon allows greater flexibility for the final placement of text. The optimal position is one where the text fits entirely within the polygon, is free of text conflict, and is near the center of the polygon. Using a scan-line technique, potential text locations are selected using two criteria: 1) the centrality of the coordinate within the polygon, and 2) the length of a horizontal scan-line segment that intersects the edges of the polygon. Centrality describes the proximity of a point to the center y-coordinate of the polygon. The length of the scan-line segment refers to the space available to place a name horizontally within the polygon. The length is calculated as the difference between the minimum and maximum x-coordinates of the line segment. A single scan-line can yield multiple line segments if the subject polygon has internal polygons (islands) which break the scan-line into smaller, discontiguous segments. The number of scan-line segments calculated for each polygon is specified as a parameter in the algorithm.

The areal names placement algorithm orders scan-line segments based upon a criterion of centrality, segment length, or a combination of segment length and centrality, as specified for the algorithm. First, the subject polygon is dissected by the specified number of scan-lines at equal intervals to obtain the horizontal line segments. The line segments are then sorted by the chosen criterion. By selecting centrality as the criterion, the longest horizontal line segments with a unique y-value are collected and reordered by their proximity to the center y-coordinate of the polygon. The line segment ordering is from middle to top and bottom using alternating scan-line positions. The sort by length option orders the line segments from the longest to the shortest segments. The combined length-centrality sort first performs a length sort on the line segments, then the longest line segments are ordered as to their centrality. Figure 2 illustrates the result of the scan-line method.

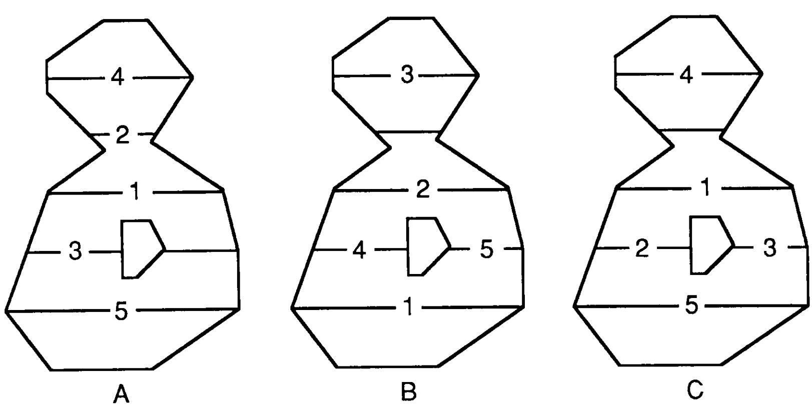

Figure 2. Different results of the scan-line text placement algorithm based on sorts by (a) centrality, (b) segment length, and (c) a combination of segment length and centrality. The numbers represent the priority of the scan-lines for text placement.

The areal names placement algorithm centers the text along a scan-line segment and tests for text conflict. An optional test is performed if the name must fit entirely within the polygon. When either test detects a problem with the text placement, the algorithm uses other options to manipulate the text.

Text for areal features may be plotted on one, two or three lines. Words on multiple lines are arranged to minimize the difference in the number of characters per line. All lines of text are centered around the midpoint of the scan-line segment. In addition, each line of text is checked for potential text conflict. If no conflict-free position is found, the algorithm provides the option to force the name to plot or to omit the name.

For some types of areal features, such as census blocks, the identifier must be located completely within the polygon. Two checks are performed upon the text position: a point-in-polygon check and a line-intersection check. Prior to the checks, a rectangle is constructed around the text to be used to approximate the text position. In the point-in-polygon test, the corner coordinates of the text rectangle are tested to determine whether they lie within the interior of the polygon. In the line-intersection test, the four line segments forming the text rectangle are checked to ensure that they do not intersect with the edges of the polygon. If either of the checks detects an overlap condition, an alternative to this text position is found.

Given that polygons appear in a variety of shapes, placement of text at a specific location may not be acceptable. The areal names placement algorithm includes several options for resolving the problem of text placement. For example, the text position may be rotated so that it is parallel to the longest line segment of the perimeter of the polygon. The text position may be offset from the original coordinate. The size of the text may be reduced. A different scan-line segment midpoint may be used. An arrow may be used to associate text placed outside the polygon with the center of the polygon. Unlabeled polygons may be fishhooked to adjacent polygons of the same name.

Depending on the map type, these options have different priorities. Generally, it is sufficient to select a different scan-line segment midpoint or reduce the text size in order to find an acceptable position for the text. The other options listed are applied if a more rigorous search for a solution to the text placement problem is required and if the options are relevant to the map type.

The areal names placement algorithm performs well if the shape of the polygon is not too irregular and the text rectangle is small compared to the area of the polygon. Text is placed on a straight line; the algorithm does not attempt to curve the name to conform to the shape of the polygon. For some areal features, such as small lakes or towns, the text size frequently exceeds the size of the polygon. These names are not arrowed to the polygon, since offsetting the name outside the polygon consumes space elsewhere on the map. In this case, the name is simply centered in the polygon and allowed to extend beyond its edges.

The use of screened fonts has proved advantageous for identifying some types of areal features. These fonts allow text to overprint other features and labels without obscuring information.

Detection of Text Overlap and Text Storage

The ability to detect text conflict and store text positions is essential for producing a legible map. Two methods have been developed that use similar procedures for detecting text overlap, but differ in the way text positions are stored. One method, a vector method, stores positional information in the form of coordinate values at map scale. The other method, a grid-based procedure, converts the positional data into row and column values. Both methods produce similar results in terms of the number of text conflicts detected and the amount of processing time required.

In order to efficiently perform text conflict detection and text storage, a text rectangle is used to approximate the position and extent of the text string. In addition, a second rectangle is constructed from the maximum and minimum x-y coordinates of the text rectangle. This second rectangle, the text envelope, is used to streamline text conflict detection.

The vector method calculates coordinate values at map scale for the text rectangle and envelope. These coordinate values and the angle of the text rectangle are stored in the memory of the computer as real numbers.

The grid method superimposes a grid over the map. The location of a grid cell is defined by row and column positions. This method stores the positional information of the envelope surrounding the text rectangle by determining two parameters: the column and row numbers of the grid cell that correspond to the upper-left corner of the text envelope and the number of rows and columns of the grid cells occupied by the text envelope. Grid cell locations of the text rectangle are determined and stored also. If the text rectangle coincides with the centroid of a grid cell, that grid cell becomes off-limits to subsequent text placement. Row and column information is packed into bit positions and stored in memory.

In both methods, the detection of text conflict consists of two checks. The first check tests whether the current text envelope overlaps any previously stored text envelopes. When text envelopes overlap, a more refined check is performed using the data stored for the text rectangles. If no overlap of text rectangles is detected, the positional data for the current text envelope and rectangle is stored.

The methods differ, however, in implementation of the checks. The vector method compares the maximum and minimum text envelope values. If the current envelope overlaps a stored envelope, a line intersection test determines whether the text rectangles overlap. The grid method tests the text envelopes by comparing the beginning and ending grid cells of the current label to the stored text envelope grid values. If a more rigorous check is required, the grid cells occupied by the text rectangles are tested for matching row and column values.

For linear names placement, an additional check for text conflict is performed. If a text string is broken into substrings in order to follow the curvature of a multiple-segment line, each substring must be checked for text overlap against other members of the text string. If any substring causes overlap, the entire string must be repositioned. The vector method processes substrings consecutively and stores the positional information before proceeding to the next substring. The linear names placement algorithm keeps track of the number of text substrings stored. If text overlap occurs while processing a text substring, the linear names placement algorithm deletes the information for each substring from the storage array and begins again. The grid method uses a different technique. When consecutive characters are closely spaced along a curved line, the proximity and angularity of the text rectangles cause frequent overlap of the same grid cells. Instead of using grid cells to represent the position of the text string, the characters of the text string are inscribed within circles. The distances between circle centroids are measured and compared to a specified distance that allows a minor degree of overlap. This internal test on the text string is performed after testing the substring against stored text using the grid cell method.

The application of either storage method has advantages and disadvantages. Factors involved in evaluating the relative performance of the two methods include the amount of memory necessary for the application and the effectiveness of the names placement.

Restricting the size of arrays is important for keeping the overall size of the mapping programs below the maximum limit permitted by Census Bureau hardware. The vector method uses twenty-six 36-bit words of memory to store the positional information for a text rectangle. The raster method generally consumes less space by packing integer values into bit positions. The number of words stored depends upon the angle, length and height of the text rectangle. Because the vector method uses more space, it can store fewer text positions. Therefore, large-scale maps containing less text can use the vector storage method; small-scale maps with greater text placement requirements must rely on the grid method for storing text positions.

The grid method has two major drawbacks. First, text overlap cannot be detected where portions of the text rectangle fall within a grid cell but do not fall on the centroid. For these cases, the line intersection test of the vector method is superior for detecting text overlap. Second, there is a limit imposed upon the grid dimensions and size of the text rectangle by the number of bits in the word used to store grid cell values. These limits restrict the map image area size, text length and text height. The current image size limitation is approximately 41.0" in both dimensions; either text length or height is limited to 10.2". The vector storage method sets no practical limits upon map size and text dimensions.

Currently, the grid method has an important advantage over the vector method. Normally, text is not permitted to extend beyond the map borders or appear within inset areas. The grid method allows the map image area dimensions and rectangular inset windows to be defined precisely. When text is checked for overlap, the text position can be compared against these limits, confining text placement to the map image area. This check is not implemented in the vector procedure, but it may be included in future software.

The most effective overlap detection algorithm would assimilate the best aspects of both methods. It would be capable of accurate text conflict detection and efficient text storage. In addition, it would use a minimum amount of computer processing time and allow as much text placement as possible in the most acceptable positions. Revisions to the software will attempt to optimize these features.

SUMMARY

The automated names placement algorithms described here represent the efforts of Census Bureau cartographers and computer programmers over several years of names placement guidelines generation, software development and refinement, and map production. As much as possible, these efforts have attempted to replicate, given hardware and time constraints, conventional placement of names on maps. Through an ongoing critical review process, the software has evolved into its present state. The result has been the effective placement of names for point, linear and areal data without resort to interactive repositioning or recursive processing. Although some label placements are awkward, the majority of text is positioned using established cartographic rules.

The Census Bureau intends to further refine its automated names placement algorithms to improve the appearance of text and to extend the algorithms to maps produced for publication purposes. For instance, improvements to the names placement algorithm may include the ability to erase and reposition text, repeat the label along the feature, break the text string by syllables, and smooth the underlying feature. The capability to detect text overlap against all linear and areal features may be added. In addition, the ability to confine text to the image area limits may be added to the vector method for text overlap detection. For publication maps distributed to the public, more aesthetic enhancements, such as key numbers, may be included. Automated names placement on these maps may be aided by interactive editing.

REFERENCES

Ahn, J. & Freeman, H. 1983. A program for automated name placement. Proceedings, Auto-Carto VI, pp. 444-453.

Cromley, R.G. 1983. An LP relaxation procedure for annotating point features using interactive graphics. Proceedings, Auto-Carto VI, pp. 127-132.

Imhof, E. 1975. Positioning names on maps. The American Cartographer, Vol 2, No. 2, pp. 128-144.

Monmonier, M.S. 1982. Computer-Assisted Cartography Principles and Prospects. Prentice-Hall, Inc., Englewood Cliffs, NJ.

Pfefferkorn, C.; Burr, D.; Harrison, D.; Heckman, B.; Oresky, C.; & Rothermel, J. 1985. ACES: A cartographic expert system. Proceedings, Auto-Carto VII, pp. 399-407.

Robinson, A.; Sale, R.; & Morrison, J. 1978. Elements of Cartography. John Wiley & Sons, New York.

Zoraster, S. 1986. Integer programming applied to the map label placement problem. Cartographica, Vol. 23, No. 3, pp. 16-27.