Software Tools for Processing U.S. Census Bureau TIGER Line Files

John Blodgett, Urban Information Center University of Missouri-St. Louis
John McIntyre, SAS Institute Inc.

ABSTRACT

The U.S. Census Bureau is generating block level detail computerized maps as a part of the 1990 Census of Population and Housing. The availability of such computerized files will greatly enhance graphical display of demographic data in government and business applications. Prototypes of these maps are available in line segment form. SAS® macros are presented that convert these line segments into geographic polygons for mapping. Files are converted in a mainframe session, and resultant SAS files are downloaded using the micro-to-host link, for polygon chaining and SAS/GRAPH® processing.

INTRODUCTION

This paper describes the results of a software development project at the University of Missouri-St. Louis. Because of the limits on time and space, this paper assumes that the audience is already familiar with the basic concepts of TIGER line files. Users needing additional background information should check one or more of the references listed at the end of this paper.

While the ultimate goal of the project was to develop some specific low-level GIS applications using the SAS System, the emphasis here is on how the files were converted to SAS format and on the specific application of generating polygon files suitable for use with the SAS/GRAPH GMAP procedure.

PHASE 1: CONVERTING TIGER LINE FILES TO SAS FORMAT

Before attempting to develop any applications software utilizing the TIGER line files, our first objective was to restructure the files as distributed by the Census Bureau so that they would be more easily and efficiently accessed using the SAS System. While this process involved using INPUT statements to transform a series of sequential files into equivalent SAS data sets, the conversion was considerably more than just a format transformation. In creating the SAS data library (SDL) version of the TIGER file, we were also interested in enhancing the structure of the data so that implicit connections between segments (records/observations) were made explicit. We were very interested in being able to directly access any relevant data in the subordinate segments (record types 2 through 6, referred to here as TIGER2, TIGER3, and so on) while processing the primary TIGER record (TIGER1). We also felt it was important to restore the concept of a Nodes file to the TIGER system. In its native form as a complex database on the Census Bureau's computers, the TIGER data structure is made up of the fundamental entities of points, lines, and polygons (called 0-cells, 1-cells, and 2-cells in the technical descriptions.) In converting the data into a format for exporting in sequential form, some of this structure was lost, or at least made less explicit. In converting to a SDL, and then in chaining to form polygons, it was our intention to restore much of this very useful original structure.

Figure 1 is a schematic diagram of the linkages on the original TIGER line files and on the TIGER SDL. It shows that the original files are linked only by sequential keys. TIGER1 (record type 1) can be sequentially merged with each of the subordinate records 2 through 4 and 6. To link a TIGER1 record with TIGER5 (alternate or *alias* feature identifiers) requires an extremely cumbersome *double merge* in which you must merge with one or more TIGER4 index records that contain repeated keys, which then have to be looked up in the TIGER5 file. This is okay for a transport format but obviously much too cumbersome to be left as the final applications format.

The bottom portion of Figure 1 depicts the structure of the TIGER SDL. It helps illustrate the following key features of our new data structure.

1. pointer variables, (that is, variables whose values contain the observation numbers of other SAS data sets so that they can be used to access those data sets using the SET POINT= statement) have been defined. In the TIGER1 data set, we have such pointers defined for all the subordinate data sets. There are multiple pointers to the TIGER7 (nodes) and TIGER5 (feature id fields) data sets.

2. a new SAS data set, TIGER7, has been created where there was no equivalent line file. Each observation in TIGER7 represents one node or one x-y coordinate combination occurring on either the from or to node of a TIGER1 record. The pointers in TIGER1 are named FNODE and TNODE. What's more, each TIGER7 observation contains an array of 7 (more or less—the exact number is controlled by a convert macro parameter) printer variables back to TIGER1. This circular pointer-chaining permits direct access chaining of TIGER1 segments.

3. the sacrifice of the ability to have unlimited references to TIGER5 for the sake of simplicity. We allow only two references to TIGER5 from TIGER1 for the purposes of specifying an alternate or alias name for a feature; these two pointer variables are named ALIAS1 and ALIAS2. More importantly, we have added a pointer (R5PTR) from TIGER1 to TIGER5 corresponding to the feature id fields recorded in TIGER1. Applications programs needing to extract feature id information from TIGER1 records can opt to drop the lengthy fields themselves and just keep the R5PTR variable which enables them to call back the full set of identifiers in a later step or program. We make use of this feature in the chaining summary report steps of our polygon chaining application.

Conversion Details

The conversion of the TIGER line files to TIGER SDL format takes place in the mainframe environment using a pair of related macros, %TIGERLN and %TIGRLNK. Our test runs were done in the MVS/XA environment using Release 5.18 of the SAS System. As might be expected when processing files of the considerable magnitude of the TIGER series (5 megabytes per county is typical), these runs use considerable system resources. In order to get all the linkages put in place, there is a considerable amount of sorting and merging that takes place. We have not nor do we plan in the near future attempting to run this conversion in a PC environment.

These macros have a number of optional features that allow the user to customize the conversion. Some of the more important ones have to do with which geocode variables the user wants to keep (do you really want both census and FIPS place codes, or do you

need to keep Indian reservations codes?), whether you want to convert your latitude-longitude coordinates to miles, and whether you want your linear feature id fields in all upper- or mixed case. The author has prepared a more detailed document summarizing the conversion, including record lengths and byte counts. As a general rule (depending on which conversion options are chosen), converting from the sequential format to SDL does not make a significant difference in the total storage space utilized. The enhancements are in ease of use and the obvious efficiencies involved in not having to execute expensive INPUT conversions.

MOVING TO THE PC ENVIRONMENT

Having converted the TIGER files to an easier to work with format, we were now ready to work on our first important application: chaining polygons for the GMAP procedure. We had the advantage of having already written a similar program for chaining off the old DIME files. However, we wrote that program in the 70's in FORTRAN and hadn't used it (or seen it) in years. What we remembered about it, however, was that the algorithm to do the chaining involved setting up a lot of arrays to keep all the data that had to be shifted around during the chaining. We had once made an attempt to program this with the SAS System and had been frustrated by the inability to use (easily, at least) multidimensional arrays, and then with the limits on the size of these arrays related to the limits on the length of the Program Data Vector. We noted that Version 6 of the SAS System had a new feature that would get around this problem for us: temporary arrays. This is when we decided that we would implement our chaining application in the Version 6 environment. For us, that meant Release 6.03 running PC DOS on an IBM PS2 Model 60.

Our biggest problem was getting the micro-mainframe link setup to work. That took a few months (off and on). Once we had it working we discovered that we could download the TIGER SDL for Boone County, Missouri, (about 5 megabytes) in about 30-35 minutes.

The TIGER Polygon Chaining Application

The application to generate GMAP procedure data sets from the TIGER SDL was initially handled by breaking the process into a series of steps and creating a SAS program module for each step. A series of DRIVER programs was used to assign parameter values and invoke the working modules. It took only about two weeks to get this up and running once the downloading problem had been solved.

Here is a brief overview of the steps involved in going from the TIGER SDL to a GMAP dataset (and then to a sample map):

1. The application is driven by a series of symbolic parms that specify options, such as what geocodes you want to chain (tracts, places, blocks, and so on), whether you need to qualify the geocodes with county codes, a criteria for when shape points are to be added to the PROC GMAP data set, and so on.

2. The first step (GENLINKS) accesses TIGER1 and picks off segments that are on the boundary of the specified geocode type. For example, it could check to see if TRACTL needs TRACTR as a criterion for selecting the segments which could yield the boundary links necessary to chain around census tracts. Since the segment is part of the chain for each of these two polygons (TRACTL and TRACTR), two observations are generated on the LINKS data set.

3. The CHNLINKS program sorts the LINKS data set and then does the work of creating the chained-links data set. It also was written to create an unchained-links data set, but in most cases that part of the code is unnecessary.

4. An optional report step can produce a detailed chaining report showing which features were chained along and the names of intersecting features.

5. A final step converts the chained links data set into a PROC GMAP data set. This is the first step in the process where any x-y coordinate variables are used. Prior to this, all chaining has been done using the FNODE/TNODE pointer variables. Shape points from TIGER2 can also be added to this step.

6. A test map can be generated using a small %DOMAP macro.

THE TIGRPOL APPLICATION (SAS/AF® SOFTWARE)

In order to make our software more accessible and easier for users to comprehend, we decided to rework our polygon-generator using SAS/AF software. The result is the TIGRGPOL.PROGRAM application. It does the same thing as our DRIVER modules did: specifications. This was our first nontrivial experience using SAS/AF software. Although we experienced the usual problems associated with being the first in our shop to try the newest technology, we feel that the extra effort was worth it.

We have created a distribution diskette of the software described here, and having the TIGRGPOL module should make running the application a lot easier for users in other shops. Ultimately, we would also like to build SAS/AF applications for the conversion phase and for some sample mapping applications.

APPLICATIONS

While the focus of this paper has been on our work in building some tools, we would like to just briefly describe some examples of how those tools can be (and have been) applied.

The most obvious application is to use the PROC GMAP data sets created to display data. Missouri was the site of the 1988 Dress Rehearsal Census, which has resulted in some 1988 census data that we can use in conjunction with our TIGER products (just as the whole century will be able to do with data from the 1990 Census starting in 1991). The first data product from the 1988 Census was the PL94-171 (Public Law or "redistricting" file). It contains limited detail but does have data on race down to the census block level showing the distribution of the black population in the city of St. Louis. We did this by applying software described in this paper, which converts to the SAS System and generates PROC GMAP data sets. We chained around the over 5000 blocks in the city, which required a little over one hour on our PS2/60. While we could have downloaded our PL94 data and created our maps on the PC, it was more convenient for us to UPLOAD our BLOCKS polygon data set to MVS and do the mapping applications there.

The first thing we did after moving our block boundary files to MVS was to run a series of GREMOVE procedure steps, which generated the full hierarchy of census geography for the city. Once you have block group or census tract boundaries, you can easily use PROC GREMOVE to get them. In St. Louis, we have also defined clusters of tracts as tract groups which have recognizable neighborhood names like Soulard and Sherman Park. We also used PROC GREMOVE to create tract group polygons. Using some utility macros developed several years ago, we were able to easily generate ANNOTATE data sets that could be used to draw the outlines of each of these higher-level geographies and to display their codes

and/ or names centered at their geographic centroids. These tools were critical in helping us create an annotated city map that not only displayed the racial data but was also easy to comprehend because of the neighborhood outlines and labels. In addition to the block-level map we were able to create several maps at the census tract level depicting trend data between 1980 and 1988. This task was greatly simplified since 1980. This could not be said about block groups or blocks. A sample map generated with PROC GMAP using the TRACTS data set is shown in Figure 2.

A second application involved creating an intersections data set to be added to our TIGER SDL. This data set represents each intersection of two named features in the SDL. It was generated using the TIGER7 (nodes) data set to generate all feature-name combinations associated with each node. This was made very easy because of the array of TIGER1 pointers in the TIGER7 observations together with the R5PTR variables in the TIGER1 records. In fact, this application was coded and debugged in less than two hours and consists of fewer than 50 lines of SAS code. The resulting data set was printed using the PRINT procedure and serves as the reference needed for the next application.

One of the most common applications needed for people dealing with demographic information in an urban environment is to summarize that data for circular areas about specified locations. These locations are typically street intersections (or can be derived by specifying a street intersection and offset distances). A SAS program that accepted a pair of street names and a radius value (in miles) was written. The program (called CIRCAREA) did a sequential search of the INTSEX data set to match the pair of street names and to retrieve the X-Y coordinates of the location. This provided the center point coordinates necessary to create a circular area aggregation report using the PL94 file as the input data set. This file, like most files that will be released for the 1990 census, contains the X-Y geographic centroid coordinates for each of the geographic areas summarized. These coordinates are in the same units (latitude-longitude) as those on the TIGER files which makes it an easy matter to calculate distances from TIGER file locations to data file center points. Thus, a program to select all blocks or block groups within an *n*-mile radius of a specified point is easy to code. The hardest part is determining the coordinates of the specified point, and this is taken care of with the INTSEX data set. For most applications, you can easily do a manual lookup of these "ground zero" coordinates.

SUMMARY AND CONCLUSIONS

An important first step is being able to utilize the power of the TIGER line files using the SAS System is to convert the files into a data structure that allows for efficient and, whenever possible, direct access to the geographic entities—lines, points and polygons—that are the fundamental elements of a geographic base file. The software developed by the Urban Information Center to convert the files to a pointer-linked SAS data library and to generate the polygon entities is an approach to creating such a structure. Applications utilizing the TIGER data can be developed much more rapidly and with much less use of computing resources by starting with the TIGER SDL and the associated polygon data sets.

This paper has concentrated on describing the buildings of an infrastructure upon which a library of application modules can be built. It is our hope and intention to share this software with others who are interested in building such applications.

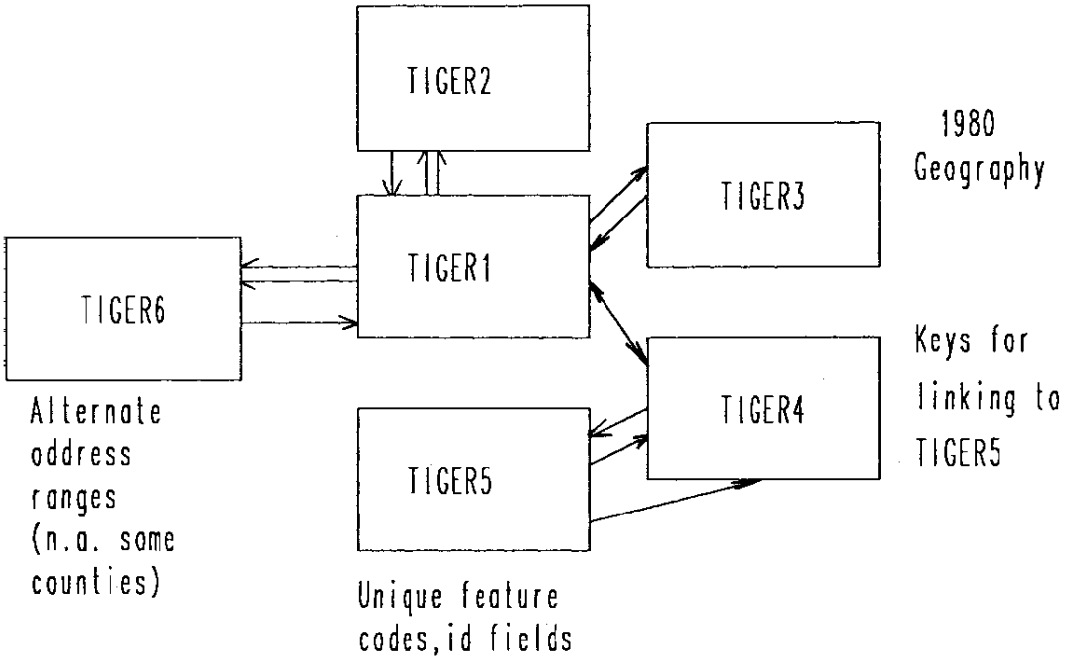
REFERENCES

Blodgett, J., "TIGER Demonstration Project: Summary", unpublished report, last revised 1/90 (available from the author via BITNET).

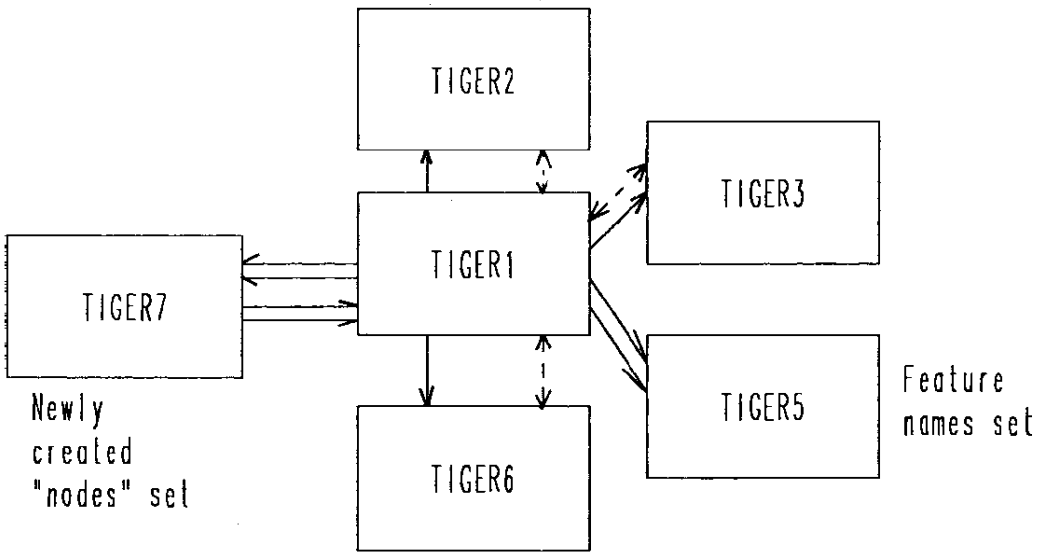
Carbaugh, L.W. and Marx, R.W. (1990), "The TIGER System: A Census Bureau Innovation Serving Data Analysts," U.S. Bureau of the Census, Washington, D.C. (Available through Data User Services Division).

U.S Bureau of the Census (1989), "TIGER/Line Precensus Files, 1990-Technical Documentation," Washington, D.C.: U.S. Bureau of the Census.

PHASE 1: CONVERT SEQUENTIAL FILES TO POINTER-LINKED SAS DATA LIBRARY

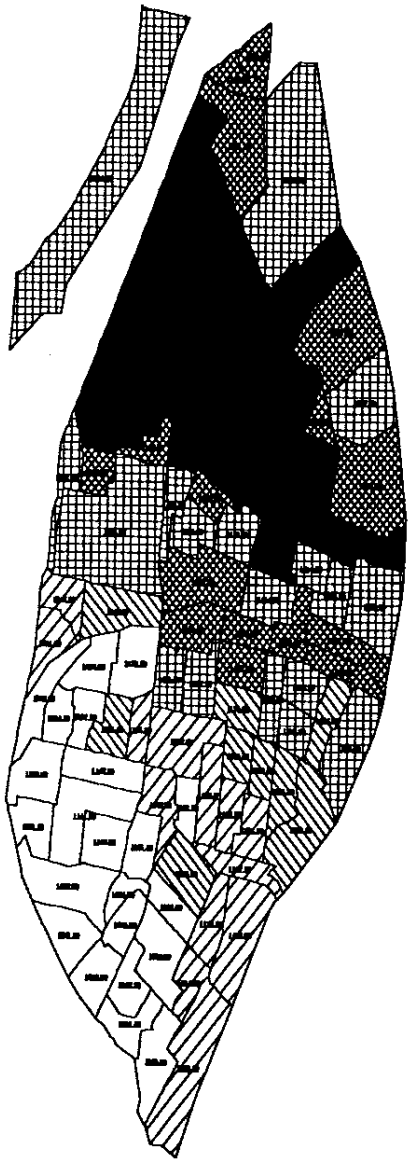


LINKAGES OF RESULTING SAS DATA LIBRARY(—— POINTER LINKAGE,---- MERGE LINKAGE)



CITY OF ST. LOUIS CENSUS TRACTS

PERCENT BLACK



PCTBLACK Less than 1% 1-5% 6-20%
21-50% 51-90% 91% or more

MAP PRODUCED BY THE URBAN INFORMATION CENTER, U.M.- ST. LOUIS
POLYGONS WERE GENERATED FROM THE PROTOTYPE TIGER FILE FOR ST.LOUIS