CLASSLESS CHOROPLETH MAPPING WITH MILLIONS OF COLORS: A DEVELOPMENTAL SYSTEM

James R. Carter, Ph.D., Carolyn S. Bolton, and M. Elizabeth Wood
Geography Department, University of Tennessee
Knoxville, TN 37996-1420

ABSTRACT

Using a Vectrix color display system driven from an IBM PC/XT a classless choropleth mapping package has been developed. The system has evolved and there is no master plan to build a finished system. The display system has a resolution of 672 x 480 and can display 512 simultaneous colors from a palette of 16.7 million colors. The choropleth mapping package that has been developed on this system has been the product of many experiments by the author and his students. The paper traces the evolution of the package and discusses how the system is being employed in the teaching of computer graphics, teaching of cartographic design, and the conceptionalization of many expert systems to sort through many of the options available.

INTRODUCTION

When Tobler (1973) stated that computer technology would give us the ability to display a continuum of shades and that we would not need to deal with class-interval questions anymore, he was most prophetic. Using a medium resolution display capable of addressing 8 bits per primary color and displaying 9 bits per pixel provides the ability to generate more shades than the eye can differentiate and is a good example of the technology that Tobler anticipated. With such a system 256 gradations of shades can be displayed between any of the pure primaries or combinations of the pure primaries--black, white, red, blue, green, magenta, yellow, and cyan. By adding or subtracting tints of any primary to the basic gradation, an almost unlimited palette of gradations can be selected to portray any given choropleth distribution.

Three years ago the Geography Department at the University of Tennessee, Knoxville purchased an IBM PC/XT and a Vectrix Corporation VX-384 color display system. The VX-384 is a separate frame buffer residing external to the PC and driven by the PC through a serial port. The VX-384 outputs RGB signals to an Electrohome monitor. Recently, a duplicate version of this system was purchased using a Vectrix VXPC board set driving a NEC Multi-Sync monitor. Two years ago the Vectrix PaintPad system was added to the VX-384 system using a Summagraphics digitizing tablet. The newer system includes PaintPad under the control of a mouse.

All of the software that has been written was developed on the VX-384. The VXPC was configured to make use of the same software. All of the programs accessing the Vectrix VX-384 display are written in Interpreted BASIC, because the interpreter was readily available, the author and students had at least a rudimentary working knowledge of the language, and it was very easy to make small changes and quickly see the effects of the change. No thought has been given to changing to another language because the response times of the VX-384 are slower than the rate at which BASIC produces commands going the the 384. In some cases we have had to build delays into the software because the output from the interpreter overwhelms the frame buffer. By staying with the interpreted BASIC new students have been able to learn to program by building on the existing programs. However, as we consider extending the system to incorporate some expert sytems we know we will have to work in another language.

MATRICES OF COLOR POSSIBILITIES

To gain an understanding of how to address the full range of 16.7 million colors in the lookup table, a program was written to display simultaneously 8 matrices consisting of 8 rows of color squares by 8 columns of color squares. This, of course, pushed the display capabilities to a maximum since 8 x 8 x 8 = 512. Assigning the RGB primaries to range from a minimum of 0 to the maximum of 255 in eight equal steps gives a good overview of all color gradations that are possible with the system. By assigning a more limited range on any or all of the primaries, it is possible to 'zoom in' on a part of this three-dimensional color spectrum. The use of CTABLE has given us some insights into working in color space. These matrices show that there is a perceptual effect of one shade appearing to be lighter where it abuts a darker neighbor and appearing to be darker where it abuts a lighter neighbor. Another insight is that it is hard to find a good variety of gradations of browns and oranges on a RGB system even thought there are millions of colors to pick from.

POLYGON BOUNDARIES AND CENTROIDS

Coordinate boundary files have been employed for the counties of Tennessee and the states of the U.S. Both of these files were taken from files already used by the authors. All of these files were in a polygon format and were kept in this format, even though they require redundant writes for all interior boundaries. The outlines of the areas to be mapped were scaled to the screen and the coordinates were transformed using a simple scaling and translation routine. The senior author selected the scaling and position of the map areas and no provision exists in the package to permit the user to scale, translate, or rotate the subject on the screen. It is felt that there are more than enough variables for the user to address in the present system without having to deal with scaling and figure/ground relationships.

Many of the boundary files we started with were excessively detailed for the resolution of the screen and the scale of the map. Two students wrote an interactive program to reduce the number of points in a polygon boundary. This program writes a single polygon boundary on the screen and then steps around the boundary going point to point in response to touchs of the keyboard. At each point the user has the option of eliminating that point or retaining it. The process is subjective but it permits the user to build a visually appealing set of boundaries. A subjective line generalization procedure is not appropriate for boundaries stored in a polygon format because the subjective generalization of one section of an interior boundary is not always the same on both representations of that boundary. This program would work much better if the boundary files were in a chain format. But, for this situation, the program produced generalized files after a few iterations. All slivers were removed in these iterations.

The VX-384 employs a polygon fill routine that starts from any seed within the polygon. The polygon boundaries are written to the screen in a color different from the background. For each seed a screen coordinate has to be specified that falls within the polygon. The process of selecting seeds within the polygon was done manually for one cartographic data set and was done by calculating geographic mid-points for another cartographic data set. Where the centroids were calculated, corrections had to be made for a few irregularly shaped polygons. For Michigan, with its Upper and Lower Peninsulas, two seeds were required. Maryland required 3 seeds because in its western extent the width of the state is so narrow that in two places the north and south boundaries occupy neighboring pixels which has the effect of creating separate polygons relative to the fill primitive. In the mapping program the file of centroids is read each time new data are assigned to the polygons.

A number of data sets have been put together to use in this package. All data sets have been organized to have a one line entry for each state or county with the FIPS code and an alpha name in addition to one or more variables. In each case a separate input routine has been written to read each data set. This is quite cumbersome and we are working to build a single routine that will read from any of the data sets in an efficient fashion.

SELECTING COLOR SEQUENCES

The values are read into an array and in the process the minimum and maximum values within any data set are identified. These two extremes are used to calculate the range of data. In a FOR: NEXT loop the individual values for each state or county are assigned a position along the linear continuum between the extremes, a shade is calculated proportional to the position along the continuum, the coordinates of the seed are read, the cursor is moved to the seed position, and the polygon is filled with the appropriate shade. The program was identified as producing classless maps but in reality the data are divided into classes--128 to be exact. Where the shades are juxtaposed in a non-graded sequence such as occurs on the choropleth map no one can distinquish any single shade across the map. Thus, in reality, 128 shades form a continuum. In fact, 64 shades, or perhaps even 32 shades, would be no less effective when used in a choropleth map program of this type. The graphics system permits 256 separate shades to be defined, but only 128 were used in the program originally and there has been no cause to use the maximum number.

For a long time we used the default color look-up table as we wrote the first map to the screen, even though it produced meaningless maps. Most beginners had great trouble understanding what the system was all about when they began with this random display of many colors. So, we set up the color look-up table so that the first map seen on the screen employs a continuum from black to white. Beginners find it much easier to understand the function of the system when they are presented with this preselected example of color choices. And, in our opinion, the gray-scale map should be seen by all users anyway for it is a great place to start because it is neutral in terms of colors. In many situations, these gray-scale maps are more powerful than many of the colored maps.

Kimerling (1986 and 1985) has found that viewers do not differentiate the dark end of a linear gray shade continuum ranging from black to white on a CRT any better than they do on the printed page. This suggests that we should use a non-linear continuum with larger steps at the dark end of the scale and smaller steps in the middle and at the light end of the scale. However, once the first display is put on the screen, the user is given control of a program called CTABLE, which permits the user to set the colors at the extremes of the continuum. The user may choose to go from any color combination to any other, i.e., a light yellow-green to a golden brown. With such a color combination there would be no true dark end of the continuum, Thus, at the moment we see no reason to employ anything other that the linear division of the gray scale.

CTABLE is a color look-up table that walks the user through the process of selecting color schemes for the shading of the polygons, for the background areas, and for the boundaries and titles. We have found that after about 15 to 30 minutes with the program a person develops a feel for color space in an RGB system and starts to seek combinations that interest them. CTABLE starts by asking the user to specify Low Red, Low Green, and Low Blue as numbers between 0 and 255. Next the user is asked to specify High Red, Green, and Blue as numbers between 0 and 255. Within 30 seconds the map on the CRT changes to the new color sequence. The user can then change the color sequence again or can change the background color or can change the boundary color. Background and boundary colors are specified by giving a value between 0 and 255 for each primary. Some of the possible combinations are given below:

Numeric values of the primaries Red, Green, and Blue give

R	G	B		R	G	B	
0	0	0	to	255	255	255	gives black to white
255	0	0	to	255	255	255	gives red to white
255	0	0	to	0	0	255	gives red to blue
255	80	40	to	40	80	255	gives bright red to bright blue
255	100	100	to	100	255	255	gives salmon to bright blue
0	50	200	to	255	200	100	gives blue to tan

A variation of CTABLE asks the user to specify a range of 0 to 100 so that the user selects the intensity of each primary as a percentage. We have not noticed any improvement in people's ability to understand the task when working with the 0-100 scale, so we have tended to stay with the 0-255 scale. Quite a few users have been able to understand the bit configuration that is in operation when the 0-255 scale is used.

Providing a legend for the continuous tone maps has given some difficulty. Ideally, we think the legend should be a bar showing the full gradation used in the map. So far, we have positioned the bar in what looks like a balanced position relative to the map. We have made no provision for the user to shape his or her own bar legend or to place the bar where they feel it is appropriate. This may be an area for the next inventive student to take on. Assigning numbers to the bar so that the user can interpret values from the map is another problem. The easiest way to do this is to add the numbers using the PaintPad program, but this is out of keeping with the nature of an automated system. The numbers used should mark the end points of the bar in fairly round numbers so that the user can relate to the data, i.e., 16 - 48 rather than 16.17 - 47.69. Then there should be one or two additional numbers placed along the bar at significant points. What those significant numbers might be will vary with the range of values. Another problem of the same type is that of selecting, sizing, and placing a title. So far, we have employed the PaintPad program to do this and it works fairly well. But again, this is out of keeping with a fully automated system.

ADDING INTELLIGENCE TO THE SYSTEM

There are a number of places where expertise could be incorporated into such as system, particularly if the system is to be used as a decision support system. We have been looking at such situations and they have become a focus for our study of expert systems (Robinson and Jackson, 1985, and Pfefferkorn, et. al., 1985). The division of the legend bar is one such place where expertise could be incorporated into the process. The selection and placement of titles is another. Still another might be in selecting palettes that might be tuned to the particular user. A persons using such a system to get map information is probably not interested in playing with the 16 million colors. What this person really wants is a map that communicates that message to tell him or her what they think they want to know. What colors will be best to do this? We may never be able to find out which colors are best but we think we could find out that certain color sequences are more effective for a particular individual than are other sequences. If this is true, then a system could be tuned to the decision maker using the system. We think it might be possible to test the user of a decision support mapping system in the early part of a session so that subsequent activities will take place employing the more effective color palettes. Building such an expert system is beyond our immediate goals, but it is educational to contemplate how such interactivity might be incorporated into the map reading process.

Data selection is another area where expertise needs to be brought into the thematic mapping process, particularly if the mapping is being done as part of a decision support system. As a point for discussion, assume we have a data base with 100, 200, or 500 variables that might be called up by title or index number. How does the user sort through this collection and select the one or two variables that might address the topic he or she wants to address? Many times the senior author has worked with persons who had specific ideas of what they wanted to see in the way of a map. After showing these persons a map or two and engaging them in some directed conversation, it often comes out that there is another map that is more in keeping with what they really want. This same problem will occur as the general public has access to some of the data bases that supposedly are now or soon will be a reality. Is there any way to transfer that knowledge-based dialog to an interactive program? (Smith, 1986). To help us understand this basic concern, we have created a simple data set that has enough power to let us explore the process.

For the 95 counties of Tennessee we assembled three variables: 1980 county population, 1985 county population, and the land area of each county in square miles. Such variables are likely to be found in any file. But, by combining these data we can come up with many other possible combinations that would be mappable. These are:

1 - the absolute count of persons, 1980
2 - the absolute count of persons, 1985
3 - absolute population change, 1980 to 1985
4 - relative population change, 1980 to 1985 as a %
5 - population density in 1980
6 - population density in 1985
7 - absolute change in population density, 1980 to 1985
8 - relative change in population density, 1980 to 1985 as a %

With these 3 basic variables we could also examine the relationship between population growth and population size. If we find that there is a significant relationship, then it would be possible to map the residuals from regression to see which areas fit the model and which areas show significant departures. There are many indices that can be derived from just these three variables. Add qualifiers of age, sex, race, migration, education levels, and income figures and the possibilities grow ever more cumbersome.

Now, where such data are available the burden is on the user to select which values should be read and to derive any combinations. Likewise, the burden is on the user to select a title, particularly when new indices are derived. The problem as we see it is to capture that reasoning, and dialog, that the expert would use to come up with a particular map or series of maps given a specific concern and a large data base. It seems to be an iterative process, where a person starts with one map and moves on to alternative maps representing the same data. Looking at an alternative may open a new avenue of exploration or it may take the inquirer back to the original or on to another variable or combination of variables. But, somewhere in the collection of data, assuming appropriate scales of analysis, there is normally information that if properly presented would help the decision maker make a more rational decision. By the same reasoning, it is probably true that shown the wrong data or even a bad representation of some relevant data, the decision maker may be mislead into making a less rational decision.

CONCLUSION

This developmental system has served many users. It has been a good training ground where students could get their hands on some hardware that had enough power to take them beyond their previous experiences. With each new development, another set of possibilities are opened up. Another student comes along and takes the system another step in another direction. And, in between these learning sessions, the system has been used to generate some effective slides for presentations.

REFERENCES

Kimerling, A. Jon, 1985, "Simultaneous Contrast and the Unclassed Choropleth Map," Technical Papers, American Congress on Surveying and Mapping, 45th Anual Meeting, 355.

Kimerling, A. Jon, 1986, Personal conversation.

Pfefferkorn, C., D. Burr, D. Harrison, B. Heckman, C. Oresky, and J. Rothermel, 1985, "ACES: A Cartographic Expert System," Proceedings, Auto-Carto 7, 399-407.

Robinson, Gary, and Michael Jackson, 1985, "Expert Systems in Map Design", Proceedings, Auto-Carto 7, 430-39.

Smith, Karen F., 1986, Robot at the Reference Desk, College & Research Libraries, Sept. 1986, 486-490.

Tobler, Waldo R., 1973, "Choropleth Maps Without Class Intervals?" Geographical Analysis 5:262-65.