COMPUTER MAPPING AND ITS IMPACT ON KANSAS CITY, KANSAS AND WYANDOTTE COUNTY

Ronald E. Domsch
Wyandotte County Base Mapping Program

Kenneth D. Mai
Department of Planning and Development

This paper is divided into two logical sections. The first section will deal with the base maps and tools available to local users and the software developed to utilize them. The second section will outline some of the applications of these items and the local impact of the applications as a whole.

MAPS, TOOLS, AND SOFTWARE

The developmental work in computer-aided cartography locally is focused around two separate but related geographic base files. The first is a GBF/DIME-File on which work began in February, 1972. The second is a parcel file 1/ which was initiated in August, 1973. No applications of the parcel file have been made to date inasmuch as research and development of the file itself is not quite complete. However, most of the mapping related software developed for the DIME file is applicable at the parcel level.

Both the parcel file and the DIME file are based on a single base map used by most governmental entities within the city and the county. This base map consists of a series of maps each covering a quarter-section of land at a scale of 1:1 200. The maps were produced by sterophotogrammetric methods. 2/ The maps depict all planimetric features, are contoured at two-foot intervals, are gridded with the Kansas Plane Coordinate System, and contain other cosmetic and notational features. We feel that the success of our base mapping programs to a great extent is based on these highly accurate base maps.

The DIME file was developed from the original file created for the Kansas City Metropolitan Area by the Bureau of the Census and is maintained using Bureau of the Census standards although in a slightly different format. Several additional fields are also present. One new concept implemented here in conjunction with our DIME file is what we have termed the Contour Node File. For each non-linear DIME segment this file contains a list of additional coordinates more extensively describing the feature. These points as well as the original DIME nodes were digitized from the base maps mentioned earlier.

Editor's Note: Examples of the maps discussed, the GRAFPAC subroutines, and selected software for shading polygons are provided on pp. 475-481. An example of a pen plotter-produced choropleth map can be found on p. 263.

With these products and tools as a dependable base, a number of computer cartographical programs were developed for informational display purposes within the city and the county. These are choropleth mapping, dot mapping, and street network mapping all with a pen plotter. Additional mapping programs utilized but not developed locally include CHORO, 3/ GRIDS, 4/ and SURFACE. 5/ The graphics software developed locally is all written using a graphic output system called GRAFPAC II.

GRAFPAC II consists of about forty FORTRAN subroutines designed to communicate graphic output information from a compiler-level language to a graphic output device. The system has been implemented on IBM, Honeywell, and CDC computers and a number of different plotters including Calcomp, Benson/Lehner, and UCC. Although thorough documentation is not known to exist, the basic documentation of the subroutines as implemented in Kansas City has been completed.

GRAFPAC II has its origins in a set of specifications proposed by the SHARE Standard Graphic Output Language Committee entitled GRAFPAC, Specifications for Standard Graphic Output Subroutines. 6/ GRAFPAC software was originally written at the University of Kansas by Dr. F. James Rohlf and others. A major rewrite of the early software by Carl Youngmann and Stephen Hollis at the Institute for Social and Environmental Studies resulted in GRAFPAC II. We are utilizing a slightly modified version with several local additions.

CHOROG is a FORTRAN mainline with subroutines designed to produce choropleth maps on a pen plotter. The input consists of SYMAP-type outlines of each area, data values for each area, legend information, and a set of parameter cards establishing various map characteristics, data classes, and shading patterns. A typical map of Wyandotte County produced on a layout of typewriter page size requires about five minutes time on an IBM 370/145 under DOS/VS and about ten minutes plotter time on a Calcomp 905/936 system. Of technical interest in CHOROG is the method used to shade polygons and to describe the shading. In order to clarify following terminology, each polygon is to be shaded with a particular "pattern" of lines where each "pattern" consists of one or more "shades." Each "shade" is a set of evenly spaced parallel lines. The description of the pattern for a particular data class consists of one or more five digit codes each of which describes a shade. The five digit code has the following interpretation:

- Digit 1 - the pen or "color" number for this set of parallel lines (zero implies that the shade is blank and no lines are drawn).
- Digit 2 - the coded angle (counter-clockwise from the positive X-axis) of the lines.
- Digit 3 and 4 - the spacing in hundredths of inches between successive lines (zero is interpreted to mean one inch).
- Digit 5 - the coded fractional offset from normal at which to begin drawing the evenly spaced lines.

The explanation of digit 5 is obscure at the very best, and can best be explained by its usage. In order to improve the appearance of an output map it is desirable to have adjacent polygons with identical patterns have their shade lines merge at the boundaries rather than miss by a random distance. To allow for this cosmetic refinement in the output all shade lines begin at an appropriate integer multiple of their spacing parameter (digits 3 and 4) from the origin of the coordinate system used to define the map. This alone, however, will not accomodate a pattern consisting of two shades which are co-parallel and which require alternating lines to be of different pen widths or colors. One of those sets of parallel lines must begin at the integral multiple of the spacing plus one-half of the spacing. Patterns with three sets of co-parallel shades require additional offsets of one-third and two-thirds for two of the shades. Digit 5 is a coded number which allows the offset spacing required for this type of shading.

DOTMAP is a FORTRAN mainline with subroutines designed to produce dot maps on a pen plotter. DOTMAP does not produce maps from point data which would then be clustered into "dots" as one might suspect. Rather, it provides another means of displaying data aggregated by area. For each polygon in the map a number of dots is calculated from the input data value for the polygon. This number of dots is randomly distributed and mapped in the polygon. The input is identical to input for CHOROG with two exceptions. The definition of class levels and their corresponding patterns are replaced with the number of "things" to be represented by a single dot. The other exception involves a modification which has been implemented but is not thoroughly tested at this time. It concerns the input of "exclusion polygons." The input of one or more exclusion polygons defines areas on the map in which no dots are to be placed. This modification is being made at the behest of users who did not appreciate dot maps displaying people living in the middle of the county lake or with residential housing units in the airport runways.

MAPDIME is a loose-knit collection of FORTRAN mainline programs with subroutines which map sections of the DIME file and optionally label streets with their name or other information, optionally draw node numbers at nodes, and optionally draw characteristically distinguished network lines with different colors or pen widths. Input to the programs consists of a DIME file and a total of two parameter cards. Additional input locally consists of our contour node file mentioned earlier. Parameters governing the placement of street names on network segments are embodied in the software but could easily be input. The parameters will not be explained in detail inasmuch as they are actually an implementation of a set of rules constructed to locate street names in an acceptable manner on an output map. These rules apply to an ordered set of segments which are linked end to end and which all have the same name. The software provides each such "chain" of segments (complete with name and coordinates) to the subroutines which label the segments. Multiple consecutive blanks are compressed within the name. The standard character size and the minimum character size are established at input. The rules of name placement follow.

1. Each segment with a length greater than a parameter-specified distance will have the name plotted on it.

2. The accumulated length of a series of segments on which no name is plotted cannot exceed a parameter-specified distance.

3. Each series of connected segments must have a name plotted on at least one segment unless the longest segment in the series is less than a parameter-specified distance.

4. Names are plotted on single segments and do not pass over endpoints unless the length of the name exceeds the length of the segment on which it must be plotted.

5. An attempt is made to plot all names on segments which will hold the name without reducing character size. In the event this is not possible, the character size and the spacing between characters is reduced within limits to make the name fit.

Node numbers are plotted at nodes in such a fashion as to not overlap segments or possible street names, although overlap does sometimes occur. Such mapping has not been attempted at map scales smaller than one inch to five hundred feet since the characters become too difficult to read. Output devices with better resolutions should be able to produce maps at slightly smaller scales than this. A great deal of reduction will not be possible without more complex software due to the average length of DIME file segments and the size of readable characters.

In addition to the programs discussed, a number of cartographical programs are used which do not produce maps. Rather, they use the DIME file linked with operational files such as the street inventory file as an internal map to perform functions which would otherwise be performed on maps. Examples of such uses will be discussed in the second portion of this paper which deals with the usage and the impact of the computer cartographical software used by Kansas City, Kansas and Wyandotte County.

USAGE AND IMPACT OF COMPUTERIZED MAPPING

The impact of computerized data processing and mapping has been felt in many applications in Kansas City, Kansas. Most of these applications have been developed in the last two years by the staffs of the Department of Planning and Development and the Wyandotte County Base Mapping Program. The GBF/DIME-File has been the base on which most of the graphics capabilities developed to date stand on.

Extensive corrections and revisions have been made to the DIME file originally released by the Bureau of the Census. The 1"=100' topographic base maps previously mentioned have made possible a high degree of accuracy in the Wyandotte County DIME file. Of particular importance to our computer graphics applications is the fact that all nodes in the DIME file have been digitized from the quarter-section maps instead of the Census Bureau's Metropolitan Map Series (MMS) maps. It is estimated that the nodes are digitized to an accuracy of $\pm$3-5 feet of their true location.

Auxiliary data files have been developed and keyed for use with the DIME file. The contour node file, containing about twice as many nodes as the DIME file, has been mentioned earlier in the paper. An address file containing a record for each address along a DIME file segment has been in existence for about three years. On its completion, the parcel file, also mentioned previously, will replace the address file.

During the summer of 1974 an existing file containing physical characteristics of the streets was keyed to the DIME file through a manual coding effort. The street width, curb width and height, type of surface material, base material, and last date resurfaced. This file has been especially useful to the Street Department.

A Highway Safety Act grant has been received from the Kansas State Highway Commission to cover half of the costs in developing a street sign inventory for Kansas City, Kansas. The data will be collected in the spring and summer of 1976 by taking films of all the streets (photologging) and extracting the data from those films. Information collected will include color, shape, size, and condition of the sign, the sign legend, distance from the street, height off the ground, and an x-y coordinate location. The street sign inventory file will also be keyed to the DIME file.

The most frequent use of computer mapping in Kansas City, Kansas has been the production of choropleth, or shaded-area, maps of both locally collected data and data from more common sources such as the 1970 Census of Population and Housing. These choropleth maps are most often plotted on the Base Mapping Program's Calcomp 905/936 plotter, but some are also done on the computer line printer. Most of the locally collected data begins as address-coded individual records. It is then matched to tract and block or other geographic areas using the GBF/DIME-File and the Census Bureau ADMATCH program. The data is then aggregated by geographic area and mapped.

Data aggregated to geographic areas is also sometimes mapped by using a random dot within polygon mapping program. This mapping technique gives the visual density effect of a dot map while protecting the privacy of the data by not mapping the exact location. Choropleth maps and dot maps have been produced from data from the local County Assessor's annual census, the Comprehensive Employment and Training Act (CETA) client monitoring system developed in Kansas City, Kansas, a local study of the developmentally disabled, housing code violations, and several maps have been made from Juvenile Court data.

A variety of applications has been developed that uses the DIME file directly for mapping. The basic map is a street map showing streets (and non-street segments, if desired) and has the street names plotted on the map. This mapping program will make the map at any desired scale. The only problems develop when the map is too small to show the street names. Small maps can be plotted without street names. For purposes of working with the DIME file, node numbers can also be plotted on the map.

As a front-end for the DIME file mapping routine, a DIME-in-polygon program has been developed. Given any arbitrary polygon it will extract all parts of the DIME file within that polygon. Any segments that cross the boundary of the polygon one or more times will be broken so that all parts of the segment within the polygon are retained. The DIME file produced by this program can then be mapped by the DIME file mapping routine. This process has been useful for making maps of the DIME file in sections. The accuracy of the digitizing can be checked by overlaying these plots on other base maps. Maps of the DIME file have also proven to be the best maps of the city available. In the last year the Street Department requested a set of DIME file maps to be used in street cleaning. The plots were made at 1"=500' and then cut into street cleaning areas. After being laminated in mylar the plots are a durable and useful map for the men cleaning the streets.

A related type of map was made for the Street Commissioner by plotting a map of the city at 1"=500'. Street names were plotted on the map and the color used to plot the street segment indicated the surface material of the street. This map was used to determine where the concrete and asphalt streets were located so that plans could be made for patching. At the same time that the map was made, the length of concrete and asphalt streets was computed to be used as part of the information in requesting bids for street repair.

A similar map was also plotted showing which streets had curbs on one, both, or neither side. This map was useful in setting up street cleaning areas because it is necessary to have curbs for the street cleaning machine to pick up dirt and sand.

Another useful application has been developed that does not produce maps as a final product. This program reads in a set of polygons and then processes the DIME file computing the length of the street segments in each area. The boundaries of the Street Department districts have been altered, based on data from this program, to make the areas more equal in size.

Plans are being made for many more mapping applications. For the parcel file, maps will be generated to edit the digitizing of parcel outlines and the entry of parcel data and to display parcel related data. Similar types of graphics will be developed during the Street Sign Inventory project. It is important to note that in developing the graphics software no attempt has been made to build a grandiose all-purpose system. Rather, small modules with single specific functions have been written. These modules can easily be combined in a variety of applications. By developing these applications in a modular fashion and basing them on the DIME file, it is possible to combine several different options on a map. Thus, a basic street map can be made, then street names can be added, node numbers can be added, and non-street features can be added. In the future it will also be possible to add street sign locations and parcel boundaries. It is this modular approach that has made computerized mapping flexible and useful to Kansas City, Kansas.

The past and proposed work in Kansas City, Kansas and Wyandotte County has demonstrated that an accurate DIME file can form a solid foundation sufficient to support a wide variety of mapping applications. It has demonstrated that this modular approach makes the application of computer mapping techniques very flexible in supporting many users. It has also shown that computer cartography is a very useful tool in bringing about increased awareness of the spatial nature of urban phenomena, both for decision-makers and technicians alike.

We believe that Kansas City, Kansas does not present a unique situation. The basic tools and techniques can be applied in any city. Using a solid foundation and modular approach, it should be possible to build up the necessary support and resources required to develop a strong and flexible computer mapping capability.

REFERENCES

1. See Thomas M. Palmerlee and Ronald E. Domsch, "A Parcel Identification and Data System: A Case Study in Implementing Information Systems," Papers from: the Eleventh Annual Conference of the Urban and Regional Information Systems Association, (Atlantic City, 1973), pp. 393-407.

2. The maps were produced by M. J. Harden Associates using precision aerial photography from a Zeiss RMK-15/23 camera with a six inch focal length lens. Wild B-8 and Kelsh stereoplotters were used for plotting. The ground control was of second order accuracy utilizing EDM equipment and one-second theodolites.

3. CHORO - Institute for Social and Environmental Studies, University of Kansas, Lawrence, Kansas.

4. GRIDS - Southern California Regional Information System.

5. SURFACE - Kansas Geological Survey, Lawrence, Kansas.

6. The draft of the GRAFPAC specifications was prepared in August of 1966 by P. Pickman of Bell Telephone Laboratories, Incorporated, Whippany, New Jersey. SHARE is the Society for Helping Avoid Redundant Efforts.

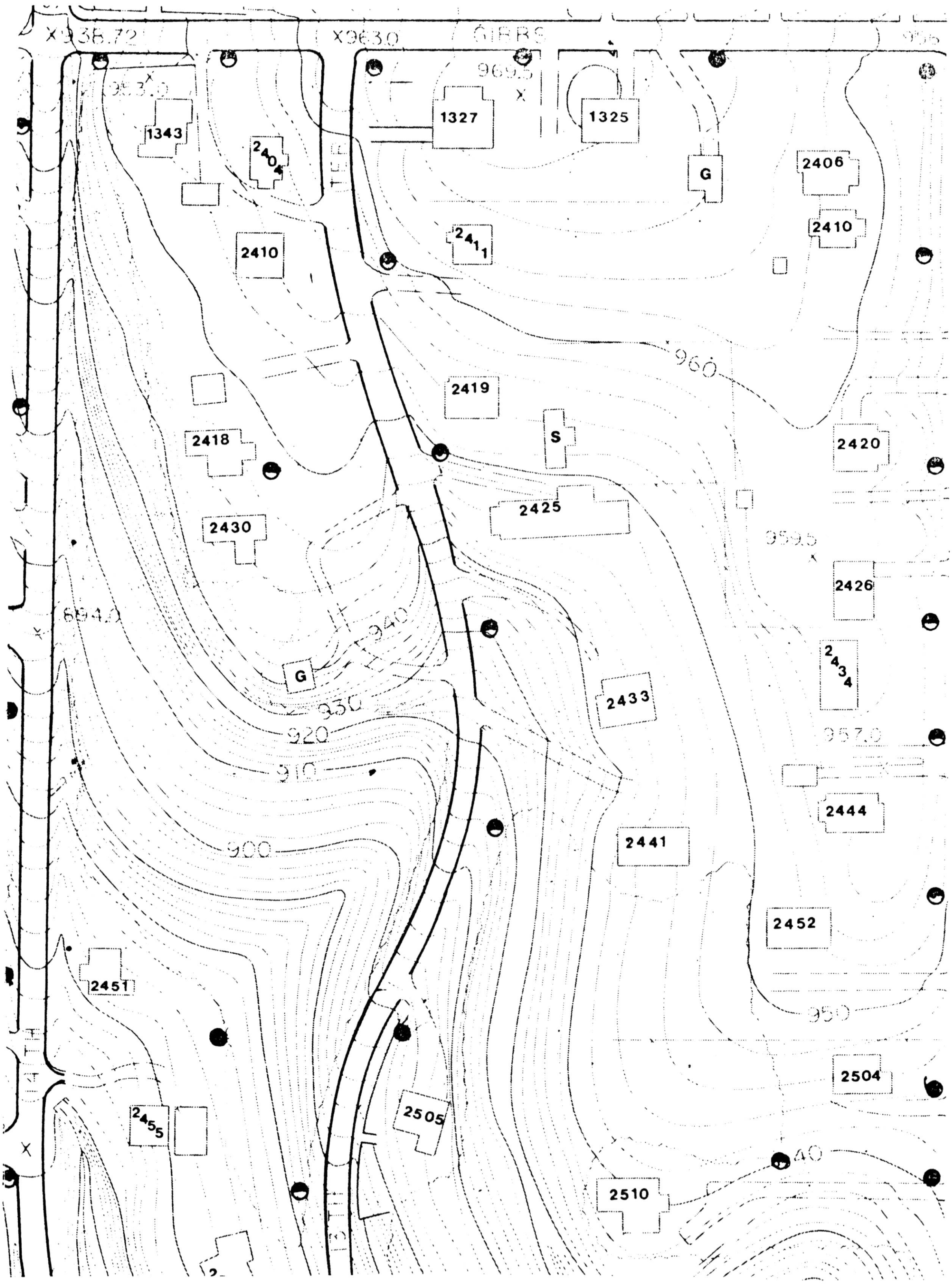

Figure 1. Topographic Base Map

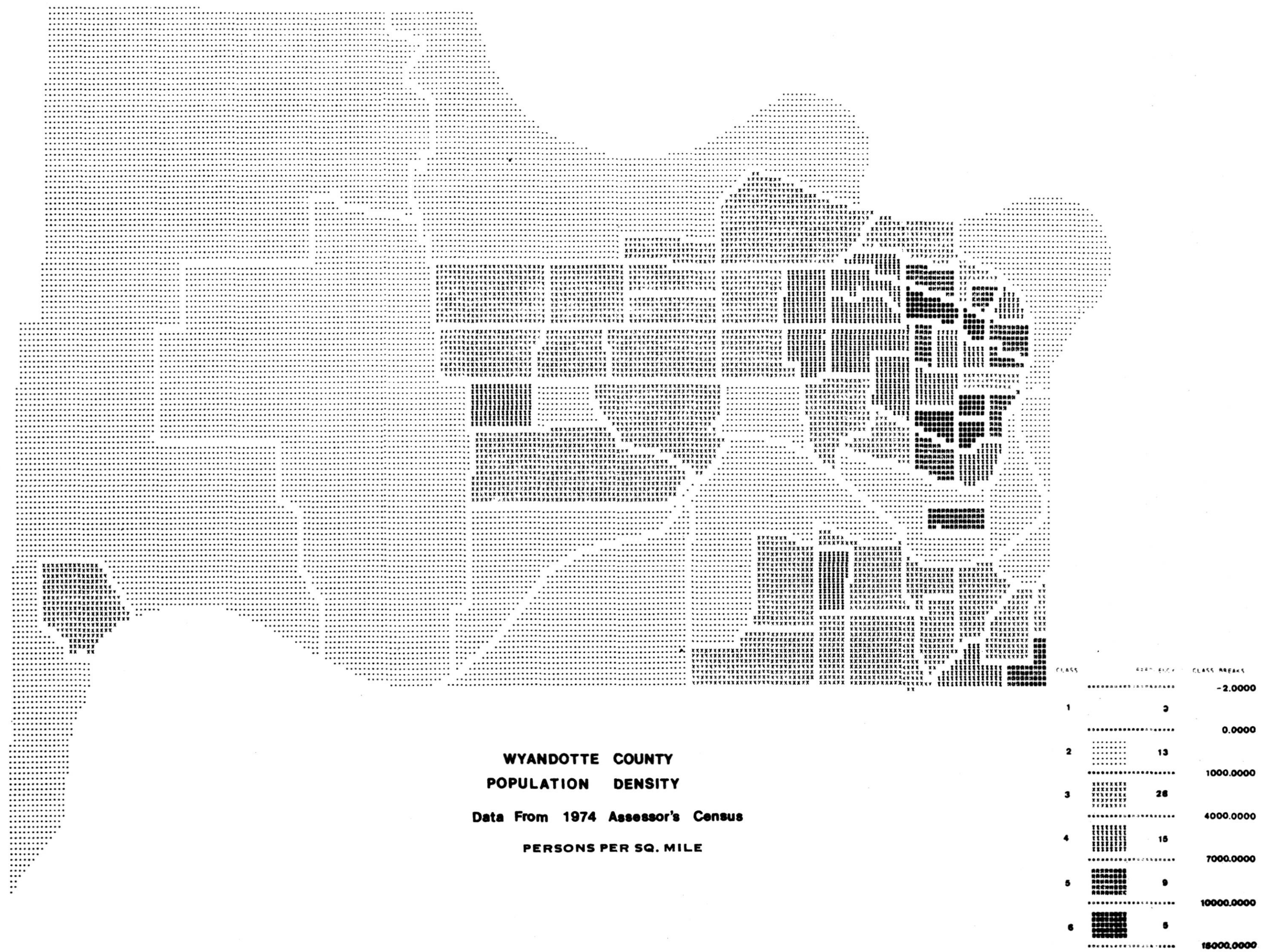

Figure 2. Line-Printer-Produced Choropleth Map

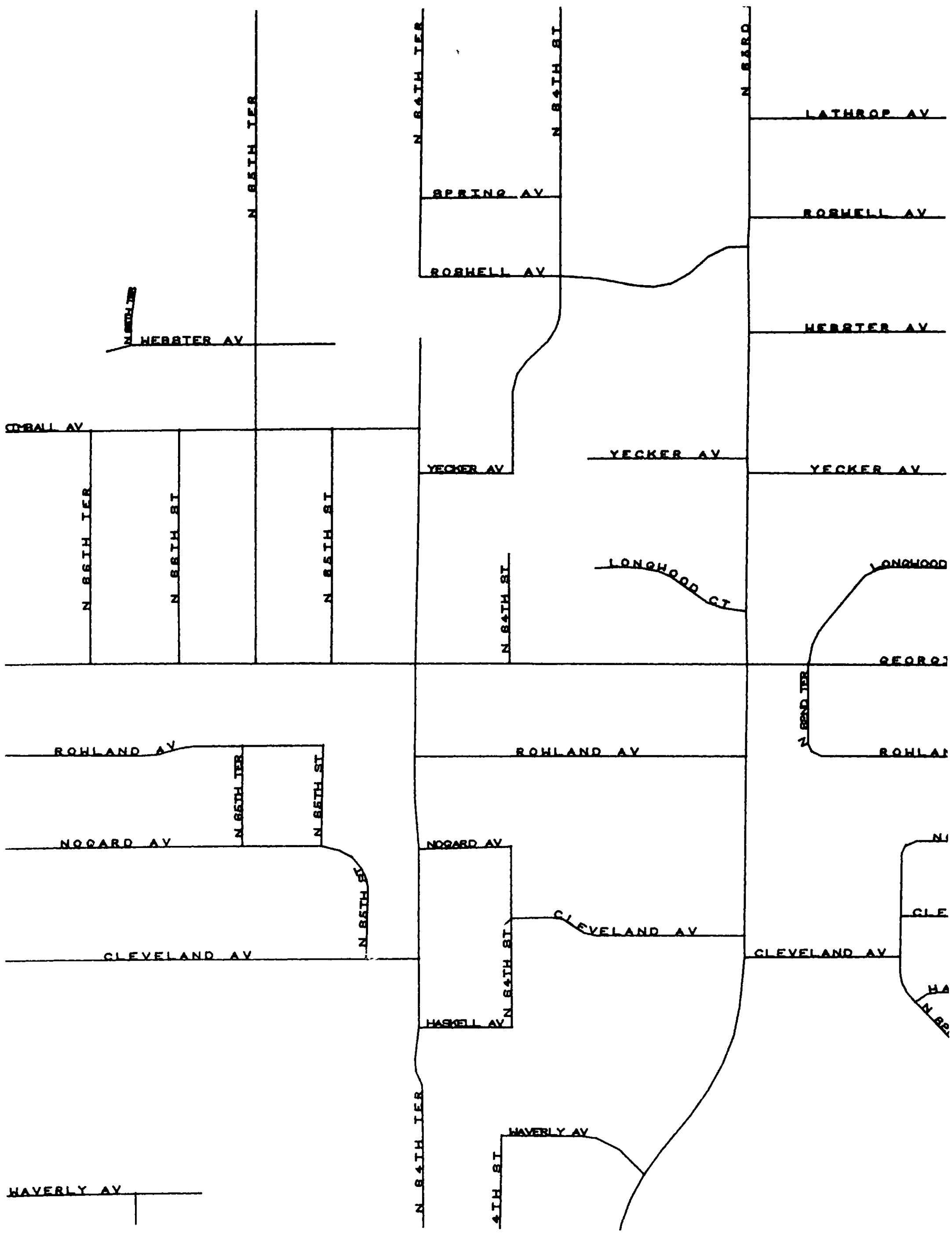

Figure 3. Street Map Generated from the DIME File

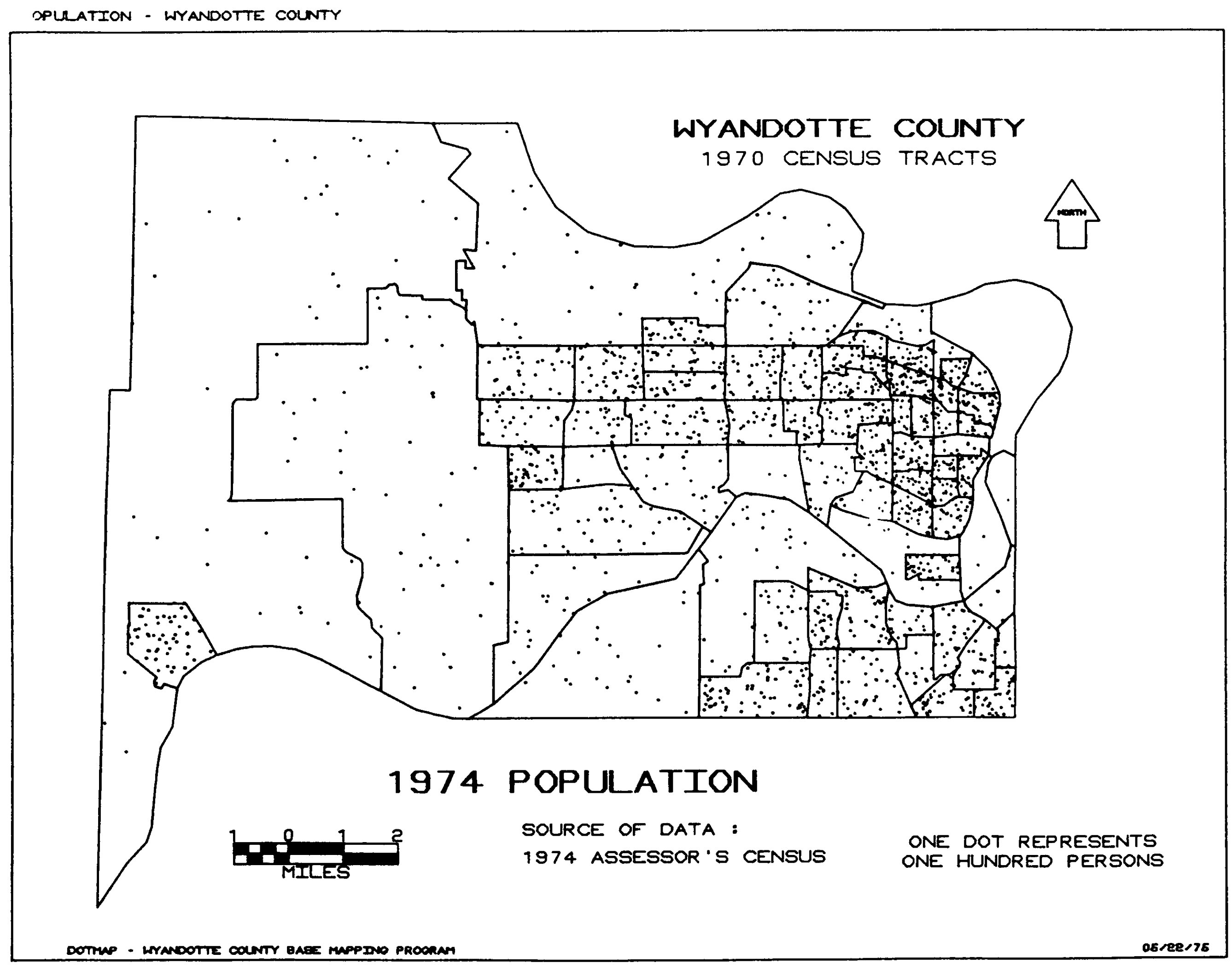

Figure 4. Example of dot map showing 1974 population of Wyandotte County

ADVANG - ADVANG completes plotting on the current drawing and positions the paper (or film) for the next plot. The plotter will then be set to prepare another plot the same size as the one just completed. (If another plot of a different size is to be made, OBJCG should be called).

ANGLEG - FORTRAN function to compute the angular direction of the vector formed by two ordered points.

BLIO - This subroutine is written in GMAP (the assembly language for the Honeywell-635). It performs the machine-level I/O for writing the plot instruction tape for the Benson/Lehner Draftomatic plotter.

BLPLTG - This subroutine sets up the actual plotter instructions for the Benson/Lehner Draftomatic Incremental, Off-Line, Rotary Drum plotter.

BOX - The routine computes the coordinates of the four corners of a rectangle.

CCINST, INSTCC - This is a GMAP (Honeywell-635 assembly language) subroutine which manipulates the bit configuration and stores instruction bytes used with the CalComp Remote Plotter/Controller Model 210.

CC210G - This routine creates and writes the plot instructions for the CalComp Model 210 Remote plotter.

CC763G, PLOT - This subroutine provides the interface between subroutine PLTITG and the CalComp PLOT subroutine for the various CalComp plotters.

CIRARG - This subroutine will plot a circular arc which is an exact curve within the physical limits of the plotting device. This is presently implemented so that the maximum distance between the true arc and the secant lines does not exceed .01 inches. (Although greater resolution could be achieved with most devices, the difference would be hardly noticable).

CIRTRG - This routine determines the (X,Y) coordinates of the end points of a line segment connecting the edges of two circles.

COMNDG - COMNDG reads a designated file containing "commands" for GRAFPAC. A limited set of frequently used functions is available.

CSIZEG - CSIZEG either sets the character size to the standard for the named device or sets the character sizes given by the user. Presently 0.16 inch is the standard size for all devices. The character size actually refers to the height and width of an envelope into which the character can be imagined to fit. The envelope includes clearance for both horizontal and vertical spacing.

CTABLG - CTABLG draws a legend table for the routine CHOROG or any other routine requiring a legend of shades.

DEVOFG - This routine ends the current plot and "turns off" the currently open device.

DEVONG - This routine sets parameters in the AMODES array which cause the specified device to be turned on.

DIVDG - DIVDG is called by LEGNDG to decode the internal codes used by GRAFPAC to define how to draw each of the symbols.

ERRORG - This routine writes out the error message, obtains a trace of subroutine calls in reverse order, closes all files, dumps the AMODES array, and aborts the job.

FPTBCD - This routine converts a floating point number (FORTRAN REAL) to BCD using a Fw.d format so that the number can be plotted using routine LEGNDG.

HAXESG, VAXESG - HAXESG and VAXESG are described together due to their relationship. They are called, generally as a pair, to draw horizontal and vertical axes of graphs.

FINISG - A call to this subroutine signifies the end of all plotting for the currently open device and output file. It is a signal to GRAFPAC II to perform the required action to wrap up the plotting output (turn off the device, rewind the plotting tape, etc.).

ICHARG - This subroutine obtains a specified "character" from a "word" and returns it in a new word left-justified with trailing blanks (FORTRAN format A1).

IDENTG - This routine labels a plot for identification and optionally creates a new plot number so the plotting device may be stopped for manual operation.

INCRPG - This routine is an internal routine used by GRAFPAC as an interface for plotting routines for devices for which the incremental pen movements must be supplied.

INTBCD - INTBCD converts an integer number to BCD (using an Iw FORTRAN format) so that the number can be plotted using the routine LEGNDG.

LEGNDG - This routine will plot a string composed of characters, numbers, and special symbols at a specified location.

LINESG - LINESG will move the pen to a specified point (with the pen up or down) or connect a series of points (pen down).

LNINRP - LNINRP determines that portion of a line which is interior to a rectangle.

MINMAX - MINIMAX determines the minimum and maximum values of the elements of an array.

NEWPNG - NEWPNG switches the designated plotting pen to the pen specified.

MODESG - MODESG initializes the parameters in the AMODES labeled common area.

OBJECG - OBJECG establishes the actual physical size of the object space and makes appropriate entries in the AMODES array.

ORIENG - ORIENG sets the orientation for plotted symbols to a specified value.

PATRNG - PATRNG calls SHADEG to shade a polygon with a series of shades to produce a pattern.

PCHARG - PCHARG "packs" a character into a designated position in a word.

PLTITG - PLTITG serves as the interface between all higher level GRAFPAC routines and the device - dependent routines.

POINTG - POINTG plots centered symbols at a series of specified points.

ROTAXG - ROTAXG rotates a series of points about a specified point by a specified number of degrees.

SEGMTG - SEGMTG plots a series of line segments defined by their end points.

SHADEG - SHADEG shades a polygon with a series of parallel lines at a given angle and spacing.

SMODEG - SMODEG initializes certain parameters in the AMODES array which are actually plot characteristics.

SUBOBG - SUBOBG calculates the subject to object space transformation values.

SUBJEG - SUBJEG establishes the limits of the subject space for the next plot.

SYMPTG - SYMPTG will plot a symbol defined by a relative coordinate system at the specified point.

TABDVG - TABDVG initializes that part of AMODES array which is unique to the particular device requested.

XINCHG, YINCHG - To convert an X or Y coordinate from an object space coordinate to subject space coordinate.

XUNITG, YUNITG - To convert an X or Y coordinate from a subject space coordinate to an object space coordinate.

Selected Software for Shading Polygons

```
DOS FORTRAN IV 360N-FO-479 3-8          PATRNG          DATE   11/09/74      TIME    14.39.01

0001          SUBROUTINE PATRNG(X,Y,N,IPAT,M)
        CPATRNG     *** GRAFPAC CHORO *** - SHADE A POLYGON WITH A PATTERN
        C
        C           RON DOMSCH / JUNE 1974
        C           WYANDOTTE COUNTY PLANNING BOARD
        C
        C     X,Y   - ARRAYS OF COORDINATES OF POLYGON PERIPHERY POINTS
        C     N     - LENGTH OF X AND Y ARRAYS
        C     IPAT  - INTEGER ARRAY OF SHADE CODES USED TO MAKE THE PATTERNS
        C     M     - LENGTH OF IPAT ARRAY
        C
        C     THE CODES IN 'IPAT' ARE 5 DIGIT CODES AS FOLLOWS - DIGITS 12345
        C
        C     ----------------------------------------------------------------
        C     | DIGIT 1       | DIGIT 2      | DIGITS 3&4         | DIGIT 5          |
        C     | (COLOR CODE)| (ANGLE)      |'SPACE' IN INCHES| (OFFSET)         |
        C     |----------------------------------------------------------------|
        C     | 0: NO LINES | 0:   0 DEG. | 01: .01 INCHES   | 0: 0   *SPACE |
        C     | 1: P1=C1    | 1: 30 DEG.  | 02: .02 INCHES   | 1: 1/2    "   |
        C     | 2: P2=C2    | 2: 45 DEG.  | 03: .03 INCHES   | 2: 1/3    "   |
        C     | 3: P3=C3    | 3: 60 DEG.  |                  | 3: 2/3    "   |
        C     | 4: P1=C4    | 4: 90 DEG.  |      .           | 4: 1/4    "   |
        C     | 5: P2=C5    | 5:120 DEG.  |     ETC.         | 5: 3/4    "   |
        C     | 6: P3=C6    | 6:135 DEG.  |                  | 6: 1/5    "   |
        C     | 7: P1=C7    | 7:150 DEG.  |                  | 7: 2/5    "   |
        C     | 8: P2=C8    | 8: INVALID | 99: .99 INCHES   | 8: 3/5    "   |
        C     | 9: P3=C9    | 9: INVALID | 00: 1.0 INCH     | 9: 4/5    "   |
        C     ----------------------------------------------------------------
        C
0002          COMMON /SHADE/ IPEN,THETA,SPACE,OFFSET
0003          DIMENSION X(N),Y(N),IPAT(M)
0004          DIMENSION DEGREE(8),FRACTN(10)
0005          DATA DEGREE/0.,30.,45.,60.,90.,120.,135.,159./
0006          DATA FRACTN/0.,.5,.3333333,.6666667,.25,.75,.2,.4,.6,.8/
        C
        C     DECODE THE 'M' SHADE CODES IN IPAT
        C
0007          NN=N
0008          DO 100 I=1,M
0009          I1=MOD(IPAT(I),10)
0010          I2=MOD(IPAT(I),1000)/10
0011          I3=MOD(IPAT(I),10000)/1000
0012          I3=MOD(I3,8)
0013          I4=IPAT(I)/10000
0014          I4=MOD(I4,10)
0015          IF(I4.EQ.0) GO TO 110
0016          IF(I2.EQ.0) I2=100
        C
        C     SET THE VALUES IN THE COMMON BLOCK SHADE FOR ROUTINE SHADEG
        C
0017          IPEN=I4
0018          THETA=DEGREE(I3+1)
0019          SPACE=FLOAT(I2)*.01
0020          OFFSET=FRACTN(I1+1)
```