

ENHANCEMENT AND TESTING OF A MICROCOMPUTER-BASED GIS FOR UNIVERSITY INSTRUCTION

Duane F. Marble
Sherry Amundson
Jatinder S. Sandhu
Geographic Information Systems Laboratory
State University of New York at Buffalo
Amherst, NY 14260

ABSTRACT

This paper traces the history of the Map Analysis Package and reviews the enhancements made to its user interface for university instruction use. Relationships between database size, available system resources and processing speed are discussed.

INTRODUCTION

University instruction in geographic information systems must be carried out in a laboratory environment where the student has extensive access to "hands-on" work with a GIS and extensive, real-world data sets. Major commercial systems are expensive to acquire and operate and are normally out of reach of most academic institutions. One answer to this problem is to utilize much less expensive software which retains the basic functional structure of the GIS at a level which permits the student to acquire a working knowledge of basic GIS operations.

The mainframe version of the Map Analysis Package (MAP) was created several years ago at Yale University by Dana Tomlin and Joe Berry. The program was widely distributed at a low price and has been used by many universities as a teaching tool in GIS. The FORTRAN source code was made freely available by Yale and, as a result, versions were ultimately available for most mainframe and minicomputers. However the mainframe version supported only character output on CRTs and line printers, documentation was limited, and no enhancements had been made since the original release of the software.

About two years ago MAP was rewritten in Pascal for the IBM PC and released as a commercial package. From the standpoint of university instruction this package (PMAP) had several severe disadvantages: it was much too expensive, the lower-cost instructional version (AMAP) was so limited as to be nearly useless, execution was very slow and the source code was not available for local examination and modification.

In 1985 Dr. Dana Tomlin (now at the Graduate School of Design, Harvard University) received support from IBM to translate the original FORTRAN source code to an IBM/AT environment. In his original, draft version much of the

code was little changed from the older mainframe version, but some color graphic enhancements were added. Harvard has since released an improved version of this software, including the FORTRAN source code, for a low price and is continuing development of the program (Tomlin, 1986).

Late in 1985 Dr. Tomlin kindly made a copy of the rough draft program and its source code available to the major GIS programs at Buffalo, Penn State and South Carolina. These institutions had a significant and immediate need for a very low cost teaching tool in the GIS area. Examination of the initial draft of the code indicated that several significant problems needed to be corrected before the program could be effectively used in a university instructional environment.

Among the general problems were a very restrictive user interface, new graphic functions which made use of the IBM Virtual Device Interface (VDI) drivers and a number of functional limitations which had been carried forward from the older, mainframe version (e.g., lack of an operational HELP function within the program). Because of time pressures, the three institutions began independent testing and enhancement of the draft Harvard software early in 1986.

Development Activities at South Carolina and Penn State

South Carolina undertook the initial effort to improve the draft program by quickly creating a basic working version which was sent to Buffalo and Penn State early in 1986. Additional work by South Carolina substantially increased the size of the analysis area supported (from 10K cells to about 60K), redesigned the user interface, and provided greatly improved facilities for data input. Several functions (e.g., error counting) were removed and a basic HELP function was added (Rasche and Cowen, 1987). This version was made available to Buffalo in August of 1986 when South Carolina temporarily ceased further development work on the program to concentrate upon database creation and program testing.

Penn State also made a number of changes in the original South Carolina version to support their immediate teaching needs in the Spring of 1986 and has since concentrated upon development of exercises to support their instructional work in GIS.

Development Work at SUNY at Buffalo

While South Carolina had made substantial improvements in the original Harvard code, analysis of their Fall 1986 release indicated that the result was still not suitable for instructional use within our particular instructional environment. (A faculty-wide PC lab currently supporting only monochrome graphics and strict university rules about purchasing copies of all software for student use.)

Among the major problems that remained were a continued dependence upon the expensive (\$499 list price) VDI drivers for graphics (which even with a substantial discount from IBM we could not afford to purchase in multiple copies for student use), screen designs that we felt still did not

make maximum use of the limited, low resolution graphics displays, and very limited internal and external documentation. Good internal documentation of the program was essential since we also intended to use the source code as a laboratory tool in advanced GIS courses, as well as a direct teaching tool in the introductory GIS applications course.

From an instructional standpoint, the package also suffered from a lack of well organized and documented data sets which could form the basis for student exercises. Our instructional methodology requires the databases to be large and representative of real-world problem situations. Two well-documented data sets had been created for foreign areas (Panama and South Thailand), but more were needed for adequate instructional work.

A major technical problem was replacement of the VDI drivers. Nearly all device-independent graphics packages for the IBM PC, such as the HALO package from Media Cybernetics and the various versions of VDI and GKS, suffer from one major flaw from a university instructional standpoint - they must be paid for! An examination of the software marketplace revealed one exception in the Fall of 1986: MetaWINDOW created by the Metagraphics Software Corporation of Scotts Valley, CA.* MetaWINDOW provides a fairly rich function set and, in one version, operates through a memory-resident driver which may be distributed on a royalty-free basis with developed applications. It also supports a wide range of graphics displays, but currently no hardcopy plot devices. The graphics calls may also be made from either Fortran, Pascal or C.

PROGRAM ENHANCEMENTS

Conversion to MetaWINDOW

The South Carolina version of the MAP package used GSS*CGI library routines and required their device drivers to be memory resident. These drivers supported only the Enhanced Graphic Adapter and could not be distributed with the executable version of the program.

The first phase in the conversion from VDI to MetaWINDOW graphics involved translating the MAP program from IBM Professional Fortran (a full ANSI standard compiler), to Microsoft Fortran (a subset of the ANSI standards). In most cases, this was a straightforward process except where opening files, string concatenation and passing huge arrays (more than 64Kb) was concerned.

The second phase involved converting each VDI graphic function call to a MetaWINDOW graphic function call. Here again, most operations were one-for-one translations. In the case of the COLOR command (Rasche and Cowen, 1987), no single MetaWINDOW function could replace the VDI function

* MetaWINDOW is a registered trademark of the Metagraphics Software Corporation.

used. The command had to be rewritten entirely.

Enhancing the User Interface

A well designed user interface can reduce the "learning" time of a system, increase human processing speed and reduce operator errors. Desirable things to have in a student use environment are: computer assisted startup, on-line help for the user, high levels of program reliability (error trapping) and the use of consistent function keys for basic operator inputs.

MetaWINDOW has a variety of advanced features that assisted us in improving the user interface. Some of these are the ability to have multiple ports on screen, user-defined virtual coordinates and clipping limits for each port, ability to open windows and key hit detection. This allows each key hit to be parsed and handled within the program. It is also possible to use multiple fonts of various sizes on any port.

Initial use of the program has been made easier by having an on-line help facility at the touch of a function key. This is especially helpful to the first time user. All databases in the current directory are displayed. The user selects a database for processing by simply highlighting it using the cursor keys. Once the MAP program is initialized, a map of the function key assignments is available as a pull-down window.

The screen design of the South Carolina version of MAP divided the screen into a display area, a legend area and a command line area (Rasche and Cowen, 1987). In designing the new screen interface, the bottom 5% of the screen area has been reserved for entry of user commands, leaving the remainder fully available for graphic output. Any map image drawn on the screen is scaled to the screen dimensions and centered. It remains on the screen until a new graphic command is executed or the user requests the screen be cleared. This allows the user to refer to the map image while formulating and executing subsequent commands. In addition any changes to the palette are reflected instantly on the map image. The text display is restricted to a window located in the top part of the screen which can be toggled on or off by the user to see the image underneath (see Figure 1). The length of a single line of text display has been increased to make commands with tabular output more readable.

Help for individual commands is now available at two levels. When help is requested through the EXPLAIN function, a full description of the command is displayed in the text window and a syntax diagram is displayed in another window located just above the command line. Both windows may be toggled on or off with function keys. There is no restriction on the length of the description because the user may page through it; it is truly an on-line manual. In addition the user may request just the syntax diagram for reference, midway through typing a command.

Epson compatible dot matrix printer and a jLaser laser printer interface. Also displaying the text in windows without erasing the background image requires that a substantial amount of memory be set aside for saving screen images.

PERFORMANCE EVALUATION AND TESTING

The Buffalo version of MAP for the PC runs on a minimum hardware configuration of an IBM PC/XT or AT or close equivalent, with 640K of memory, an optional 8087/80287 math co-processor, and a 10MB hard disk. Graphic output may be displayed on any device supported by MetaWINDOW, such as the Enhanced Graphics Adaptor (640 x 350 resolution), the IBM Color Graphics Adaptor (320 x 200 resolution), or the Hercules monochrome graphics adaptor (720 x 348 resolution).

The program allows the user to inadvertently initialize a database that exceeds available disk space. Therefore we have tested the software to identify the limits to data volume imposed by the system, and the level of disk storage required by different data volumes. In addition we have conducted various performance analysis tests to establish tradeoffs between data volume and processing speed, and between data volume and display resolution. The full results of these tests are available in a technical report released by the SUNY at Buffalo GIS Laboratory (Amundson, 1987). The report provides users with a guideline for setting limits on data volumes and for matching hardware configurations to their data needs.

The spatial database in MAP for the PC is composed of one or more maps of a common scale describing a single study area. The database is currently set to hold a maximum of 99 maps. Each map is divided into grid cells; each grid cell contains a two-byte integer. Dimensions of the grid are defined by the user when the database is created, and are the same for all maps in the database.

Limitations imposed by disk capacity

The total data volume may be obtained by multiplying 2 (the number of bytes per cell) by the map dimensions, by the number of maps. According to this formula a single map layer 200 cells long and 250 cells wide would hold 50,000 cells and would be 100k in size. Taken to the 99-map limit, this single database would require 10MB of disk storage! Users must be aware of the tradeoffs between the number of cells in a map and the number of maps they expect to create, keeping in mind that the entire database must fit within the limits of available disk space.

Limitations imposed by memory size

The dimensions of the database affect program size because the program reserves internal storage space for two maps at any given time. Given the size of the remainder of the program, the size of the MetaWINDOW driver, and the 640K limit on available memory, the maximum number of grid cells in a map has been established at approximately 40,000 cells. An overlaid version of the program has been created

as well, using the PLINK86 Plus* overlay linker. Because the overlaid version reduces the code in memory, a larger map size of approximately 64,000 cells can be accommodated.

Limitations imposed by screen capacity

The number of cells that can be displayed on the screen is limited by the screen dimensions of the graphics adapter. In the case of MAP for the PC, the current limit of 64,000 cells to a map does not exceed the screen dimensions of any of the display adapters supported by MetaWINDOW; overall screen capacity does not limit map size. However each of the map dimensions must be restricted if we wish to see the entire map at once. Map dimensions cannot exceed the corresponding screen dimensions without some loss of detail.

Impacts of map layer size on resolution and speed

The program scales the map to the screen dimensions. Therefore the map size determines the resolution of the displayed map. The more cells there are in a map, the more cells are squeezed into the screen area, and the finer the display resolution will be. The effect of various map sizes on screen resolution is illustrated in the GIS Lab Tech Report cited earlier.

Increasing the map layer size slows down the processing speed. This relationship is tested extensively in our system evaluation report. Using PFINISH**, a performance analyzer program, we have clocked the speed of map data transfers between disk and memory, and the speed of selected GIS functions on several different map layer sizes.

CONCLUSIONS

The enhancements made to MAP for the PC have produced an easy to use program that can run under various hardware configurations. Extensive internal documentation has been added to make future modifications easier. The discussion of limitations imposed on map size by disk space, memory size and graphic devices gives a useful guideline for running the program effectively. The present version provides a stable teaching environment for students in GIS applications courses and a well documented tool for GIS design courses.

REFERENCES

Amundson, Sherry, 1987. Performance Evaluation of the Map Analysis Package: A Microcomputer-Based GIS. Technical Report 87-1, SUNY at Buffalo GIS Laboratory, Amherst, New York.

* PLINK86 Plus is a registered trademark of Phoenix Technologies, Ltd.

** PFINISH is a registered trademark of Phoenix Computer Products Corporation.

Metagraphics Software Corporation, 1986. MetaWINDOW Reference Manual. Scotts Valley, CA: Metagraphics Software Corporation.

Phoenix Technologies, Ltd, 1986. Pfinish User's Manual. Norwood, MA: Phoenix Computer Products Corporation.

Rasche, Brian and David J. Cowen, 1987. An Integrated PC-based GIS for Instruction and Research. Proceedings, Auto-Carto VIII. Falls Church, VA: American Society for Surveying and Mapping.

Tomlin, C. Dana, 1983. Digital Cartographic Modeling Techniques in Environmental Planning, unpublished Ph.D. dissertation, Yale University.

-----, 1986. The IBM Personal Computer Version of the Map Analysis Package. Report GSD/IBM No. 16, Graduate School of Design, Harvard University.