The Apple Macintosh in Cartographic Instruction

prepared for the

Microcomputers in Cartographic Education Workshop

May 4, 1986
St. Paul, Minnesota

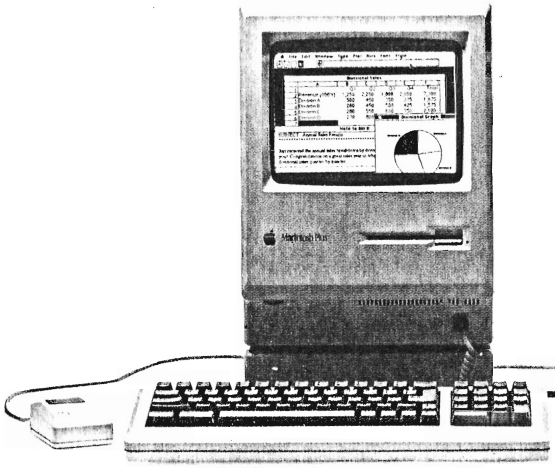
by

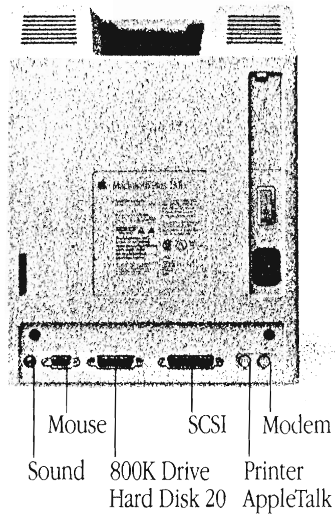
Dr. Michael Peterson
Department of Geography / Geology
The Remote Sensing Applications Laboratory
University of Nebraska - Omaha
Omaha, Nebraska 68182
Tel. (402) 554-2726

Outline of Topics:

1. The Macintosh Computer
2. MacPaint and Thunderscan
3. MacChoro
4. MacDraw
5. Atlas for the Macintosh
6. Programming the Macintosh
7. The Macintosh Toolbox

Macintosh Plus.



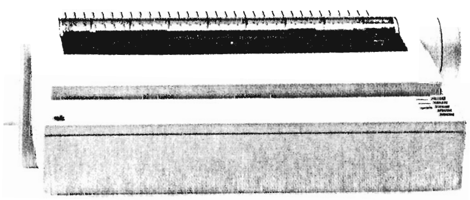


Macintosh Hard Disk 20

Plug the Hard Disk 20 into your Macintosh Plus, and its appetite for information increases by leaps and bounds. With a 20-megabyte capacity (that's about 10,000 pages worth), the Hard Disk 20 is the perfect place to store all your applications and files. Its Winchester-type technology lets you switch between programs or access information much faster than with a 3½-inch disk. And the Hard Disk 20 fits neatly under your Macintosh Plus. So it not only gives you more disk space to work with. It gives you more desk space to work on.

Apple ImageWriter II

The ImageWriter™ II printer lets you combine detailed graphics and near-letter-quality text in letters, reports, illustrations—



Apple ImageWriter II

LaserWriter. LaserWriter Plus.

Both share a sophisticated built-in page-description language called PostScript,™ which directs the printers to turn out text and graphics at the astonishing resolution of 90,000 dots per square inch. That's four times the resolution of ordinary laser printers (22,500 dots per square inch).

Both can print on plain copier paper, letterheads, labels, envelopes, and overhead transparencies.

And both can be very cost-effective solutions to an entire workgroup's printing needs. Thanks to a built-in AppleTalk connection, you can hook up as many as 31 Macintosh Plus computers—or even IBM PCs—to a single LaserWriter.

ITC AVANT GARDE
ABCDEFGHIJKLMNOP
abcdefghijklmnopqr

ITC BOOKMAN
ABCDEFGHIJKLMN
abcdefghijklmnopqrs

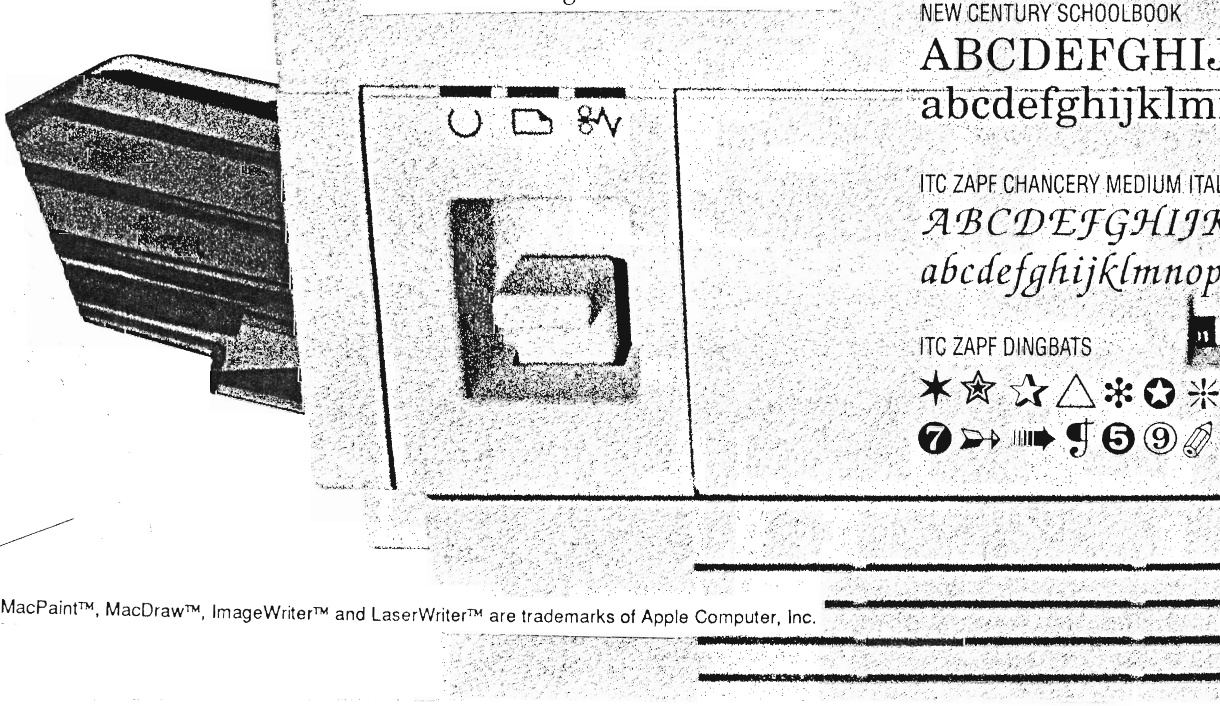
HELVETICA NARROW
ABCDEFGHIJKLMNOPQ
abcdefghijklmnopqrstuv

PALATINO
ABCDEFGHIJKLMNO
abcdefghijklmnopqrst

NEW CENTURY SCHOOLBOOK
ABCDEFGHIJKLMN
abcdefghijklmnopqrst

ITC ZAPF CHANCERY MEDIUM ITALIC
ABCDEFGHIJKLMNO
abcdefghijklmnopqrstuvw

ITC ZAPF DINGBATS



1.

...invited
...nth annual
...itectural Council

...arch 9 at 5 p.m.
...r at 6:30 and dinner at 8:30

...ton Pavilion, Blue Room
... Merchant Square

...et parking will be provided

Martin, James. A Breakthrough in Making Computers Friendly: The Macintosh Computer, Prentice-Hall, 1985.

THE MACINTOSH

The Apple Macintosh is an innovative computer that begins seriously to address the needs for both ease of use and enhanced function in the design of both hardware and software. At the time of this writing, much more application software is needed for the Macintosh, but Apple Computer provides tools with which this can be built and made to fit into the overall strategy for ease of use.

In this book, we will have much praise for the Macintosh. We will voice many complaints as well. To put things into perspective, let us state right now that we feel the benefits provided by the Macintosh far outweigh any of its shortcomings. In our opinion, the Macintosh represents the state of the art in small-computer technology.

Figure 1.1 shows a typical Macintosh system. We will next say a few words about each of the hardware components that make up the Macintosh.

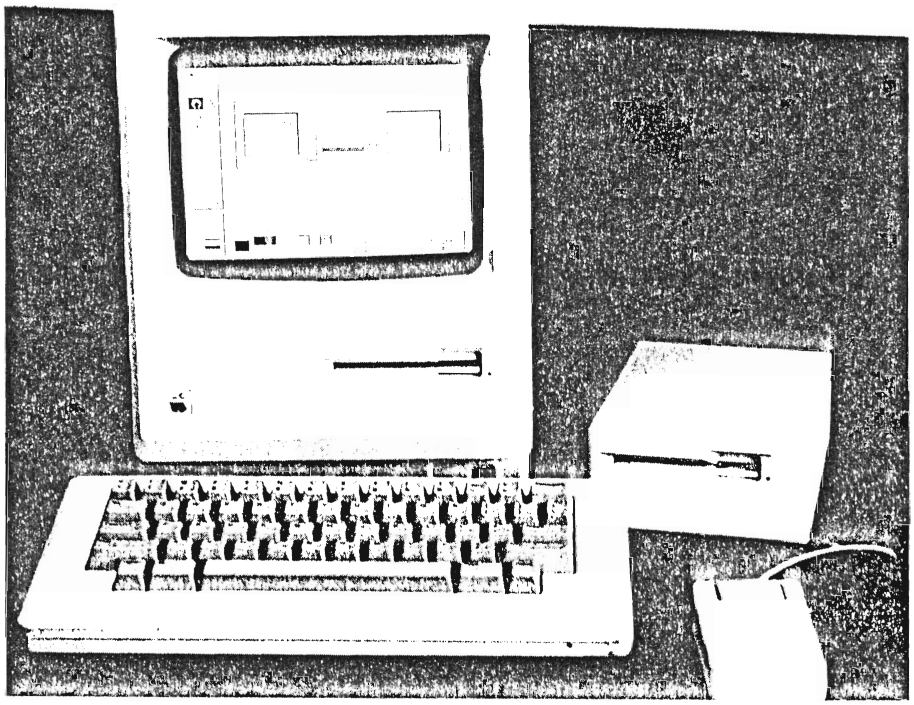


Figure 1.1 A typical Macintosh system.

We find that the mouse is an ideal pointing device and is generally superior for that purpose to joysticks, track balls, touch screens, light pens, and graphics tablets. The typical user is able to manipulate the mouse for most functions after a very short time. However, it does take quite a bit of practice to become really adept at using the mouse when creating or editing graphics using a program such as MacPaint. We feel that graphics tablets might be more useful than the mouse for detailed work with complex graphics, and we would hope that graphics tablets will become available as an option for the Macintosh. But the mouse remains a good all-around tool for controlling the Macintosh. When we switch over to one of our IBM or IBM-compatible computers, we often find ourselves unconsciously reaching for the mouse and being slightly disappointed when we realize that it is not there.

Much research has been done showing the importance of very fast interaction with computers [3, 4, 5]. Slow interaction tends to disrupt the thought processes of the user, severely degrading the user's performance and often the quality of his or her work. A mouse facilitates fast interaction with the machine. Using the mouse, pointers and cursor bars can be moved around the screen very rapidly. Clicking the mouse button on one menu item can instantly display additional choices.

A BRIEF HISTORY

INTRODUCTION

The Macintosh had its beginnings in research begun by Alan Kay at the Xerox Palo Alto Research Center (PARC). Kay's vision was a computer he called the *Dynabook*, a computing system with the power and speed of a mainframe in the size and shape of a notebook. The system would have a flat screen and be capable of storing vast amounts of data internally, as well as being able to communicate with networks. It would be so easy to use that anyone would be able to tap its power.

THE ALTO AND THE STAR

The research at Xerox, while it has not yet produced a Dynabook, did produce a number of revolutionary computers. One, the *Alto*, was released in the middle of the 1970s with a price tag of over $30,000. Another, released somewhat later, was the *Star*, with a price tag of roughly $16,000. The Star is still being used today and has many enthusiastic users. Both the Alto and the Star used a mouse, displayed information in windows, and supported pop-up menus that were instantly available on the screen. Much of the research that went into developing the user interface features of the Alto and the Star benefited the Lisa and the Macintosh.

Legend has it that in the late 1970s, about the time that the Lisa project was beginning, Steve Jobs, one of the founders of Apple Computer, was touring the PARC development laboratories with a number of other people from Apple. The PARC people demonstrated some prototype equipment that implemented windows and pop-up menus and used a mouse as a pointing device. Jobs is said to have pointed to the machine and said to the people accompanying him: "I want a machine like that." According to the story, that was when the concept for the Lisa was born.

THE MACINTOSH

The Macintosh was introduced in January 1984. Although it appeared later than the Lisa, it was not actually an outgrowth of the Lisa development. The two machines were developed roughly in parallel by separate development groups. The two machines did not begin to share development resources until quite late in the development cycle. The Macintosh project began in 1979 when Jef Rafkin was given the charter of developing a small computer that would retail for approximately $500 and would attach to an ordinary television set. The project operated for a time with just two other people. Rafkin tried to create a computer that would be as easy to use as any other electrical appliance. According to insiders at Apple, Steve Jobs, then a vice-president, was at that same time trying to get control of the Lisa project, then also in its beginning stages. But Apple president Michael Scott and marketing manager Mike Markula considered Jobs too inexperienced to be entrusted with a project so large and so important to Apple. They assigned him to the Macintosh project instead, replacing Rafkin as boss.

All through the early 1980s, there was fierce competition between the Lisa and Macintosh development groups. At one point, the Macintosh group even flew a pirate flag above their headquarters. It was largely through the efforts of John Sculley, new Apple president, that the two computers emerged as compatible as they are (although that still is not very compatible). Since they share the same microprocessor chip (the Motorola 68000), the Lisa was used as the development system for checking out the first Macintosh applications.

Finally, in 1985, Apple decided to integrate the Lisa completely into the Macintosh product line. The version of the Lisa that contains a 10-megabyte hard disk underwent minor engineering changes and was renamed the Macintosh XL. Apple then made available conversion software to convert files created under the Lisa Office System software into files that could be processed by roughly equivalent Macintosh software. Unfortunately, Apple soon afterward discontinued the Macintosh XL, and it is no longer being manufactured.

HARDWARE ARCHITECTURE

What is so unique about the Macintosh hardware architecture is that it packs so much computing power into so small a package that uses so few chips. Figure B.1 presents an overview of the Macintosh digital electronics.

The Macintosh hardware is based on the Motorola 68000 microprocessor. Although Apple bills the 68000 as a 32-bit processor, it would be classified by most hardware engineers as a 16/32-bit processor; it operates using a 16-bit address bus but has registers that can operate on 32-bit words.

The system's RAM is triple-ported, which means that the 68000, the CRT-display circuitry, and the sound-generating hardware all share the same RAM access lines, making it appear that the current program, the video, and the sound all operate simultaneously. The address line sharing is handled so that the 68000 has preferential access. The processor clock runs at 7.83 MHz, and the 68000 access to the RAM gives an average speed of about 6 MHz. The ROM, however, connects only to the 68000. Most of the system's time-critical code executes out of the ROM, giving that code greater execution speed.

Application code calls the subroutines in ROM through the use of 68000 instructions called "line 1010 unimplemented" instructions. These instructions get addresses from an address table stored in low memory, effectively allowing ROM routines to appear to the application code as extensions of the 68000 instruction set.

The CRT display is bit-mapped, with a resolution of 512 by 342 pixels. Each horizontal line of video is mapped to 32 16-bit words of data in memory. The entire screen image is mapped to 10,994 16-bit words in memory.

The sound-generating circuitry is sophisticated enough for simulating human speech, and programming has already been developed to give the Macintosh that capability. Pitch is controlled with 24 bits of precision using the 68000 32-bit registers. This makes possible four different "voices," each capable of generating 16,777,216 different frequencies. A timer also provides a simple square wave, the pitch of which is programmable. The amplitude of the generated sound has a 20-decibel range in eight steps.

The disk controller uses the chip referred to by Apple as the IWM ("integrated Woz machine"). It is a chip that reproduces the circuitry originally developed by Steve Wozniak, one of the founders of Apple, for the Apple II disk controller. It transfers data at up to 500 kilobytes per second and is capable of running the disk-drive motor at up to 400 different speeds.

The Macintosh handles serial communications with a Zilog 8530 chip. The 8530 handles synchronous and asynchronous communications at speeds up to 230.4 kilobits per second using a self-clocking data format and up to one megabit per second using an external clock.

The 6522 versatile interface adapter (VIA) provides I/O functions for system timers, the mouse, the keyboard, and for other general purpose I/O lines.

SYSTEM SOFTWARE

Much of the system software support for the Macintosh is stored in the ROM installed in each machine. The ROM augments the 56 standard instructions implemented by the 68000 microprocessor with more than 480 new instructions that are designed to implement the Macintosh user interface. The overall architecture of the Macintosh system software is illustrated in Figure B.2.

The Macintosh hardware interfaces with a number of low-level routines that make up the Macintosh operating system. Unlike other computing systems, none of the software that the user sees, including applications programs, interfaces directly with the Macintosh operating system. The Macintosh operating system is so transparent to the user that the designers did not even give it a name.

Between the application program and the Macintosh operating system is a powerful set of tools collectively called the "user interface toolbox." Most of the toolbox functions are performed by routines programmed into the ROM. It is the toolbox that gives the Macintosh its unique personality and helps all Macintosh applications operate in an integrated fashion.

Image Mapping Systems

presents

MacChoro

What is MacChoro?

A completely menu-oriented computer mapping, data classification and map-design program for the Apple Macintosh computer. **MacChoro** uses the excellent graphics capabilities of the Macintosh to create Choropleth maps. The choropleth method uses shadings to represent values and is one of the most common methods of statistical mapping.

Unclassified Maps -- Normally, data being mapped must be classified prior to mapping. Because the Macintosh is capable of producing so many dot shadings between white and black, it is possible to use the unclassed method of choropleth mapping. With this method, the intensity of the shading is proportional to the data value.

Four Methods of Data Classification -- MacChoro includes four common methods of data classification:

- standard deviation
- equal-interval (constant interval)
- quantile
- natural breaks (largest differences)

Each of these classification methods may be displayed with up to 16 classes. This means that you have a total of 61 classification options (unclassed plus 4 methods of classification with 15 classes each).

Speed -- Perhaps the most remarkable feature of **MacChoro** is its speed - a typical map is classified and displayed in approximately **3** seconds. This allows you to quickly view your data with a variety of classification options.

Data Manipulation -- MacChoro incorporates extensive data manipulation capabilities. Data may be modified through division, multiplication, subtraction or addition by another data set or by a constant. Logarithm or square root transformation is also possible. The data may be interactively viewed and edited and any modifications may be saved in the same or a different file. A data file can be initially created with **MacChoro** or may be extracted from a spreadsheet program such as Multiplan, Excel or Jazz.

Ease of Use -- All options are activated with the mouse. You'll only need the keyboard to enter new file names, enter data, enter constants for data manipulation and to enter text for the specific labelling of the map.

How does MacChoro work?

A **MacChoro** display is composed of 14 'elements' - map, legend, first neatline, second neatline and ten text elements. The user defines a rectangle for each element by moving and 'clicking' the mouse at the upper-left and lower-right corners. Map, legend and text elements are scaled, centered and placed within the user-defined rectangle. Elements can be outlined with a visible rectangle, moved to a new location or erased. In addition, the user-defined rectangles may be saved and used at a later time.

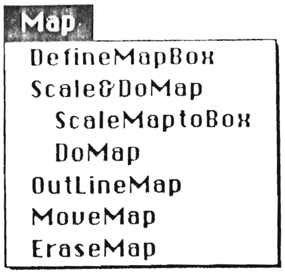
MacPaint your Map -- Once a map is created, it may be transferred to **MacPaint** for adding other text or graphics.

How can MacChoro be used? -- MacChoro is an excellent tool for data analysis. Maps can be created quickly and data can be easily standardized for more accurate mapping. Data analysis is enhanced by the ability to quickly view the map with a variety of classification options. **MacChoro** is also excellent for preparing maps for publication because of the high-quality of printed maps that can be obtained from both the Apple Imagewriter dot-matrix printer and the LaserWriter laser-printer. **MacChoro** is especially suited to analyze the effects of data classification. By using **MacChoro**, one obtains a true appreciation of the effects of the different classification methods on a mapped distribution.

Available Map Files -- MacChoro works with specially formatted map files. Map files currently available are:

- United States (included with **MacChoro**)
- Any of 48 states by county
- North America by country
- South America by country
- Asia by country
- Canada by province
- Europe by country
- Africa by country
- World by country

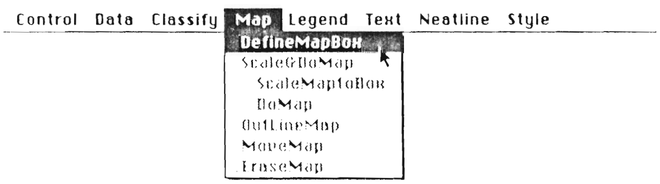
Map files not included here may be formatted by the user or through special arrangement with **Image Mapping Systems**.



The 3 steps to create a map

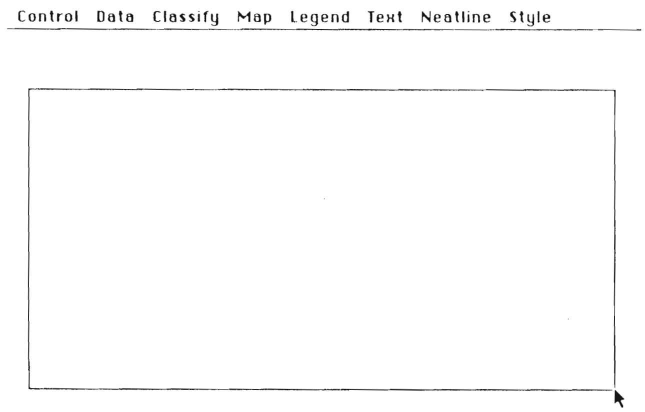
Step 1.

Select 'DefineMapBox' option from **Map** menu (after data file has been defined).



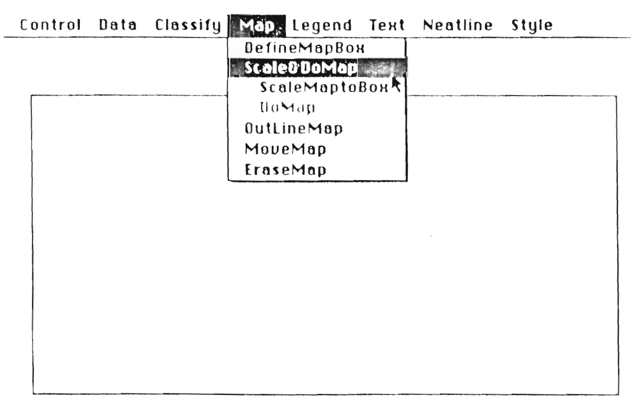
Step 2.

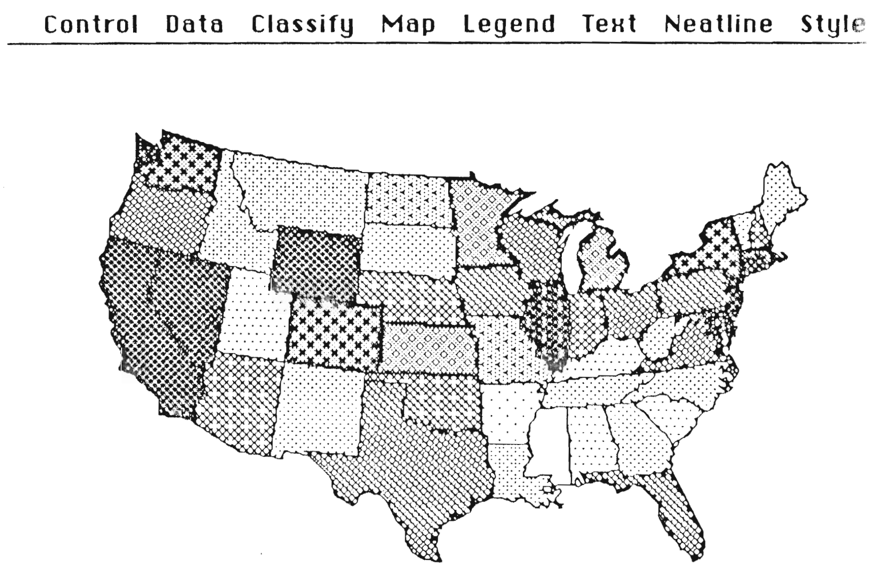
Use mouse to define box for map by clicking and dragging mouse.



Step 3.

Select 'Scale&DoMap' option from **Map** menu. Map is scaled, centered and displayed as 'unclassed' map (sixteen shadings).

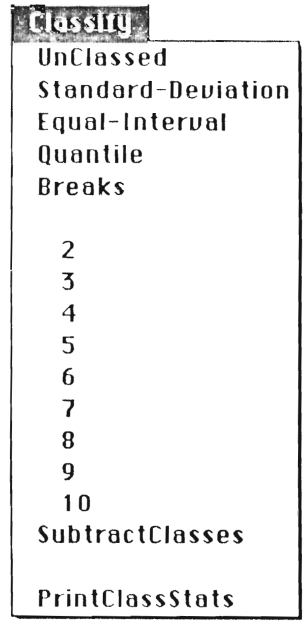




Result: Unclassed map with up to 16 shadings.

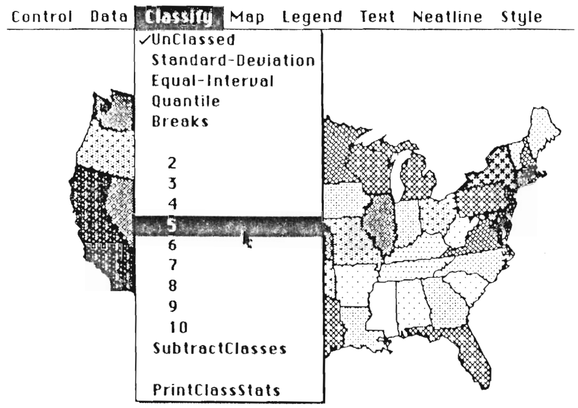
Image Mapping Systems, Box 31593, Omaha, NE 68132

The 2 steps to re-classify the map



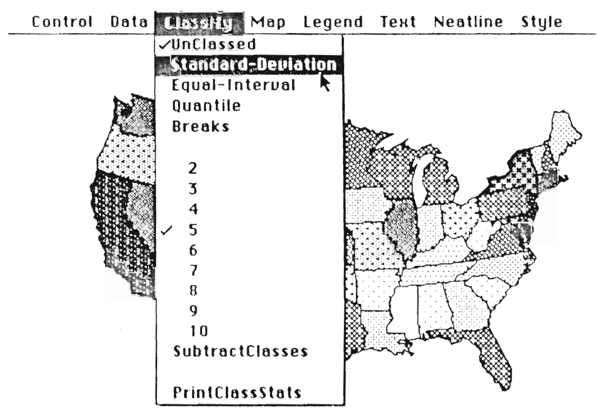
Step 1.

Select number of classes from '**Classify**' menu.



Step 2.

Select classification method from '**Classify**' menu. Map is automatically classified and displayed. Time to classify and display map: 3 seconds.



Result:
Five-class standard-deviation classified map.

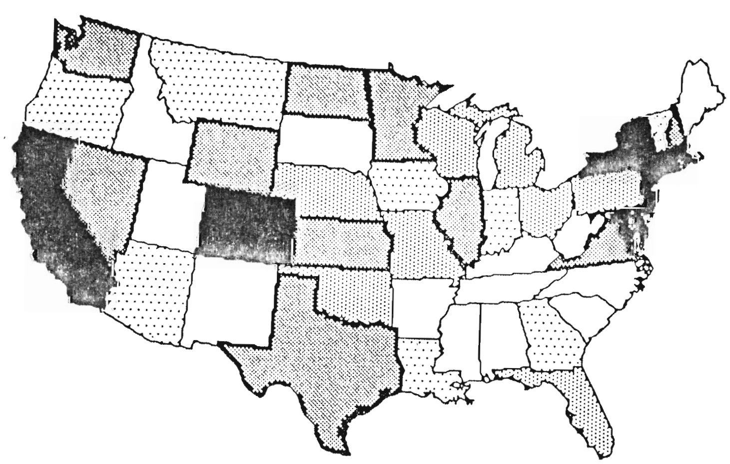
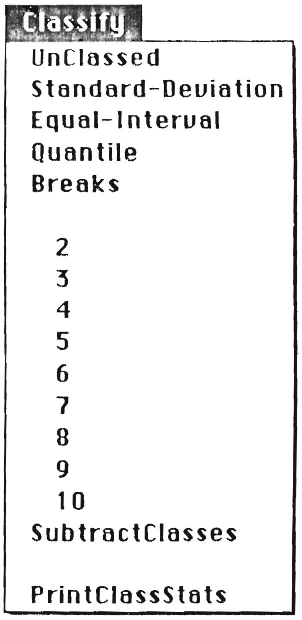


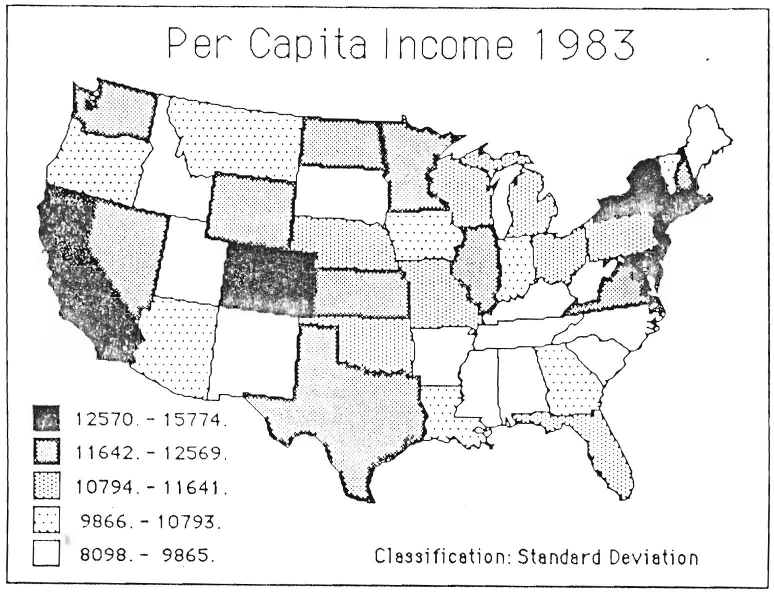
Image Mapping Systems, Box 31593, Omaha, NE 68132



Classification Methods:

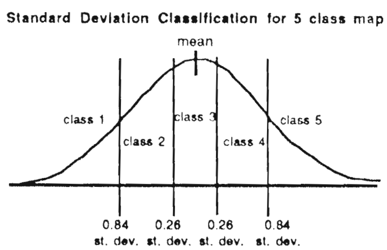
MacChoro includes 4 common methods of data classification plus unclassed:

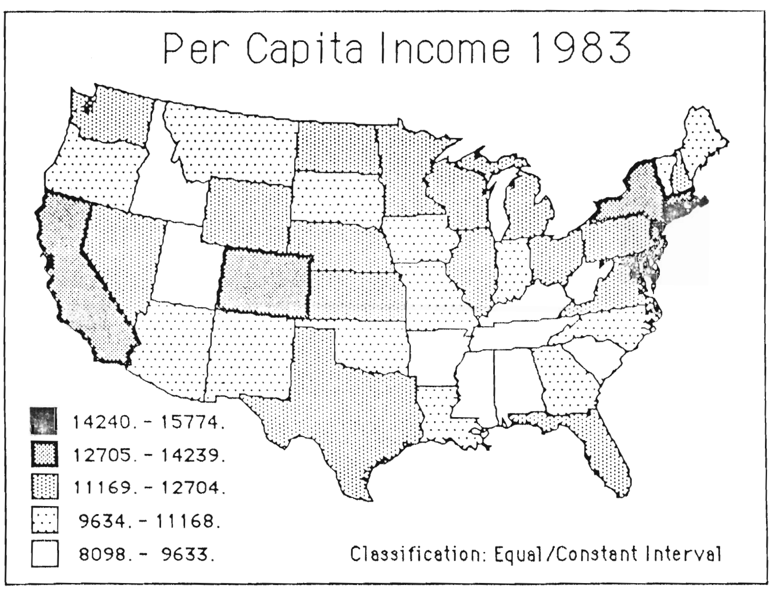
- standard deviation
- equal-interval (constant interval)
- quantile
- natural breaks (largest differences)
- •• unclassed



Standard Deviation

Classification is based on the standard deviation and mean such that normal curve is divided into equal segments. Example:





Equal Interval / Constant Interval

Classification based on equal divisions in the range of data. Example:

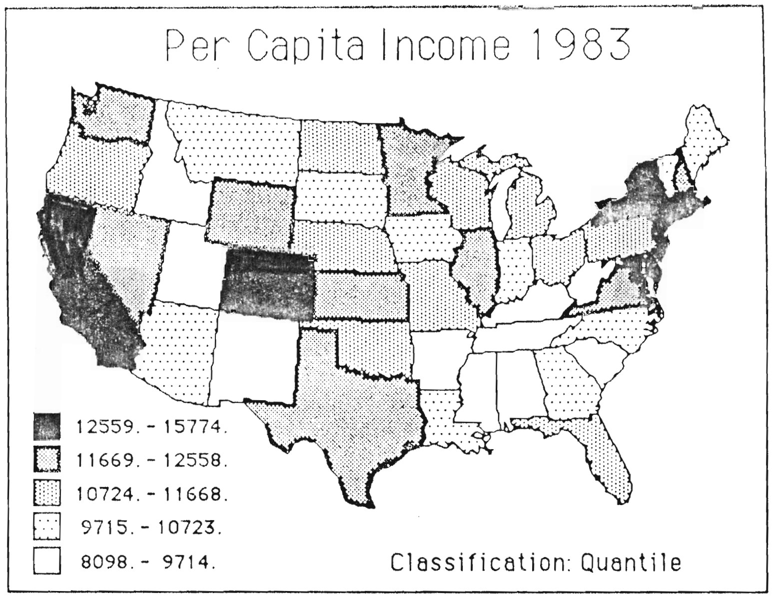
\$15,774 - \$8,098 = \$7,676 (range)

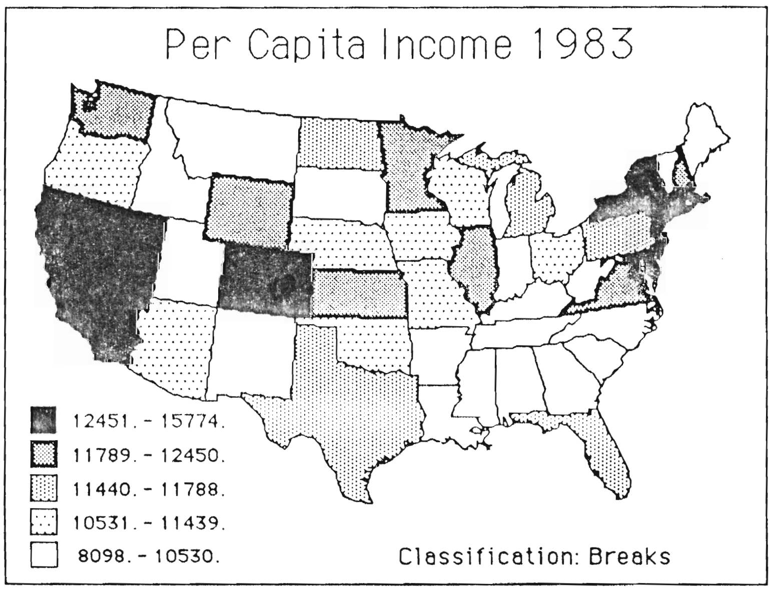
\$7,676 / 5 classes = \$1,535.2 (interval)

Image Mapping Systems, Box 31593, Omaha, NE 68132

Quantile

Equal number of observations in each class. Example:
49 states / 5 classes = 9.8 states per class



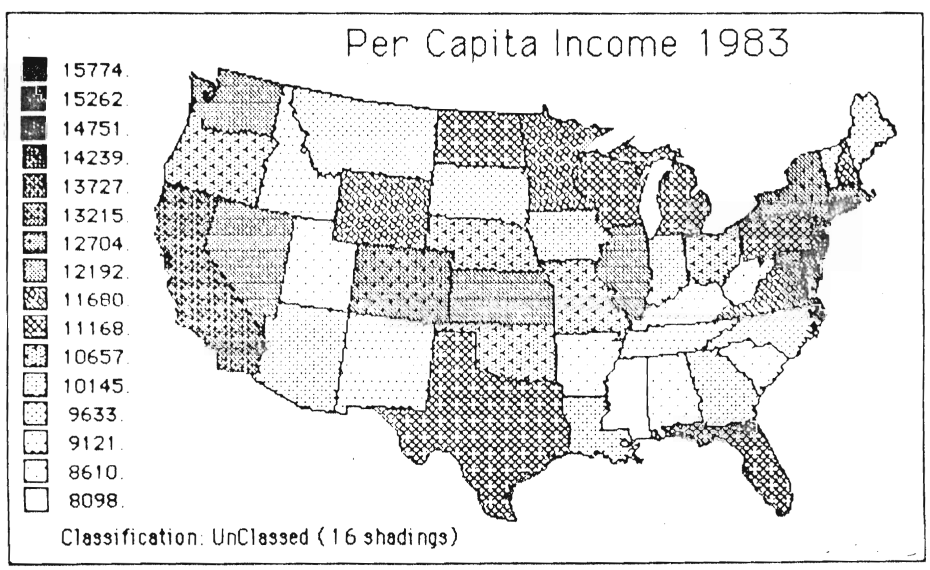


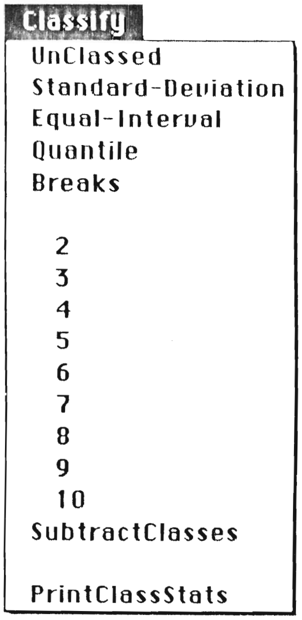
Breaks

Largest differences between data values is used for classification. For a 5 class map, class-breaks are half-way between four largest differences.

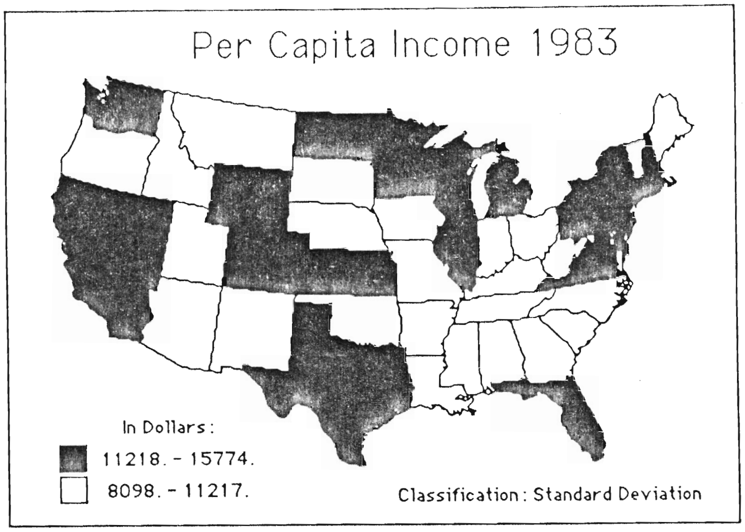
Unclassed

Shading is proportional to the data-value. (Present limit: 16 shadings).

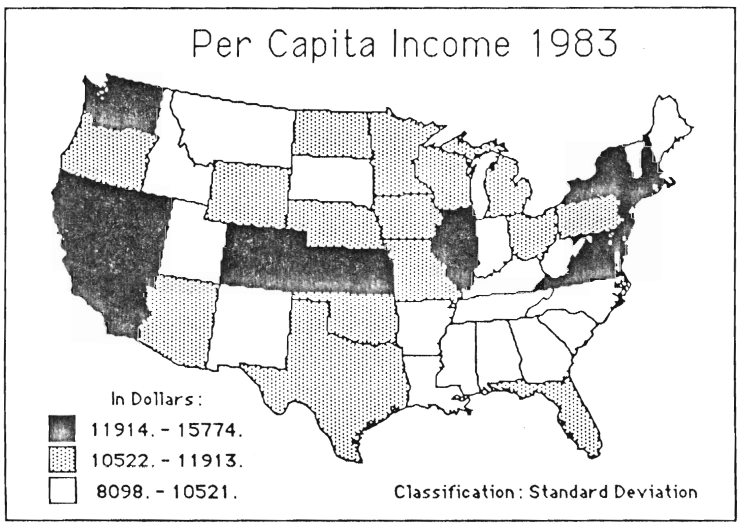




2 classes



3 classes

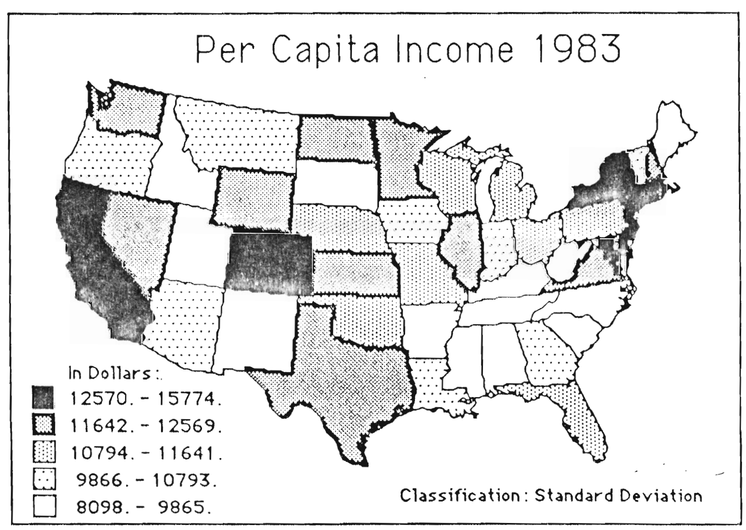


Number of Classes

(2 - 16)

By viewing the data with different numbers of classes, one not only obtains a greater understanding of the data but one can better select the number of classes that is most representative of the distribution.

5 classes



Once you have chosen your classification method, the number of classes can be changed:

- for 2-10 classes by selecting the number from the 'Classify' menu
- for 11-16 classes by selecting
  1) 'UnClassed'
  2) the classification method
  3) 'SubtractClasses' option

Data Manipulation:

MacChoro incorporates extensive data manipulation capabilities. Data may be modified through division, multiplication, subtraction or addition by another data set or by a constant. Logarithm or square root transformation is also possible. The data may be interactively viewed and edited and any modifications may be saved in the same or a different file. A data file can be initially created with **MacChoro** or may be extracted from a spreadsheet program such as Multiplan, Excel or Jazz.

Data

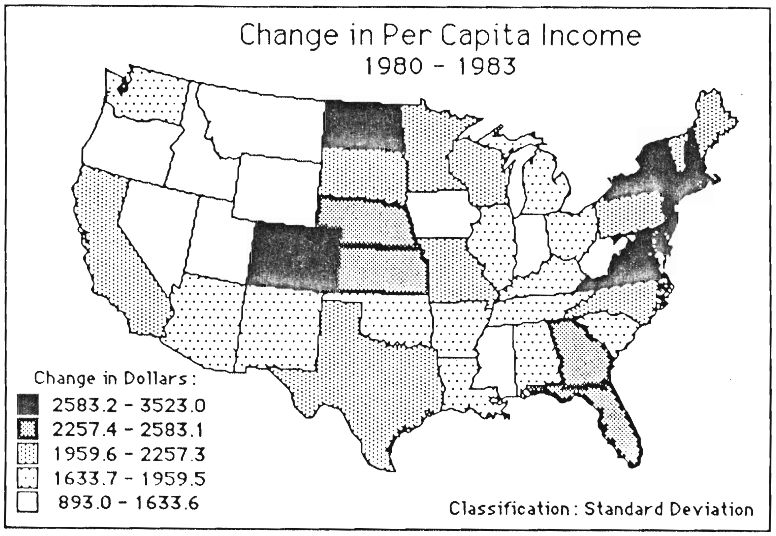
DefineDataFile
ReadDataFile
RewindDataFile

DivideData
MultiplyData
SubtractData
AddData

DivideByConstant
MultiplyByConstant
SubtractConstant
AddConstant

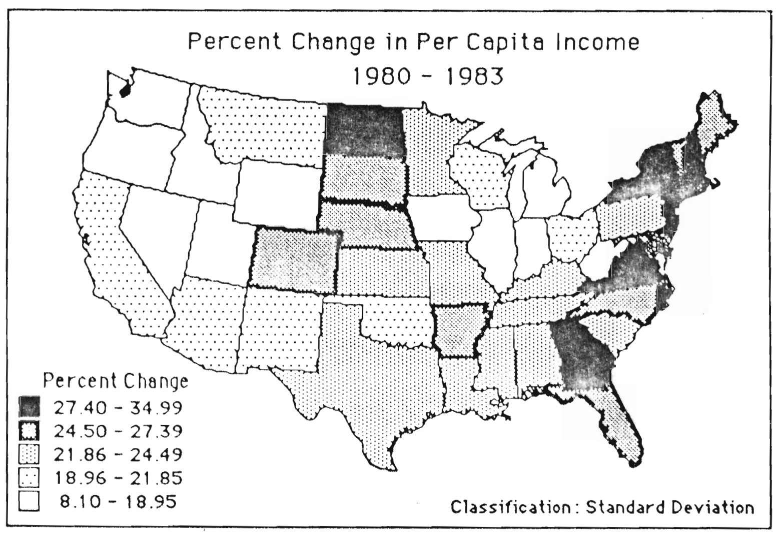
LogBase10
Square Root

ViewData
EditData
SaveData



Per Capita Income 1983 - Per Capita Income 1980 =
Change in Per Capita Income 1980 - 1983

(Change in Per Capita Income 1980-1983 /
Per Capita Income 1980) X 100. =
Per Cent Change in Per Capita Income 1980 - 1983



Note: Change in Per Capita Income reflects both changes in income and changes in population.

Data

- DefineDataFile
- ReadDataFile
- RewindDataFile

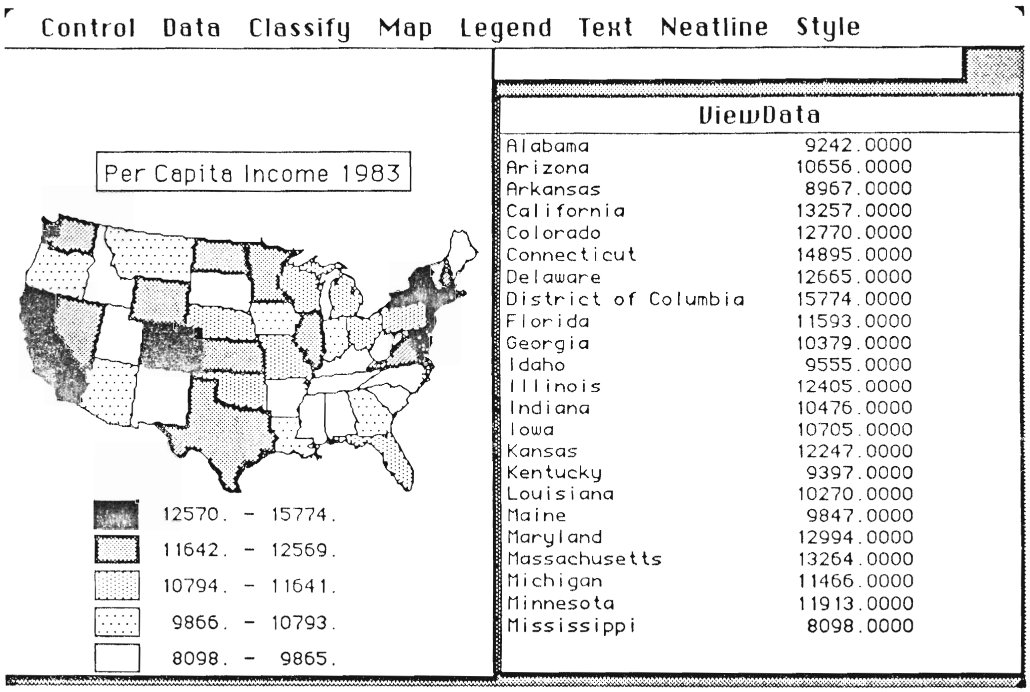
- DivideData
- MultiplyData
- SubtractData
- AddData

- DivideByConstant
- MultiplyByConstant
- SubtractConstant
- AddConstant

- LogBase10
- Square Root

- **ViewData**
- EditData
- SaveData

View Data and Map Simultaneously



ViewData	
Alabama	9242.0000
Arizona	10656.0000
Arkansas	8967.0000
California	13257.0000
Colorado	12770.0000
Connecticut	14895.0000
Delaware	12665.0000
District of Columbia	15774.0000
Florida	11593.0000
Georgia	10379.0000
Idaho	9555.0000
Illinois	12405.0000
Indiana	10476.0000
Iowa	10705.0000
Kansas	12247.0000
Kentucky	9397.0000
Louisiana	10270.0000
Maine	9847.0000
Maryland	12994.0000
Massachusetts	13264.0000
Michigan	11466.0000
Minnesota	11913.0000
Mississippi	8098.0000

Legend

- DefineLegendBox
- DoBoxLegend
- DoRectLegend
  - WiderRect
  - TallerRect

- BoxesApart
- BoxesTogether

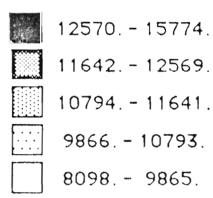
- GapsSmaller
- GapsLarger

- MoveBreaksIn
- MoveBreaksOut

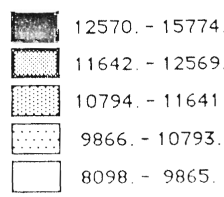
- OutlineLegend
- MoveLegend
- EraseLegend

Legend Options

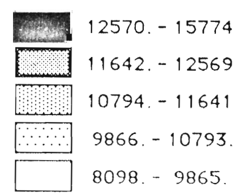
Options under 'Legend' menu offer extensive control over design of legend.



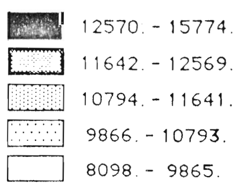
DoBoxLegend



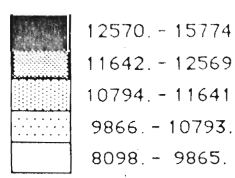
DoRectLegend



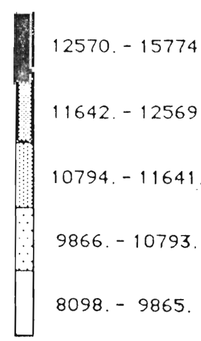
WiderRect



GapsLarger



BoxesTogether



TallerRect

Image Mapping Systems, Box 31593, Omaha, NE 68132

Welcome to the World of

Microsoft FORTRAN

for the Apple Macintosh!

The Microsoft FORTRAN Compiler for the Apple Macintosh is a full implementation of FORTRAN 77. It has all the language features of mainframe FORTRAN, plus a powerful interface to the Apple Macintosh's operating system and graphics capabilities.

1.1 FORTRAN 77

The FORTRAN programming language, a contraction of the words FORmula TRANslation, is a computational problem solving language. Because it resembles familiar arithmetical language, it greatly simplifies the preparation of problems for machine computation. Data and instructions are organized in a sequence of FORTRAN statements. This sequence of statements is referred to as the source program. A program written in the FORTRAN language can be processed on any machine which has a FORTRAN compiler with little or no modifications to the source program. In this sense, the FORTRAN language is said to be machine independent.

Microsoft Corporation's FORTRAN 77 is a complete implementation of the 1977 ANSI version of the FORTRAN language. In addition, many useful extensions have been added including many of those which will be incorporated in the next standard published by ANSI. The compiler operates in three passes, with each pass consisting of a separate overlay, allowing it to operate in a minimum amount of memory.

Microsoft FORTRAN provides all of the file I/O facilities required by the ANSI standard resulting in the capability of a FORTRAN program to execute with little or no knowledge of the file I/O conventions of a particular operating system.

In addition, standard implicit input and output units are provided allowing a FORTRAN 77 program to input from a data file and output to a line printer without opening, closing, or naming a file. Even the actual unit number that the file is internally connected to never needs to be referenced. The following program will copy its input to its output on standard conforming implementations of FORTRAN:

```
      PROGRAM COPY
      CHARACTER*80 TEXT
1     READ (*,10,END=2) TEXT
      PRINT *,TEXT
      GOTO 1
2     STOP
10    FORMAT (A80)
      END
```

using the structured facilities of Microsoft FORTRAN 77, the same program could be written as:

```
      program copy

      character*80 text
      integer eof

        do
          read (*,'(a)',iostat=eof) text
          if (eof) exit
          print *,text
        repeat

      end
```

All the Tools You Need

What sets the Macintosh apart from other personal computers is its revolutionary *user interface*. In plain English, the word *interface* means a junction or boundary where two things meet. In computerese, it refers to the set of rules and conventions by which one part of an organized system (like the Macintosh) communicates with another. Wherever two components of the system come together, they exchange information by way of an interface.

The Macintosh system consists of hardware (physical components such as chips, circuits, and other electronic and mechanical devices) and of software (programs). The most important component of all is the human being peering at the screen and fiddling with the mouse and keyboard. This flesh-and-blood component of the system is known, in technical parlance, as the *user*. So the user interface is the set of conventions that allow the human user to communicate with the rest of the system.

In the past, user interfaces were typically based on a screen full of text characters (usually displayed in garish green) and a keyboard for typing those characters. To tell the computer what to do, you had to memorize a complex command language, so you could press exactly the right keys in exactly the right order. If your actions didn't conform to what the computer expected of you, it would tell you so in terms ranging from curt to unintelligible. On the whole, it was sometimes hard to tell that the human was the boss and the computer, the servant.

The Macintosh changes all that. In place of the time-honored character screen and keyboard, it uses a high-resolution, "bit-mapped" display and a hand-held pointing device, called a mouse. This results in a whole new way of communicating between people and computers. The bit-mapped screen can present information in vivid visual form, using pictorial "icons," elaborate graphical effects, and varied patterns and textures. Text can be depicted exactly as it will appear on the printed page—in black characters on a white background, with a variety of typefaces, sizes, and styles. The mouse provides a direct, natural way of giving commands. This is done by using the mouse to point and manipulate images directly on the screen instead of typing arcane command sequences using the keyboard. The programmers at Apple have put a great deal of thought and effort into taking advantage of these features to produce a user interface that feels natural and comfortable. The result of their efforts is the User Interface Toolbox, 64 kilobytes of tightly engineered, hand-crafted machine-language code that's built into every Macintosh in *read-only memory* (*ROM*). With it, you can write programs that use overlapping windows, pull-down menus, scroll bars, dialog boxes, and all the other wonders you see on the Macintosh screen. This book will teach you how.

Strictly speaking, the Macintosh ROM is divided into three parts: the Macintosh Operating System, which handles low-level tasks such as memory management, disk input/output, and serial communications; the QuickDraw graphics routines, which are responsible for everything displayed on the screen; and the User Interface Toolbox, which implements the higher-level constructs of the user interface, such as windows and menus. As a rule, we'll be using the term "Toolbox" to refer loosely to the entire body of built-in code that's available to a running program; only occasionally will we use it in the narrower sense of the user-interface code alone, to distinguish it from the Operating System and QuickDraw.

14.

Chernicoff, Stephen. Macintosh Revealed: Unlocking the Toolbox, Vol. 1. Hayden Publishing, 1985.