AUTOMATING LINEAR TEXT PLACEMENT WITHIN DENSE FEATURE NETWORKS

David H. Alexander
Carl S. Hantman
Geography Division
U.S. Bureau of the Census
Washington, D.C. 20233

ABSTRACT

In preparation for the 2000 census, the Census Bureau plans to provide maps with address information to local governments for review and update. The algorithms that have been used by the Census Bureau in the past have not been flexible enough to place this kind of text readably on small-scale maps. This paper describes a new non-interactive algorithm which has been developed at the Census Bureau. This algorithm simultaneously considers all the address range text which must be plotted in a certain area of the map and can make incremental adjustments to the location of each text item in the set in order to position it for maximum clarity before any positions are fixed. The flexibility that this approach provides has already made it possible to produce maps that display address ranges, census block numbers, and linear features and their names at a scale significantly smaller than comparable maps produced for the 1990 Census.

INTRODUCTION

During the 1980's, cartographers and computer programmers at the U.S. Census Bureau developed a fully automated mapping system to produce the thousands of maps that support census field operations and data products. The software developed for the automatic placement of feature names on maps formed an integral part of this system. The text placement algorithms were generally successful in meeting the requirements of the maps for the 1990 decennial census.

Currently, the Census Bureau is developing a Master Address File for use in the 2000 decennial census and other surveys and censuses. In conjunction with this effort, the Census Bureau is updating its TIGER data base to keep it as current and accurate as possible. As part of the TIGER Improvement Program (TIP), the Census Bureau will provide local governments with maps displaying streets and address ranges within their jurisdictions. Participating local officials will update the maps with new and revised feature and address information. These changes will then be incorporated into TIGER to improve the linkage of the Master Address File to the TIGER data base.

THE PROBLEM

The addition of address range text to maps that must also display linear and areal feature names, poses cartographic challenges for automated text placement algorithms. In addition, the large number of unique map sheets that will be generated for this program precludes interactive intervention for text placement refinement. Fully automated placement algorithms must be used and must result in effective maps.

These maps must allow local officials to readily determine what information exists in TIGER and to enter updates. To make the maps as manageable as possible, the number of map sheets must be held to a minimum by using the smallest practical map scales. The requirement to display high volumes of text items within relatively small areas of the map makes demands that the existing Census Bureau text placement algorithms could not meet.

Over the last three decades, as computers have been applied more and more to the job of making maps, one of the last areas of cartographic skill to be addressed has been the automated placement of feature labels. Even in recent years, with the advent of numerous GIS and mapping packages, the full automation of text positioning still lags behind other components of the mapping process.

Researchers working in this field have pointed out the difficulty of the task (Ahn & Freeman, 1987). Names must be easily read, clearly associated with the feature to which they belong, and avoid overlapping other text (Imhof, 1975). The complexity of rules and algorithms necessary to achieve fully automated names placement have caused many otherwise automated systems to ultimately rely on interactive methods for final text placement.

Much of the research on automated names placement has taken place either in academic institutions or research environments. This has included the work of Ahn and Freeman (1987), Basoglu (1984), Zoraster(1986), and Marks and Shieber (1991). Their work has primarily emphasized the problems of point and areal label placement.

The linear address ranges, however, provide the greatest challenge for the Tiger Improvement Program (TIP) maps. Address information is particularly difficult to place because, unlike most linear feature text, the address data cannot be positioned along a wide range of locations following the feature. The address number must be placed in close proximity to the point where the address break (the last potential address number at an end of a segment) occurs and on the correct side of the street. Since these maps will serve as a window into TIGER, the positioning of the address numbers must effectively convey the address range for each side of each street segment stored in the data base.

The production environment in which automated mapping research is conducted at the Census Bureau provides its own set of constraints. The automated map production system consists of many integrated software components that have been continually refined and supplemented to meet changing cartographic requirements. Limited staff resources and tight time constraints do not allow large scale rewriting of the system. New mapping needs must be met by software that can be readily and effectively incorporated into the existing mapping system within very tight schedules.

Automatic name placement algorithms used at the Census Bureau have followed a non-recursive, sequential approach. Categories of text having the greatest importance for the map would be placed first. Each subsequent name processed is positioned at a location that is not already occupied by earlier placed text. Backtracking capability has not been provided. Once a piece of text had been placed, it could not be repositioned. If a free area, which maintained the visual association of name with feature, could not be found for the new text, the name would either be suppressed or allowed to overlap previously placed text. On large scale maps this was generally not a problem. Smaller scale maps, however, had a higher ratio of unlabeled to labeled features, because of increased text conflict in areas of high feature density (Ebinger & Goulette, 1989). For the TIP maps, a more flexible approach was required.

THE SOLUTION

Unit of Analysis

The first and most important decision to be made in the development of this new algorithm was to choose the algorithm's fundamental unit of analysis. Three possibilities were considered: the segment, referred to as a 1-cell in TIGER, the point or 0-cell, or the polygon or 2-cell.

Using the 1-cell as the unit of analysis meant placing addresses on both ends and both sides of a section of a linear feature in one step. An advantage of this is that the TIGER system retrieves address information one 1-cell at a time. This minimizes the need to store many addresses at once and the need to retrieve address information more than once per 1-cell. The chief disadvantage is that the biggest challenge in placing addresses is avoiding collisions between addresses which are neighbors at a street intersection and thus belong to different 1-cells. The 1-cell method would place one of these addresses without considering the placement of the other.

Using the 0-cell as the unit of analysis meant placing all the addresses around an intersection of linear features at one time. This solved the problem of placing addresses which are neighbors at a street intersection. It lost the ability, however, to deal with the next most important challenge in placement, coordinating the placement of addresses on the same side of the same 1-cell. This is especially important when considering short 1-cells.

The third method, which proved to be the best, is to consider an areal unit to be the fundamental unit of analysis. If each address range end applies to one side of one section of a linear feature, then each can be associated unambiguously with the area of which that section of the linear feature forms one side. All that needs to be done then is to restrict each number to being placed strictly within that area and it becomes possible to consider each such area as the unit of analysis. The only possible collisions that any number need guard against will be with the other numbers associated with that area. If a data structure could be devised that would contain everything about an area and the addresses that needed to be arranged within it, then all the information needed to place these addresses, avoiding any overlaps among them, would be present.

Address Suppression

The exact definition of these areas remained to be established. The smallest possible unit was the 2-cell, which is the fundamental unit of area in the TIGER system. The data and the software would be simplest if the 2-cell was adopted as the fundamental unit of analysis, getting all the data for each 2-cell, placing the addresses, and then moving on to the next 2-cell. The limitation of this approach is that it does not allow 2-cells to be merged to consider larger areas as a unit. In some situations, merging 2-cells can provide advantages for enhanced text placement. Certain classes of 1-cells are not plotted on certain map types and others that are plotted are irrelevant for addressing. When a linear feature is broken by such a feature, there is an opportunity to simplify the addressing for those 2-cells by merging 2-cells together and by suppressing addresses at the intersection of certain 1-cells (see Fig. 1).

In order to be able to merge 2-cells together, several 2-cells must be gathered at one time and searched for 1-cells which can be eliminated. The next highest level of geography that the TIGER system provides is the census block. This is a convenient unit to use as the fundamental unit of analysis because it gives some opportunity for merging 2-cells, but is not so large that the merging process would be too time-consuming and also because the TIGER Improvement Program (TIP) maps display geography which is composed of whole blocks.

The first restriction on this suppression is that the neighboring 1-cells must have the same feature identifier (street name). It is critical for anyone using these maps to study the pattern of addressing to see changes in street names and so addresses will always be printed at these points.

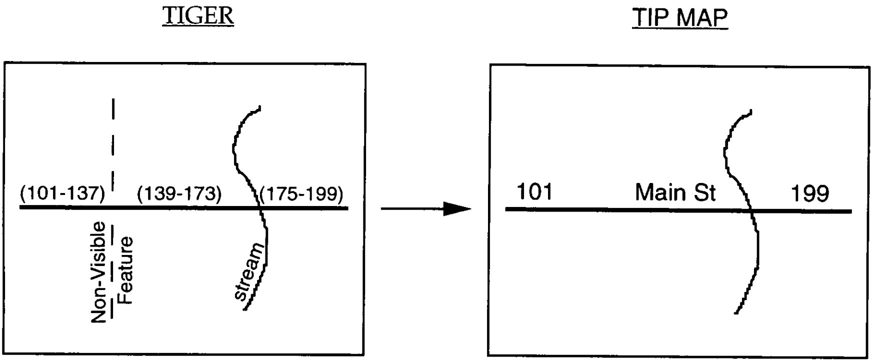

Fig. 1 Address suppression across 2-cells

The second restriction is that there is no addressable feature intersecting at this point on the side under consideration. If a non-addressable linear feature, such as a stream or a railroad track, intersects on the same side as the addresses under consideration, this is not important to the addressing scheme and the address break there may be suppressed. If any linear feature intersects on the opposite side as the addresses under consideration, this is also not a part of the addressing scheme on the other side of the street and it may also be suppressed. A street intersection on the same side, however, will always force the address break to be shown.

Another restriction is that the merged address range must not be misleading. If there is a change in parity (between odd and even address numbers), a change in direction of increase, or a gap of more than two between the adjacent 1-cells, the map user is likely to be misled or confused by looking at the merged range. To let the user see these important shifts in addressing schemes and to not see addresses that are not there, any of these conditions at an address break will force the address break to be shown.

Because the area that the algorithm works on during each pass is a single block, it is impractical to consider suppressing pairs of addresses separated by a block boundary. The final restriction on merging 2-cells and suppressing addresses, therefore, is that addresses are not suppressed across block boundaries.

The restrictions on suppression are therefore:

1) the feature identifier (street name) is unchanged
2) there is no addressable feature intersecting on the side under consideration
3) the address pattern must be consistent between 1-cells with no address gap
4) there is no suppression across block boundaries.

Address Ranges and TIGER

The discussion up to now has assumed that addresses are always simple ranges, with a low number belonging to one end and a high number at the other end. The TIGER system has the capacity to, and often does, store more complicated address information than this. This is because TIGER can store multiple address ranges associated with each 1-cell. This presents another problem. It is already difficult to present simple ranges on a map and yet for the TIGER Improvement Program that these maps support, these variant ranges may be very important. The compromise between displaying all address data as stored in TIGER and simplifying the display to provide map clarity and practical scales, was to merge multiple overlapping ranges into one range for display purposes. Multiple ranges with gaps in the normal two unit number ascending or descending sequence, and those which reverse direction or change parity would not be merged. Instead, a partial range, based on that portion of address information which is contiguous and reflects the same direction and parity, is printed on the map. A special symbol ('+') is appended to each number to show that there is an anomalous address range situation in TIGER that is too complex to display without creating excessive map clutter (see Fig. 2).

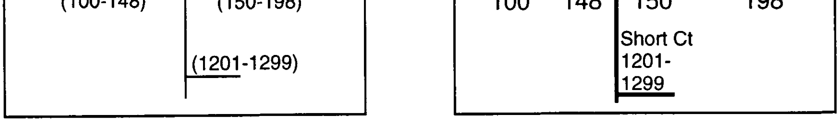

Fig. 2 Display of variant address ranges

Placement

Once all the information about linear features and addresses for a particular sub-area of the map is in place, the algorithm must begin to make decisions about where to place each number. This must be done so that there is a direct, unambiguous visual association between each number and the street side-end that it labels and so that no number overlaps a linear feature or another number (Ahn and Freeman, 1987).

The placement rules that create this visual association are:

1) Place each number as close to the correct end of the street as possible
2) Place each number along a subsegment at least as long as the number
3) Place each number parallel to a subsegment
4) Place each number a standard offset distance from the subsegment
5) Place pairs of numbers adjacent at an intersection in a balanced way
6) When necessary, stack pairs of addresses in a consistent and intuitive manner.

The heart of the algorithm starts with a pair of numbers which are neighbors at a street intersection. Each of the streets is searched starting at the intersection and moving away, searching for a subsegment which is large enough to hold each number. When these are found, a position is calculated for each number along the subsegment so that when the numbers are plotted parallel to and offset from their respective subsegments, the upper left and upper right corners of the boxes that enclose the numbers meet on the line that bisects the angle of intersection.

There are many reasons why the algorithm described above will fail to work for all cases. One is that a street may have no subsegment long enough to hold an entire number. The present implementation of this algorithm will suppress that number. It will be necessary, before TIP mapping begins, to provide a better solution in which the address is placed on the longest subsegment available close to the intersection if its neighbors curve away from it, or to search for a chord of the polygon of sufficient length if the neighboring subsegments curve back toward it.

Another problem may arise if the first subsegments of sufficient length do not allow for placements that do not collide. Currently the algorithm calculates the positions along the infinite lines corresponding to the subsegments where the numbers would go in order to avoid collision. If either of these positions is off the subsegment, the subsegment pair fails and the algorithm will try another pair. The algorithm is set up to try every possible combination of subsegments given the two linear features. If no pair of subsegments will serve, then the present implementation will suppress and the algorithm needs to be improved as described above.

Another problem occurs when the location for a number would position it across the midpoint toward the other endpoint of the 1-cell. Such a placement is likely to result in confusion as to which end of the street an address is associated with, especially if it meets the address from the other end of the street coming its way. The solution for this situation is to stack the addresses in a consistent manner. When stacking, the number which belongs to the left as the addresses are read is placed on top and has a hyphen appended to it (see Fig. 2). The address for the right is placed on the bottom. If the flagging character '+' is required for these addresses, it is appended to the bottom address only.

In many instances there is only one address to be placed at an intersection and this is simpler than the general case described above. In this case, it is still necessary to look for a subsegment long enough to hold the number, but the inside upper corner of the box is offset a small distance from the nearest subsegment, instead of looking for the bisecting line.

Overlap Detection

The operations described above have used geometrical calculations on pairs of rectangles to place address ranges so that they do not overlap. This has the advantage of allowing very precise placements which waste a minimum of space, but cannot be used to detect any collision within a large group of objects because of the excessive time that it would consume. The algorithm also uses a grid-based procedure like the one described by Ebinger & Goulette (1990) for two purposes. One use is as a back-up to ensure that overlaps that the algorithm did not foresee do not happen. After the text has been placed for a block by the procedure described above, the grid is checked to see if any of the text overlaps already placed text. If not, the grid is updated with the new text and the text information is written to a Map Image Metafile (MIM), a text file containing basic graphics commands, from which the mapping program reads it when the complete map is put together. If there is a collision, further adjustment of the text positions will be necessary.

The other use of the grid-based text overlap detection method is to pass information about text positions between different modules of the mapping system. Because of the special challenges of placing address range text, the TIP mapping process has assigned address range placement the highest priority of the map text types and so this algorithm has a clean slate to begin with. Subsequent text placement modules, e.g. block numbering, must have an efficient way to check for overlaps with address range text as they operate. The block numbering module does this by reading the address range MIM, which has already been created, and loading the positions into the grid-based system. The address range algorithm does not do this itself, but it will have to be included whenever a map is created on which address ranges do not have the highest priority of the text types.

FUTURE DEVELOPMENT

One of the most important areas for improvement for the future is to make the algorithm flexible enough to consider the placement of other types of text along with the address ranges. Up to now, the method has assumed that the placement of address ranges has been unquestionably the highest priority. Address ranges could be placed without consideration for other text and then the other text would be placed later to avoid the address ranges. In the application that this has been developed for, in which the placement of address ranges is clearly the most challenging part of text placement, this is satisfactory. In another application, however, it might be important to be able to develop rules more flexibly, with less clear-cut priorities among text types. In this case, it would be necessary to consider other text along with the address range text.

The situation that it would be easiest to adapt to would be if the other text were associated with the same small areas on the map. The notion that each address range number belongs to a definite polygon on the map enables the algorithm to work with blocks and polygons as its fundamental units. If some text type, a block number for example, is also associated with similar areas, the fundamental flow of the algorithm need not be changed and some additional overlap checks and placement steps can be added for each area.

A more difficult case will be to coordinate with the placement of text which adheres to linear features or to larger areas, linear feature names, for example. In such a case, it won't be generally possible to know which of the areas a piece of the linear feature name text will be placed in. All the areas adjacent to the feature will have to be considered first. With linear text labeling features in every direction, it will be necessary to look at every area before deciding on address range and linear feature name placement for any of them.

One possibility would be to attempt to place the linear feature name text in each of the candidate areas, quantify the quality of the resulting placement and then go back and place each where it fit the best. This means that it will not be possible to break text items among segments of linear features that cross areas. Allowing the ability to break up the text and coordinate its placement with the address ranges will add a great deal of flexibility to the placement of linear text, but will also add considerable complexity to the algorithm.

CONCLUSION

The algorithm for the placement of address ranges described in this paper will provide greater success in the display of TIGER data on Census Bureau maps. By increasing the flexibility of the techniques employed for the fully automated positioning of feature labels, these methods have already achieved a more complete display of map information, without resorting to larger scales or increased numbers of sheets. The utility of the maps as a window into TIGER is thus improved and the task of the map user is facilitated.

As the principles involved in the address range algorithms are extended to additional categories of linear, areal, and point labels, even greater success can be expected in producing a well integrated and balanced display of cartographic data.

REFERENCES

Ahn, J., and H. Freeman. 1987. "On the Problem of Placing Names in a Geographic Map." International Journal of Pattern Recognition and Artificial Intelligence, vol. 1, no. 1, pp. 121-140.

Basoglu, U. 1982. "A New Approach to Automated Name Placement." Proceedings Auto Carto V, pp.103-112.

Imhof, E. 1975. "Positioning Names on Maps.", The American Cartographer, vol.2, no.2, pp. 128-144.

Ebinger, L., and A. Goulette. 1990. "Noninteractive Automated Names Placement for the 1990 Decennial Census.", Cartography and Geographic Information Systems, vol. 17, no. 1, pp.69-78.

Marks, J., and Shieber, Stuart. 1991. "The Computational Complexity of Cartographic Label Placement." Harvard University, Center for Research in Computing Technology, Cambridge, Massachusetts.

Zoraster, S. 1986. "Integer Programming Applied to the Map Label Placement Problem." Cartographica, vol. 23, no. 3, pp. 16-27.