A POLYGON TRUNCATING TECHNIQUE PROVIDES ZOOM FOR PROC GMAP

Gregory C. Decker, Burger King Corporation

ABSTRACT:

PROC GMAP does not provide the establishing of a viewport in the world units of the map data. Hence it is not possible to extract a subarea of a map(zoom) by the bounds of some window which are independent of a maps polygons.

This paper describes a single data step that provides line clipping and polygon reconstitution. One can then graph only what one wants to see at the scale desired. Response variable features of PROC GMAP, as all other features of PROC GMAP, are retained. A working knowledge of PROC GMAP will be assumed.

INTRODUCTION:

In this paper I am going to explain the concepts and utility of subsetting a map data set by stipulating the four corners(bounds) of some desired window or viewport in the units of the map data set to allow PROC GMAP to graph only and exactly what one wants to see. If the chosen bounds are larger than the maximum limits of the map data set one has a zoom out or shrinking of the image. If they are smaller then one has a zoom in or blow up of a subset of the image. The ability to zoom is a function of being able to control the scale and the inclusion and exclusion of the data that will make an image.

A key concept to the understanding of this topic as well as PROC GMAP itself is a polygon. A polygon is a closed plane figure bounded by straight lines. PROC GMAP extends that definition in that a polygon may also be a straight line or a point. A map data set is a list of coordinates that are the end points of straight lines. PROC GMAP accepts this list of points as input and draws a line from point to point. The SEGMENT variable identifies to PROC GMAP a list of points that will make a contiguous plane figure which is a polygon. PROC GMAP graphs polygons!

Subsetting polygons to graph a smaller area works well until one needs to graph partial polygons. This need arises from the fact that a rectangular boundary will usually truncate a map made up of one or more irregularly shaped polygons. One can not simply eliminate those points from the data set that fall out side the limits of the bounds. PROC GMAP's single-mindedness will draw from point to point giving a distorted and incorrect image unless additional points are included to correct the graphed map.

THE PROBLEM:

The ability to zoom is strictly a user responsibility of adequate map data preparation. An algorithm can provide the proper manipulation of map data sets so that by only inputting the bounds of the window to be viewed all polygons that fall on the edge can be reconstituted. What is required is an algorithm to handle objects(polygons) that may appear full on-screen, partially on-screen and fully off-screen. We can narrow down the solution by keeping the objects that are fully on-screen and eliminating the objects that are fully off-screen. What is left are the partials to be truncated by our window. While testing for inclusion in our output map dataset we will make note of the first point that is within our window(there may only be one point). Later we will need to know that the direction of our first boundary crossing is from the inside to the outside.

Two problems must be dealt with in processing partials. The first is to calculate the point at which a line segment crosses the window boundary. This is sometimes call a 'push' or 'clip'; a foreshortening of the line segment. The algorithm presented here calculates this crossing point by solving the equation of the line that crosses the boundary at the boundary. The second problem is to include one or more window corner points if the polygon reenters the window on a side other than the one it exited from. This technique will provide the closure (reconstitution) that GMAP requires to retain the mapping integrity to properly graph our map.

THE PROGRAM:

I will not give a line by line explanation of this program but some explanations are in order. If you are not familiar with programming with the base SAS product this should not be your first program. It is written as one data step but could easily be two or more as well as an option to GMAP. The variables XMIN,XMAX,YMIN and YMAX are the bounds and are read in from SAS dataset LIMIT. STATE, COUNTY, TRACT and SEGMENT are ID variables. X and Y determine the boundary points of an individual polygon. Arrays XX and YY contain the entire list of points for an individual polygon. START is the subscript to the first point within the window. CURRENT and LAST indicate the status respectively of whether points are in the window(true) or out(false). CSIDE indicates which side of the window a line has crossed as LSIDE indicates the previous side crossed. CNT1-CNT4 handles the unusual case when a polygon circumscribes the window.

It should be noted that this algorithm expects a cartesian coordinate system. If the map data is in some other system it will have to be transformed and possibly projected. I usually convert my map data sets from coordinates in latitude and longitude to an offset in statute miles from some known and friendly point in the map data set. What the units are does not really matter. PROC GMAP will automatically scale from user units to the hardware units of the output device.

Most implementations of SAS restrict the number of numeric variables in a data step which will limit the number of points allowed per polygon. Some census tract polygons may contain well over one thousand points so that a PROC GREDUCE will be required.

NEW YORK CITY CENSUS TRACTS EXAMPLE:

Let me call your attention to figures 1 and 2. Figure 1 is the result of executing the first part of the truncator. One gets those polygons (census tracts) that fall totally within the window and those that fall across the boundary. It takes the clipper (LINK CLIP) to produce figure 2 with accurately reconstituted polygons.

Note that the scale in figure 1 is not determined by the window but by the reaches of the polygons. Figure 2 puts the user in control by controlling the bounds of the window rather than relying on the haphazard size and shape of polygons. An additional data step not discussed here produced the grid boxes which are 5 miles on a side and are also polygons.

APPLICATIONS AND NEW DIRECTIONS:

The most obvious application is to be able to subset ones map data strictly by coordinate bounds, usually some plus and minus distance from some known point. If one does this as a pattern as elements of an array of maps one can produce a very large scale map by piecing together the individual graphs which will now fit together exactly.

If you have a plotter with an appropriate SAS driver one can produce transparencies on velumn paper to any scale with the HSIZE and VSIZE GOPTIONS for map overlays to make digitization must easier.

The program presented removes polygons outside a window and keeps those that internal as well as restructures those that are partials. With a little recoding the algorithm can do the opposite; keep the polygons outside a window and remove the interior. This interior could contain another map or text.

CONTACT: Gregory C. Decker
Burger King Corporation
7360 N. Kendall Drive, M.S.
1235
Miami, Florida 33152
305 596-7039

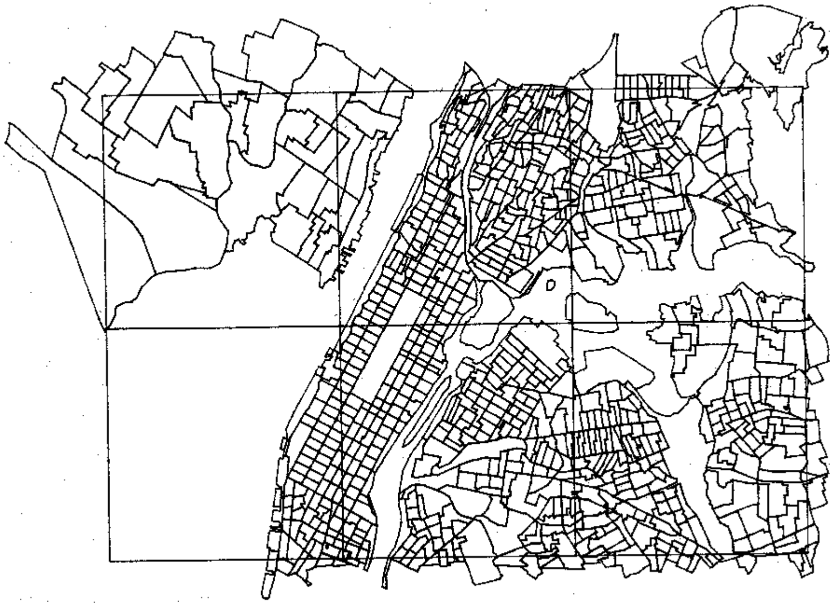
SAS is the registered trademark of SAS Institute Inc., Cary,NC,USA.

PROGRAM SOURCE CODE:

```
*.................................;
* P O L Y G O N   T R U N C A T O R ;
*.................................;
 DATA TRACTS(KEEP=TRACT)
      POLYMAP(KEEP=X Y TRACT SEGMENT);
 ARRAY XX(I) A1-A700;
 ARRAY YY(I) B1-B700;
 ARRAY XCORN(I) X1-X4;
 ARRAY YCORN(I) Y1-Y4;
 RETAIN LAST LX LY;
 RETAIN X1-X4 Y1-Y4 LSIDE;
 RETAIN XMIN XMAX YMIN YMAX;
 RETAIN CNT1-CNT4 0;
 RETAIN I J START 0
        A1-A700
        B1-B700;
   SET INMAP;
   BY STATE COUNTY TRACT SEGMENT;
   IF _N_=1 THEN DO;
         SET LIMIT;
         X1=XMAX; X2=XMAX;
         X3=XMIN; X4=XMIN;
         Y1=YMAX; Y2=YMIN;
         Y3=YMIN; Y4=YMAX;
      END;
   I=I+1;
   XX=X;
   YY=Y;
   IF START=0 THEN DO;
      IF X>=XMIN AND X<=XMAX AND
         Y>=YMIN AND Y<=YMAX
         THEN START=I;
   END;
   IF X<XMIN THEN CNT1=1;
   IF X>XMAX THEN CNT2=1;
   IF Y<YMIN THEN CNT3=1;
   IF Y>YMAX THEN CNT4=1;
   IF LAST.SEGMENT THEN DO;
      IF START>0 THEN DO;
         END=I;
         I=START;
         LX=XX;
         LY=YY;
         LSIDE=.;
         LAST=1;
         DO I=START TO END BY 1;
            LINK CLIP;
         END;
         DO I=1 TO START BY 1;
            LINK CLIP;
         END;
         IF LAST.TRACT THEN
            OUTPUT TRACTS;
         START=0;
      END;
      ELSE IF CNT1+CNT2+CNT3+CNT4=4
      THEN DO;
         X=XMAX; Y=YMAX;
                 OUTPUT POLYMAP;
         Y=YMIN; OUTPUT POLYMAP;
         X=XMIN; OUTPUT POLYMAP;
         Y=YMAX; OUTPUT POLYMAP;
      END;
      I=0; CNT1=0; CNT2=0;
           CNT3=0; CNT4=0;
   END;
   RETURN;
*-----------------------------------;
 CLIP:   /* POLYGON LINE CLIPPER */
   X=XX; Y=YY;
   IF X>=XMIN AND X<=XMAX AND
      Y>=YMIN AND Y<=YMAX
      THEN CURRENT=1;
   ELSE    CURRENT=0;
   IF CURRENT AND LAST
      THEN OUTPUT POLYMAP;
   ELSE IF ((CURRENT AND ^LAST) OR
           (^CURRENT AND LAST) THEN DO;
           IF CURRENT THEN DO;
              TX=LX;
              TY=LY;
           END;
           ELSE DO;
              TX=X;
              TY=Y;
           END;
           IF X=LX THEN DO;
              M=.;
              B=.;
           END;
           ELSE DO;
              M=(Y-LY)/(X-LX);
              B=Y-(M*X);
           END;
           IF TX<XMIN THEN DO;
              NY=(M*XMIN)+B;
              IF NY>YMAX THEN DO;
                 NX=(YMAX-B)/M;
                 NY=YMAX;
                 CSIDE=1;
              END;
              ELSE IF NY<YMIN THEN DO;
                      CSIDE=3;
                      NX=(YMIN-B)/M;
                      NY=YMIN;
                   END;
                   ELSE DO;
                      NX=XMIN;
                      CSIDE=4;
                   END;
           END;
           ELSE IF TX>XMAX THEN DO;
```

```
                NY=(M*XMAX)+B;
                IF NY>YMAX THEN DO;
                   CSIDE=1;
                   NX=(YMAX-B)/M;
                   NY=YMAX;
                END;
               ELSE IF NY<YMIN
                     THEN DO;
                        CSIDE=3;
                  NX=(YMIN-B)/M;
                        NY=YMIN;
                     END;
                     ELSE DO;
                        NX=XMAX;
                        CSIDE=2;
                     END;
             END;
             ELSE IF TY<YMIN
             THEN DO;
                CSIDE=3;
                NY=YMIN;
                IF M=. THEN NX=TX;
                ELSE NX=(YMIN-B)/M;
             END;
             ELSE DO;
                CSIDE=1;
                NY=YMAX;
                IF M=. THEN NX=TX;
                ELSE NX=(YMAX-B)/M;
             END;
        IF LSIDE^=. AND LSIDE^=CSIDE
           AND ^LAST THEN DO;
           J=LSIDE;
           DO WHILE (J ^= CSIDE);
              X=XCORN;
              Y=YCORN;
              OUTPUT POLYMAP;
              IF J=4 THEN J=1;
              ELSE J=J+1;
           END;
        END;
        X=NX;
        Y=NY;
        OUTPUT POLYMAP;
     END;
IF CSIDE ^=. THEN LSIDE=CSIDE;
IF CURRENT AND ^LAST THEN DO;
   X=XX;
   Y=YY;
   OUTPUT POLYMAP;
END;
LAST=CURRENT;
LX=XX;
LY=YY;
RETURN;
```

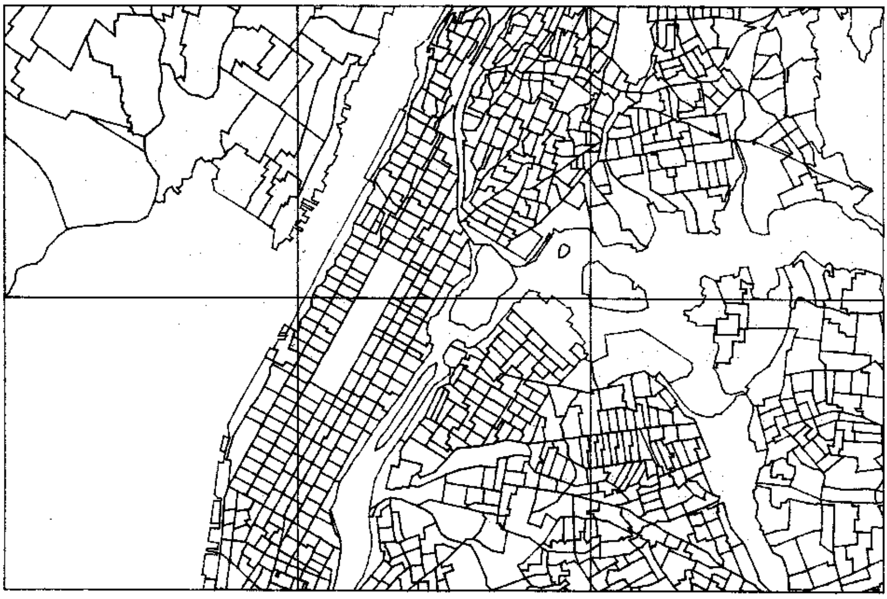
SELECTED CENSUS TRACTS IN NEW YORK CITY



CENSUS TRACTS THAT ARE WITHIN OR ON BOUNDARY WINDOW

FIGURE 1

SELECTED CENSUS TRACTS IN NEW YORK CITY



CENSUS TRACTS THAT ARE WITHIN BOUNDARY WINDOW

FIGURE 2