

MICROCOPY RESOLUTION TEST CHART  
NATIONAL BUREAU OF STANDARDS-1963-A

2

AD-A170 612

NPS52-86-017

# NAVAL POSTGRADUATE SCHOOL

Monterey, California



DTIC  
ELECTE  
A 04 1986  
B

User Interface Design for Two-Dimensional  
Polygonally Encoded Geological Survey Maps †

Joann M. Ammann

Robert B. McGhee

Michael J. Zyda

July 1986

Approved for public release; distribution unlimited

Prepared for:

Chief of Naval Research  
Arlington, VA 22217

MMF FILE COPY

NAVAL POSTGRADUATE SCHOOL  
Monterey, California

Rear Admiral R. H. Shumaker  
Superintendent

D. A. Schrady  
Provost

The work reported herein was supported in part by the Foundation Research Program of the Naval Postgraduate School with funds provided by the Chief of Naval Research.

Reproduction of all or part of this report is authorized.

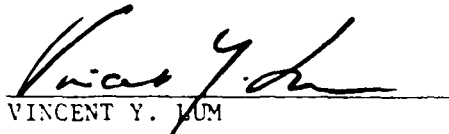
This report was prepared by:



MICHAEL J. ZYDA  
Assistant Professor  
Computer Science Department

Reviewed by:

Released by:



VINCENT Y. LUM  
Chairman  
Department of Computer Science



KNEALE T. MARSHALL  
Dean of Information and  
Policy Science

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NPS52-86-017	2. GOVT ACCESSION NO. <b>AD A170612</b>	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) User Interface Design for Two-Dimensional Polygonally Encoded Geological Survey Maps †		5. TYPE OF REPORT & PERIOD COVERED
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Joann M. Ammann Robert B. McGhee Michael J. Zyda		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93943-5100		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61152N; RR000-01 N0001486WR4E001
11. CONTROLLING OFFICE NAME AND ADDRESS Chief of Naval Research Arlington, VA 22217		12. REPORT DATE July 1986
		13. NUMBER OF PAGES 66
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report)
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report)  Approved for public release; distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)  cartography, user interface design		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)  We present in this study an overview of a cartographic processing pipeline for the generation and maintenance of polygonally encoded data bases from published U. S. Geological Survey maps. The focus of this research centers on the development of an interactive editing system. The editor, serving as the final step in the overall project, provides the user with the capability to correct and modify dated topographic characteristics. A variety of processing and digitizer induced errors introduced into the data base from previous utility		

REPRODUCED AT GOVERNMENT EXPENSE

DD FORM 1473  
1 JAN 73

EDITION OF 1 NOV 69 IS OBSOLETE

S. N. 0102-LE 014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)



## User Interface Design for Two-Dimensional Polygonally Encoded Geological Survey Maps ‡

*Joann M. Ammann, Robert B. McGhee and Michael J. Zyda \**

Naval Postgraduate School,  
Code 52, Dept. of Computer Science,  
Monterey, California 93943

### ABSTRACT

We present in this study an overview of a cartographic processing pipeline for the generation and maintenance of polygonally encoded data bases from published U.S. Geological Survey maps. The focus of this research centers on the development of an interactive editing system. The editor, serving as the final step in the overall project, provides the user with the capability to correct and modify dated topographic characteristics. A variety of processing and digitizer induced errors introduced into the data base from previous utility steps can also be corrected. Included is a discussion on the internal indexing scheme used for managing revisions and the techniques and algorithms for updating the data bases.

Categories and Subject Descriptors: 1.3.6 [Methodology and Techniques]: human interface design, interactive graphics dialogues, user interface; 1.3.m [Miscellaneous]: cartography;

General Terms: Techniques;

Additional Key Words and Phrases: cartography, user interface design;

---

‡ This work has been supported by the Naval Postgraduate School Foundation Research Program and a grant from the Naval Ocean Systems Center, San Diego (Ref. # N0001486WR4B123AC).

\* Contact author.

## TABLE OF CONTENTS

I. INTRODUCTION .....	7
A. DATA ENTRY .....	7
B. MOTIVATION .....	9
C. SCOPE OF THE STUDY .....	10
D. ORGANIZATION .....	10
II. A CARTOGRAPHIC DATA PROCESSING SYSTEM .....	12
A. OVERVIEW .....	12
B. PRE-PRINTED MAP SELECTION .....	12
C. DIGITIZATION .....	14
D. IMAGE PROCESSING .....	15
1. General .....	15
2. The Diehl Feature Extraction System .....	16
E. DATA COMPRESSION .....	18
F. THE EDITOR .....	19
III. RESEARCH FACILITIES .....	20
A. IRIS WORKSTATION .....	20
B. IRIS GRAPHICS LIBRARY .....	20
IV. THE EDITING SYSTEM .....	23



A. GETTING STARTED .....	23
B. SCREEN LAYOUT .....	23
C. INPUT DEVICES .....	25
D. MENU STRUCTURE .....	25
1. Area Edit .....	27
2. Point Edit .....	30
3. Change View .....	30
4. Read in a File .....	31
5. Write to a File .....	31
6. Exit System .....	32
E. SYSTEM PROMPTS .....	32
F. FEATURES SUPPORTED .....	33
G. FILE FORMAT .....	34
V. MANAGING USER REVISIONS .....	36
A. INTERNAL INDEXING SCHEME .....	36
B. UPDATING THE DATABASE .....	37
1. Update Database .....	39
2. Data Compression .....	39
3. Expand Area .....	42
4. Subset Area .....	42

VI CONCLUSIONS AND RECOMMENDATIONS .....	45
A. CONCLUSIONS .....	45
B. LIMITATIONS AND RECOMMENDED EXTENSIONS .....	46
APPENDIX A - SAMPLE EDITING SESSION .....	49
APPENDIX B - USGS TOPOGRAPHIC MAP SYMBOLS .....	53
APPENDIX C - UPDATE DATABASE ALGORITHMS .....	60
LIST OF REFERENCES .....	66
BIBLIOGRAPHY .....	67
INITIAL DISTRIBUTION LIST .....	68

## I. INTRODUCTION

Since the computer's inception, the cartographic industry has profited from repeated hardware and software advances. Computer-aided mapping systems effectively reduce labor costs associated with the production and maintenance of cartographic documents. Orbiting satellites widen land coverage to encompass the entire globe. Significantly improved remote-sensing systems extend the cartographer's field of sight beyond the visible spectrum. Yet despite such progress, the process of cartographic capture and encoding remains a continuing concern.

### A. DATA ENTRY

Current mapping systems lack the capability to automatically enter and classify data from the source. Most employ electromechanical digitizers with which human operators manually trace lineations and boundaries, recording (X,Y) coordinates at defined intervals or pivot points. Collection of information in this format is commonly referred to as vector mode storage. Numeric keys, located on hand-held devices, assist feature labeling with specified category codes. The above procedure is often a time-consuming, fatiguing and error prone operation.

Alternatively, manufacturers offer automatic digitizing facilities promoting raster mode storage. In raster mode, images are divided into scan lines and

further subdivided into small picture cells. Information is stored in a sequential fashion by horizontal scan line pattern arranged from the top of a picture downwards. If retention of additional area characteristics is desired, manual intervention, tiring by nature, is necessary to separate and identify individual components.

The market also supports automatic line-following digitizers capable of sensing and tracking linear features with servomechanisms. Aimed at alleviating the attentive and tedious handling common to other methods, these systems have limited use and require high quality maps as input. Uncertainties arising at intersecting features are resolved through the assistance of an operator.

The solution to automatic data input and feature identification from paper maps presents a most difficult challenge. The production of a fully automated system may possibly never be attained. Present day manufacturers have already acceptably confined their systems' capabilities to only certain sub-tasks. Two major reasons account for the limited progress seen in the area of data capture:

- 1. the market is characteristically small and undercapitalized, offering little growth potential.
- 2. hardware facilities have lacked sufficient computing power and display resolution capabilities to handle the demanding load of processing and manipulating cartographic data.

## B. MOTIVATION

Many government mapping agencies and commercial map companies have realized the potential benefits of maintaining digital cartographic data bases. The U. S. Geological Survey, for example, is in the process of converting all of its United States quadrangle series maps into digital form. Topographic data is being accumulated into separate categories such as digital elevation, surface hydrography, public land-survey network, geographic names and other classes. Specialized projects, requiring digitized map files, are newly emerging. For instance, work conducted by Richbourg, Rowe and Zyda [Ref. 1] on solution techniques for two-dimensional route planning problems for mobile autonomous vehicles identifies a need for topographic information grouped by terrain speed regions.

The inability of current data bases and mapping systems to fulfill the latter example's requirements served as a catalytic force in reopening the study for improving computer-assisted cartographic entry and encoding techniques. A discussion of the processing system under development is presented in Chapter Two. The major goal of the research is to offer an alternative approach from the popular interactive digitizing methods used in handling data entry from published maps. Although it is difficult to design an accurate, reliable and low cost scheme, the proposed approach seems promising.

## C. SCOPE OF THE STUDY

This study develops and examines an interactive editing system for polygonally encoded data bases generated from the cartographic processing system described in [Ref. 2]. The editor, serving as the final step in the overall project, provides the user with the capability to update and modify topographic information. A variety of common processing and digitizer induced errors, including miscoded coordinates, line noise and superfluous points can also be corrected. The editor's algorithms were implemented using the C programming language with calls made to available Graphics Library routines. A Silicon Graphics, Inc. IRIS (Integrated Raster Imaging System) Turbo 2400 workstation was chosen for this study as it supports the high resolution and fast transmission rates needed for manipulating cartographic data.

## D. ORGANIZATION

Chapter Two reviews the complete cartographic data processing pipeline under development at the Naval Postgraduate School. Each step within the overall package is briefly discussed. Chapter Three reports on the available hardware and software research facilities utilized. Chapter Four introduces a polygonal editing system, the topic of this study. Chapter Five addresses the techniques employed for managing user revisions. The algorithms used for transferring an updated map image from the screen into stored data are also discussed. Finally Chapter Six is an examination of the presented interface's

limitations and weaknesses with suggestions for future enhancements and improvements. Attached as appendices are before and after *snap shots* taken from sample editing sessions, a list of USGS map symbols supported by the editor and the algorithms utilized for updating a cartographic data base.

## II. A CARTOGRAPHIC DATA PROCESSING SYSTEM

### A. OVERVIEW

The integrated cartographic processing package, developed at the Naval Postgraduate School, is organized as a set of five stages, each performing a distinct function. A collection of methods and techniques, well-known in some cases and novel in others, assist in making this prototypical model operational. The component steps are shown schematically in Figure 2.1 in the order they are generally executed. Two of the stages (items 1 and 2) deal primarily with the selection and preparation of maps for use in subsequent steps; two (items 3 and 4) handle actual digital information processing and the creation of a computerized data base, and the remaining one (item 5) conducts revision and editing of files stored on disk. Steps three and four are the subject of research conducted by Dodd (Ref. 2), while step five is the focus of this study. Each stage of the project is briefly described below.

### B. PRE-PRINTED MAP SELECTION

prior to implementing any cartographic information processing system, selection of a standardized mapping scheme is necessary. For this research, primary input consisted of sectional pieces from U. S. Geological Survey (USGS) Quadrangle (7.5-minute x 7.5-minute) series maps with conventional units at a



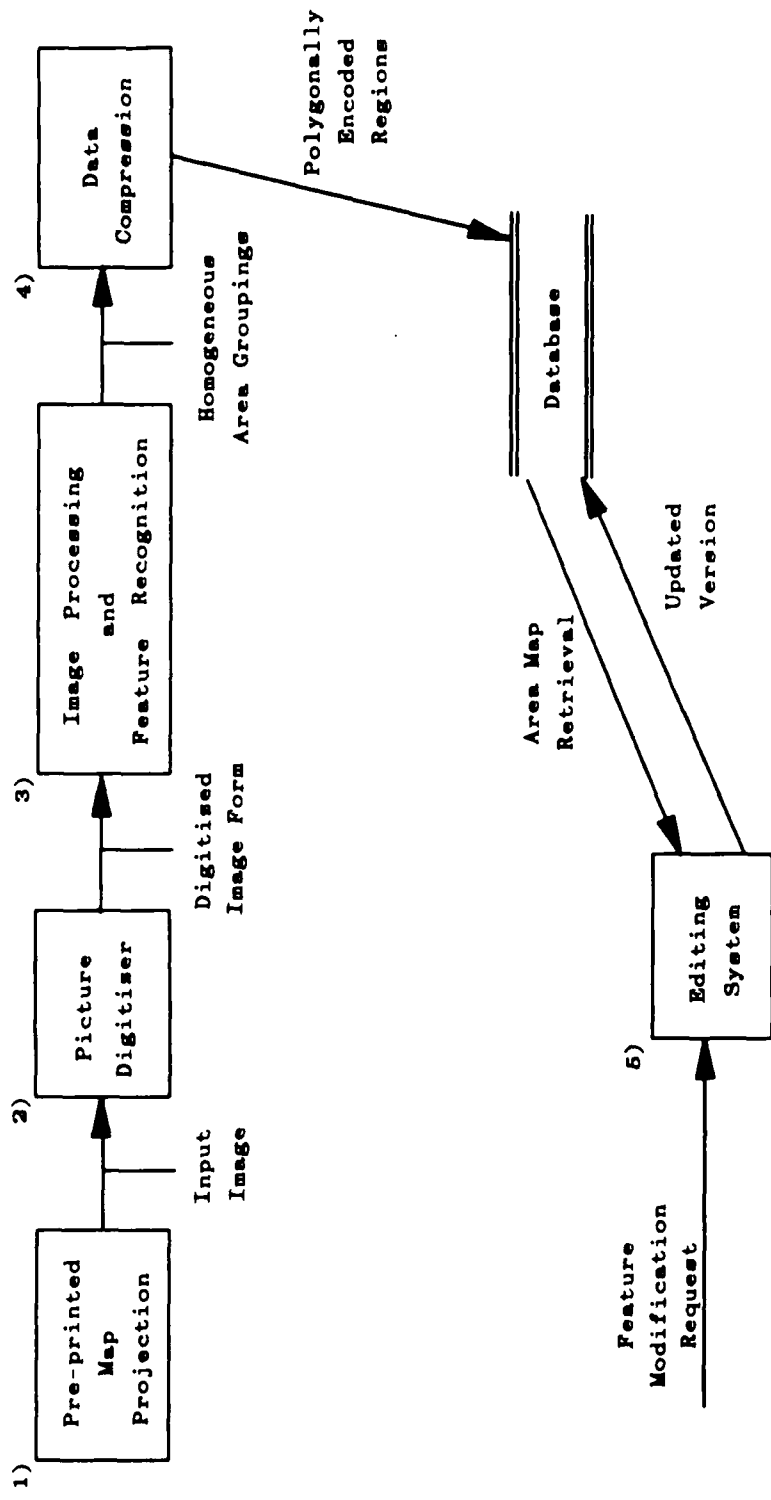


Figure 2.1 Processing Pipeline for Cartographic Information Capture and Revision

1:24,000 scale. Widely available and accepted, each USGS map complies with established specifications in regards to size, scale, symbolism and content [Refs. 3 and 4]. The use of color assists in discriminating the various cartographic features (see Table 2.1 below).

**TABLE 2.1  
USGS MAP COLOR USAGE SUMMARY**

Color	Application Area
BLACK	Cultural features such as roads and buildings.
BLUE	Hydrographic features such as lakes, rivers, wetlands and reservoirs.
BROWN	Hypsographic features shown by contour lines such as slopes and elevations.
GREEN	Surface cover including woodlands, scrub, orchards and vineyards.
PURPLE	Features added from aerial photographs during map revision. The changes are not field checked.
RED	Important roads and public land survey system.
WHITE	Nonvegetative features such as barren waste areas and piles.

### C. DIGITIZATION

The raw pictorial representation of the paper map is converted into computer readable form for subsequent processing through the use of a video digitizer camera. Digitization comprises the scanning of a document to resolve the character information into small picture elements, or pixels. The optical

response at each pixel controls detection circuitry for the generation of positionally defined digital output.

An EYECOM<sup>1</sup> Picture Digitizer and Display unit interfaced to a Digital Equipment Corporation (DEC) PDP-11 computer was utilized for this project. Additional auxiliary lighting and filters (red, green and blue) complement the setup. Files produced after digitization are downloaded onto a DEC VAX 11/780 computer. The File Transfer Protocol (FTP) program on the connecting ETHERNET allows file movement to the final destination host, a Silicon Graphics, Inc. IRIS Turbo 2400 workstation.

#### D. IMAGE PROCESSING

##### 1. General

After transforming a map's contents into digital form, ideally one applies an image processing system which promotes automatic component recognition. A component is any feature that is judged to be significant at a chosen scale and that exists within an area. As discussed above, present day systems fall short in reaching this *optimum* capability. Most if not all commercially available systems require varying degrees of human intervention to demarcate cartographic information. Although the market supports some automatic line-following systems, these tools are useful only for certain types of features and only on maps with high quality line work. In general, human assistance is needed to bridge

---

<sup>1</sup> A trademark of Spatial Data Systems, Inc.

interruptions, to resolve uncertainties at intersections marked by unreliable directional trends and to affix identifying codes for individual features.

A technique proposed by Diehl [Ref. 2] attempts to resolve this noted deficiency. Although in a prototypical state, his method offers an alternative avenue for further research exploration in solving this most difficult problem. An overview of his research follows.

## 2 The Diehl Feature Extraction System

From a black box viewpoint (see Figure 2.2), Diehl's system analyzes individual pixels of an input video image and through heuristic reasoning, categorizes each pixel into a subset of colors synonymous to those comprising the initial paper map. In the digitization process, each pixel has the potential of adopting one of over sixteen million color identities<sup>2</sup> under the RGB color system or 24 bits per pixel, with 8 bits each for red, blue and green intensity values. Determination of actual pixel color assignment is derived by sample readings taken over small contiguous map areas. This averaging procedure accounts for color variations observed in pixels of seemingly uniform regions. Points lying in the proximity to boundaries separating dissimilar colors often assume *fuzzy* values while interior ones approximate original map colors to within a measurable deviation. Preliminary tests show Diehl's work to be highly accurate in correctly assigning pixels of internal feature sections with small error

---

<sup>2</sup> A color in the spectrum can be formed by blending the additive primitives (red, green and blue).

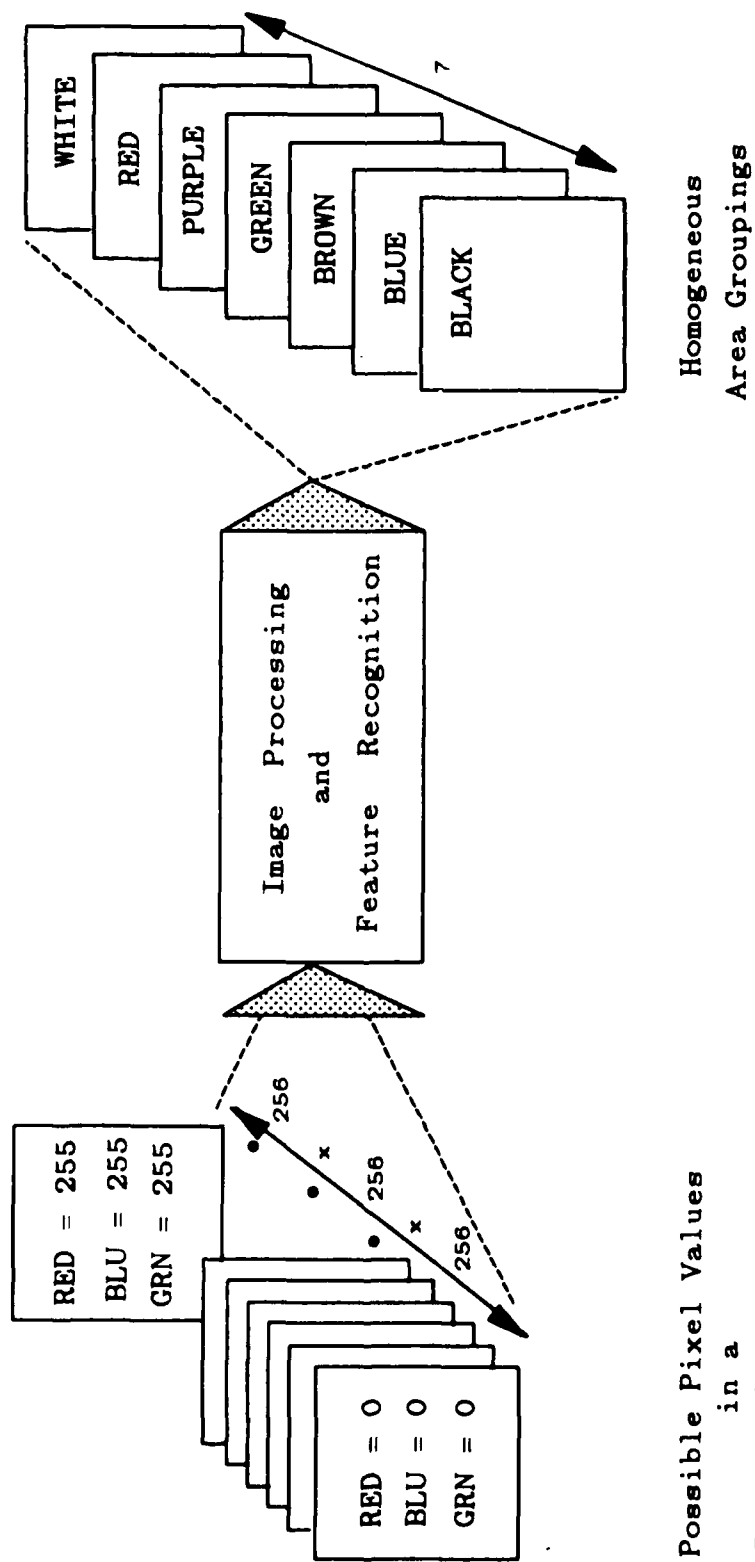


Figure 2.2 Conversion from video image to homogeneous area regions [Ref 2]

rates occurring near obscure border lines. The resulting data base generated from this processing scheme can either be retained as is or be funneled through the Data Compression Stage to optimize storage space.

#### E. DATA COMPRESSION

Once the map has been segmented into one of seven equivalence classes, an appropriate storage format (raster versus vector) must be chosen. Several factors were considered during the selection process :

- speed tolerance for terminal redisplay and screen refresh,
- degree of resolution needed to maintain accuracy,
- retrieval rate for information of interest,
- ease in data manipulation during editing sessions,
- storage resource requirements,
- quantity of information to be retained per unit of measure, and
- applicability of format to problem area.

Since our interest centers on the generation of data bases for use in autonomous vehicle route planning, vector storage proved to be most economical.

A triangular tessellation, formulated by Diehl [Ref. 2], serves as the fundamental data storage mechanism. In general, the process consists of two repeatedly called parts :

- identification of the next homogeneous area, and
- division of the identified region into polygons.

A detailed discussion of the algorithms involved is presented in Chapter Five. These algorithms were slightly modified and incorporated into the editing system's facility for updating a cartographic data base.

#### F. THE EDITOR

Finally, an editing system is introduced to ensure proper maintenance of the data base generated from the previous utility steps. Typically most printed maps on the market are several years old and are rarely updated. A computer-assisted map data base editor allows easy modification and correction of dated topographic characteristics. The digital map data can be altered either to eliminate or remedy errors and blunders or to revise established files when new updates arrive. Often proved to be time consuming for the user, this is an essential step in the overall process of assuring a map's usefulness and accuracy. Possible adverse effects could result based on outdated or erroneous information portrayed by an unkept data base. The contents of the remaining chapters focus on the editing system's design.

### III. RESEARCH FACILITIES

#### A. IRIS WORKSTATION

The IRIS Turbo 2400 Graphics Workstation, manufactured by Silicon Graphics, Inc., served as the primary tool for this research. By incorporating custom built VLSI circuits into its design, the IRIS offers an attractive alternative to the more conventional workstations [Ref. 5]. The special-purpose hardware, designed to replace less efficient software, yields high processing speeds and increased performance reliability necessary for manipulating cartographic data. The system notably combines real-time color graphics with Unix operating system software and Ethernet network communications. A very high resolution color monitor provides crisp displays and fine delineations for even the most complex work. Figure 3.1 outlines the distinctive features of the system.

#### B. IRIS GRAPHICS LIBRARY

The IRIS Graphics Library is an extensive collection of utility and graphics subroutines providing high level access to the hardware enabling graphical objects to be easily manipulated as geometrical objects (points, lines, polygons, etc) rather than pixels. A series of coordinate systems and mapping instructions provides the user with the capability to define such objects in world space.



A subset of the Library's routines were used for this research. They can be generalized into the following nine categories :

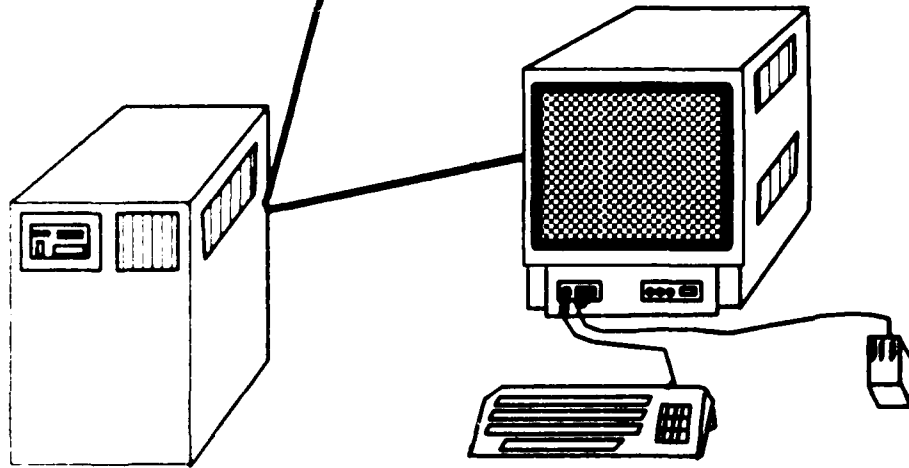
- *Global state* commands initialize the hardware and control global state variables.
- *Primitive drawing* commands draw points, lines, polygons, circles, arcs, and text strings on the screen.
- *Drawing attribute* commands select characteristics for drawing lines, filling polygons, and writing text strings.
- *Coordinate transformation* commands perform manipulations on coordinate systems, including mapping user-defined coordinate systems to screen coordinate systems.
- *Display mode and color* commands determine how the bitplane image memory is used and how objects are colored on the screen.
- *Input/output* commands initialize and read input/output devices.
- *Object creation and editing* commands provide the means to create hierarchical structure of graphics commands.
- *Picking and selecting* commands identify the commands that draw to a specified area of the screen.
- *Textport* commands allocate an area of the screen for writing text.<sup>5</sup>

A detailed account on command usage for each of the above classes is covered in the IRIS User's Guide.

---

<sup>5</sup>Silicon Graphics Inc., *IRIS User's Guide Version 2.1. Page 1-2.*

Ethernet connection between IRIS's and VAX's



- 32-bit Motorola 68020 Processor
- 1024 X 768 X 32-bit Display Memory
- 2 MB of CPU Memory
- 144 MB of Disk Storage
- Floating Point Accelerator
- Geometry Pipeline with Geometry Engines and Geometry Accelerators
- 60 Hz Non-interlaced Display
- 18-bit Z-buffer for Hidden Surface Elimination
- Hardware Smooth Shading
- Unix System V
- IRIS Graphics Library
- Cartridge Tape Unit
- Ethernet to VAX's
- Digitizer Tablet

Features of the IRIS Turbo 2400 Graphics Workstation

Figure 3.1

## IV. THE EDITING SYSTEM

### A. GETTING STARTED

The editing system is invoked by a single command to the IRIS workstation :

**% MAPEDIT**

Upon execution, an initial title page is displayed for a few seconds prior to the appearance of the main program. Prompts and instructions guide the user through the entire operation of the editor.

### B. SCREEN LAYOUT

The screen is partitioned into five non-overlapping windows (see Figure 4.1). Each is reserved for a particular use. Window One serves as the primary work bench for area viewing and editing of a sectional map. Window Two exhibits a dynamic picture display of the current map image in miniature form. Window Three contains any menu options available during a particular point in time. The use of *pop up menus*, described below, assists in reducing screen clutter and user distraction. Window Four is restricted to point viewing and editing. A small region (20 pixels by 20 pixels) surrounding the desired point is magnified by a factor of ten. Finally, Window Five displays pertinent system status information and helpful instructions to the user.

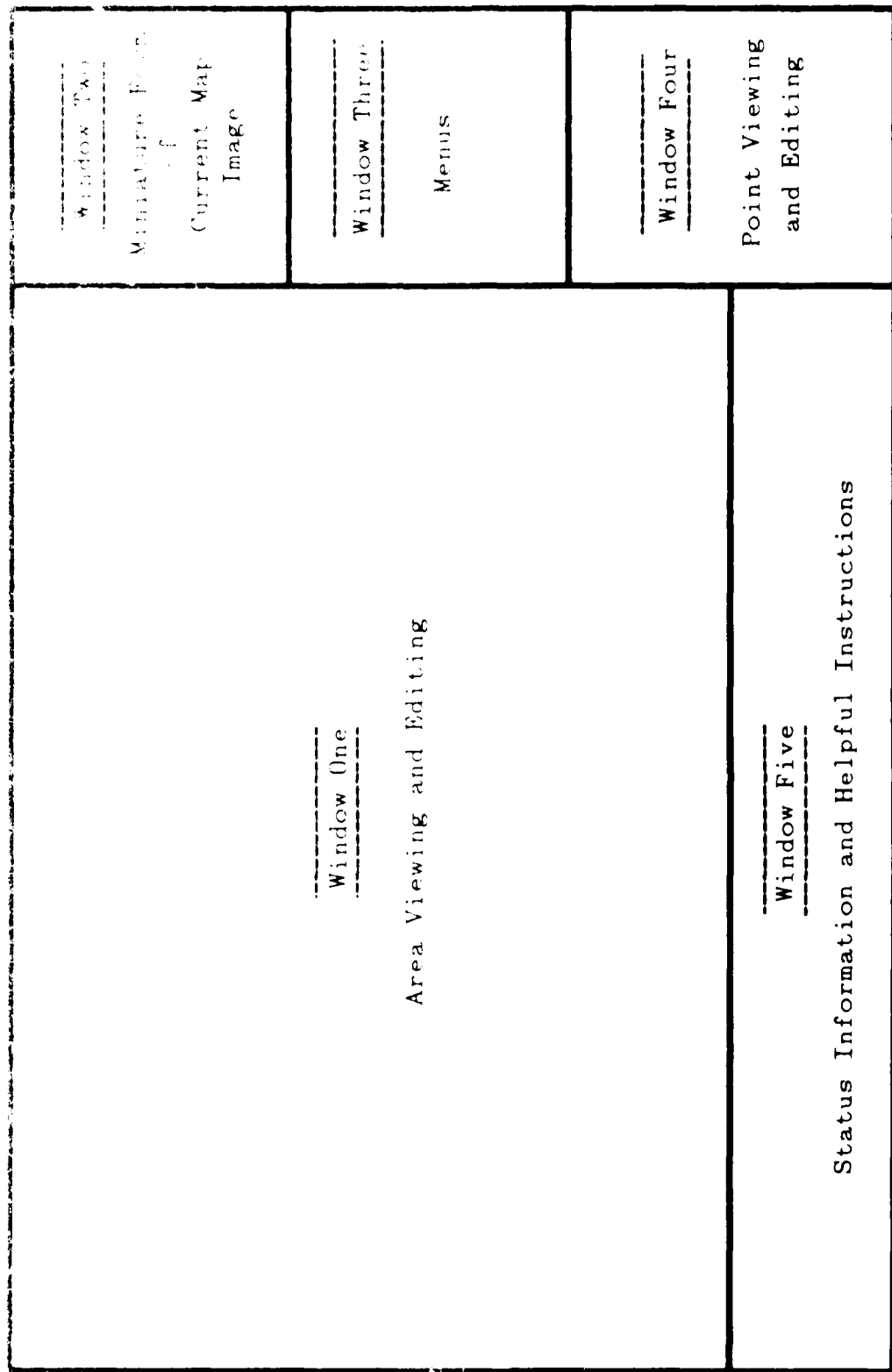


Figure 4.1 Screen Layout for the Editor

The editor also employs temporary and overlapping windows to allow more surface area from a fixed size screen. Designed to preserve partially obscured terminal contents, the windows enable viewing of information on drawing selections, feature classifications and directory listings. These overlays are displayed only upon request.

### C. INPUT DEVICES

User responses are entered either through the IRIS keyboard or three button mouse. The keyboard's function is limited to the entry of input/output filenames, desired directories for display, and feature categorization codes. Remaining input is regulated by the mouse.

The three mouse buttons (Left, Middle and Right) offer flexible and easy control. By properly positioning the cursor on the monitor screen and simultaneously pressing the appropriate button, the user can effortlessly perform such actions as the selection of menu choices, areas to be updated and different drawing options. Movement of the mouse across a hard surface dictates the cursor's actual location on the screen.

### D. MENU STRUCTURE

The editing system consists of a series of *pop up* menus organized in a hierarchical structure (see Figure 4.2). Source code for these menus was derived by modifying a general purpose menu package provided by Silicon Graphics.

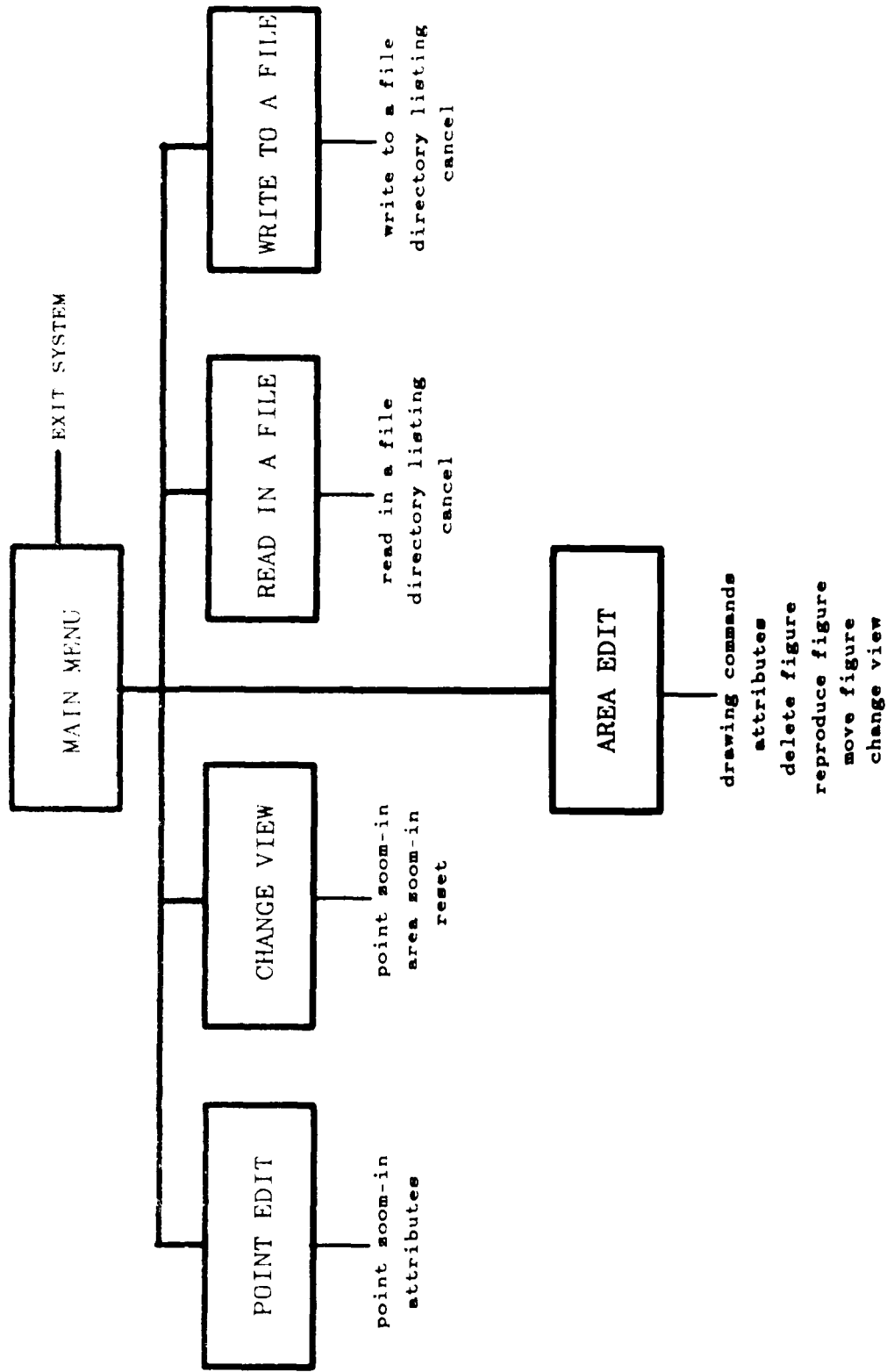


Figure 4.2 Menu Structure for Editing System

The *pop up* menu routine is very simple. When invoked, a menu is automatically drawn on the screen displaying the possible program selections. By moving the cursor with the mouse device across an option, that entry is highlighted. Upon the depression of any mouse button, three actions occur :

- ☛ the entry under the cursor at that time is selected,
- ☛ the menu disappears from the screen, and
- ☛ the program branches to the procedure specified by the option.

If the cursor lies completely outside the menu field when the button is pressed, the menu remains on the screen and the routine continues to wait until an appropriate response is entered.

A description of the operations and capabilities for each of the available commands follows.

1. Area Edit

This procedure permits large region editing of the map by allowing the user to add, delete, move and change displayed components. Refer to the section "Point Edit" for modifying individual pixels.

Features are designed by combining different drawing commands and attributes. Other modifications such as deleting, reproducing, moving and changing attribute settings require the desired figure be identified. Selection of an individual object is possible by positioning the cursor over it and then depressing the middle mouse button. The editor will flash the chosen figure and request

confirmation of its correctness. A rejection by the user causes the system to search for the next best solution. This process of searching and requesting approval continues until the user is satisfied with an object or when no further figures are found. If a selection is made, the current editing action is then performed. A warning is displayed when no figure is found.

Six choices are available in the Area Edit Menu :

- (1) **Drawing Commands** - allow feature sketching using one or more of the following shapes<sup>4</sup> :

- Circle
- Rectangle
- Polygon
- Horizontal Line
- Vertical Line
- Single Line
- Continuous Line

The last produced shape can either be erased or reproduced without entering the "Delete Figure" or "Reproduce Figure" choices of the Area Edit Menu. The new shape constructed will be situated at the cursor's current location for further repositioning on the map.

- (2) **Attributes** - enable viewing or redesignation of the current setting or a specific figure's characteristics. This option comprises four subcommands:

- View Attribute Set
- Change Attribute Set
- View Figure Attributes
- Change Figure Attributes

Selection of the first subcommand provides a display of the current attribute settings without modification.

---

<sup>4</sup> CAUTION - The IRIS system does not support concave polygons; drawing these figures may produce unpredictable results.



The second subcommand allows any of the current attributes to be changed. When the editor is first entered, they are initially set to the following values :

Color	- White
Linestyle	- Continuous
Linewidth	- One pixel wide
Fill Pattern	- Solid
Feature Code	- 001 (nonvegetative)

The first four settings are modified by choosing a desired alternative from among the visually displayed *pop up* pallets. There are twelve colors (black, <light and dark> blue, <light and dark> brown, <light and dark> green, <light and dark> purple, red, yellow (unknown) and white), fourteen linestyles, five linewidths and ten fill patterns. Feature codes are altered by entering in the new value after a system prompt. Depression of the PF1 key displays the entire listing of descriptions with corresponding codes for review.

The third subcommand allows one to view a particular figure's attributes without modification.

To change an individual figure's attributes, the fourth subcommand is employed. After the figure is located, current settings for the figure are displayed. Changes are entered in a similar manner as described in the process for altering current attributes.

- (3) **Delete Figure** - removes the selected object from the display. The vacancy is filled with the color white and identified as nonvegetative.
- (4) **Reproduce Figure** - produces a carbon copy of the selected object with identical attribute settings.
- (5) **Move Figure** - repositions the selected object to any desired region on the map. The cancel option returns the object to its original location.
- (6) **Change View** - refer to the section "Change View" discussed below.

## 2. Point Edit

Area Editing commands can become quite cumbersome when performing slight alterations to existing features or when identifying unknown regions around border lines. The Point Edit facility offers a viable alternative. With this procedure, one can easily modify individual pixels within a map display using the following two commands :

- (1) **Point Zoom-In** - refer to the section "Change View".
- (2) **Attributes** - allow viewing or redesignation of the color and/or feature description of the current setting or of any specified pixel within the map. The desired point to be updated is first located with the Point Zoom-In command. Selected pixels assume the currently specified color and feature categories. By default, these settings are initialized to nonvegetative (color = white, feature code = 001). The editing color may be changed at any time from a pallet of map colors displayable upon request. Feature description codes are modified by entering in the new value after the system prompt. The entire listing of description codes can be viewed by depressing the PF1 key. Changes recorded on the magnified area are also reflected in the current map view.

## 3. Change View

Magnification of different map sections may be necessary to assist the human eye in correctly perceiving and locating small features requiring updating.

Three options are provided by the editor :

1. **Area Zoom-In** - 2x and 4x magnification of selected user-defined areas. The current map view is overwritten by the enlarged area. (screen location : Window One).
2. **Point Zoom-In** - 10x magnification on a 20 pixel x 20 pixel area surrounding the desired point. (screen location : Window Four).

- (3) **Reset** - restores the screen to the original map view.

Exiting this procedure, when called from the Main Menu, invokes the Reset command.

4. Read in a File

This choice allows entry of previously stored map data for further editing. After display of the system prompt, the user types in the desired filename at the keyboard. If the file exists, the data is drawn on the screen overwriting any prior map images; otherwise, an appropriate error message appears with control returning to the Main Menu. The user may cancel any inadvertent calls made to this routine by using the Escape key on the keyboard.

A directory listing facility is also available. By depressing the PF1 key, a prompt is shown requesting the desired directory name: the user's current directory serves as default. This facility is capable of listing all files in any directory not read protected.

5. Write to a File

To save changes entered by the user, a method is necessary to capture the screen display as stored data within a file. This command selection provides such a capability. The projected map image is transformed into similarly colored areas which are then subdivided into polygons before being written to the created file. Specification of an existing filename results in the destruction and

overwriting of any old contents. The actual algorithm used for this procedure is presented in detail in Chapter Five.

The process of designating an output filename in addition to the Escape and PF1 key functions are identical to those previously described (refer to the section "Read in a File").

#### 6. Exit System

Selection of this option terminates the editing session. Changes not previously saved are lost forever. The graphics hardware system is returned to its initial state. Input devices are unqueued, the viewport and textport are restored to full screen, and the communication buffers are flushed.

### E. SYSTEM PROMPTS

System information and prompts appear on the screen at various times during the execution of the editor. The prompts serve as a guide through an editing session without reference to an operating manual. These helpful messages reduce the amount of information the user must memorize while keeping him informed of what is happening now and what to expect next. They include :

- 1) **Menus** - allow selection among multiple program alternatives. (screen location : Window Three).
- 2) **Key Function Instructions** - display the active keyboard keys at any given time. The keys are identified by dark blue squares with an abbreviation of the key inside (for example, ESC signifies the Escape Key). Keys not listed are inactive; pressing them has no effect. (screen location : Window Five).

- (3) **Mouse Function Instructions** - list the outcome of depressing any of the three mouse buttons. The buttons are represented by dark blue squares with a letter ( **L** for left **M** for middle and **R** for right ) inside. Buttons not shown are inactive; pressing them has no effect. (screen location : Window Five)
- (4) **System Status** - report on the program's reaction under the following conditions - reading in a file, writing out a file, delay due to processing, and no action taken. (screen location : Window Five).
- (5) **Error Messages** - warn when erroneous or adverse conditions are present (for example, invalid input). Depending on the situation, the system tries to recover by returning to its last working state. (screen location : Window One).
- (6) **Other Prompts** - request additional input information from the user such as a filename, directory name and/or feature category code. (screen location : Window One).

## F. FEATURES SUPPORTED

Cartographic components are designed and updated using the Area and Point Edit options described above. The drawing commands and attributes, provided by the editing system, enable the user to generate most of the symbology related to USGS maps (see Appendix B).

A three-digit description code assists in feature identification. During the pre-processing stages, all components are initially classified into one of the following broad categories based on color:

Color	Code	Description
Yellow	000	Unknown
White	001	Nonvegetative
Black	002	Man made object

Red	003	Road
Blue	004	Body of Water
Brown	005	Contour line
Green	006	Vegetation
Purple	007	Feature from aerial photo

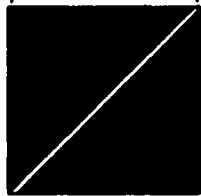
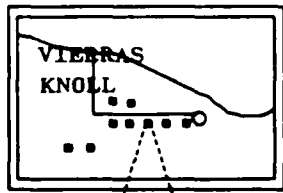
These initial settings can be redefined with either edit facility. The remaining possible feature descriptions with corresponding program defined codes are listed in Appendix B.

#### G. FILE FORMAT

The cartographic data base used for this study was generated from the processing scheme presented in Chapter Two. The data within each file represents a section from a USGS map stored in binary format.

Each map section is divided into a number of uniformly colored areas. An area is identified by a set of short integer codes representing its color, description and allowable speed zone. Each area is further subdivided into a number of polygons as determined by the Data Compression Algorithm. The polygons are defined by short integer codes representing their type (point, line, polygon, filled polygon) and vertex locations. An end of area marker separates each successively defined area region. Figure 4.3 depicts a typical file format.

The editor extends the seven map colors to twelve. Light and dark colors for green, blue, purple and brown support cartographic feature contrast in given regions and have no significance with respect to classification. Yellow identifies unknown areas.



==

```

      .
      .
      .
color  description  speed
draw_command vertices
      .
      .
      .
draw_command vertices
      end of area marker
      .
      .
      .
      0 002 0
      3 100 105 105 105 100 100
      3 100 100 105 105 105 100
      -1
      .
      .
      .
end of file

```

- KEY -		
<u>color</u>	<u>draw commands</u>	<u>verticies</u>
0 black	0 point	x1y1
1 white	1 line	x1y1 x2y2
2 red	2 polygon	x1y1 x2y2 x3y3
3 light green	3 filled polygon	x1y1 x2y2 x3y3
4 dark green		
5 light blue		
6 dark blue	<u>end of area</u>	<u>speed</u>
7 light purple	-1	(not used)
8 dark purple		
9 light brown	<u>description codes</u>	
10 dark brown	(see appendix B)	
11 yellow (unknown)		

Figure 4.3 Typical File Format with Key

## V. MANAGING USER REVISIONS

A major concern in the design of any man-machine interface is the ability to insure *real-time* response rates. Users by nature become very anxious when reactions to their requests do not occur within a few seconds of input. Even prompting messages, indicating processing is in progress, lose their effectiveness after a period of time. The user may begin to fidget and start pressing buttons and "trying stuff", perhaps leading to disastrous results. In order to minimize such processing delays experienced during an editing session, an internal indexing scheme was incorporated into the editor's design. This plan, in addition to the method used for updating a data base, are presented below.

### A. INTERNAL INDEXING SCHEME

As mentioned in Chapter Four, each map is divided into several uniformly colored areas which are further subdivided into a number of polygons. Each area is defined as an object.<sup>5</sup> Due to the feature complexity of most cartographic documents, it is not unusual to have in excess of 5000 objects defined at any point in time. This large number places a strain on the IRIS's memory management for locating a particular object quickly. Response time slowly decreases at an inverse

---

<sup>5</sup>An object, the graphics packaging mechanism provided on the IRIS, is composed of a set of `gcl` graphics commands.



rate to the number of objects created. To help alleviate this problem, an artificial indexing scheme (see Figure 5.1) was added to the editing system.

The scheme consists of an indexing array subscripted from 0 through 3277. As an area of the map is entered into memory, it is assigned a short integer identification number. The area is then placed into a linked list, arranged in descending order, at an array location equal to the absolute value of the index number divided by ten.

To locate an object for updating or modifying, the pick mechanism from the IRIS graphics library is employed. The picking facility allows the user to identify an object by maneuvering the cursor as a pointing device. Objects intersecting a small rectangular area around the cursor are considered "hits" and their identification numbers, assigned during input, are placed into a names stack. The indexing scheme, which reduces search time to a maximum of 20 objects, can then be used to quickly locate each object by number for manipulation.

## B. UPDATING THE DATABASE

Upon completion of an editing session, changes to a map are saved by using the "Write to a File" option from the Main Menu. The four algorithms, listed in Appendix C, comprising this command selection include : Update\_Database, Data Compression, Expand\_Area and Subset\_Area. The latter three were written and used by Diehl in the Data Compression phase of the cartographic

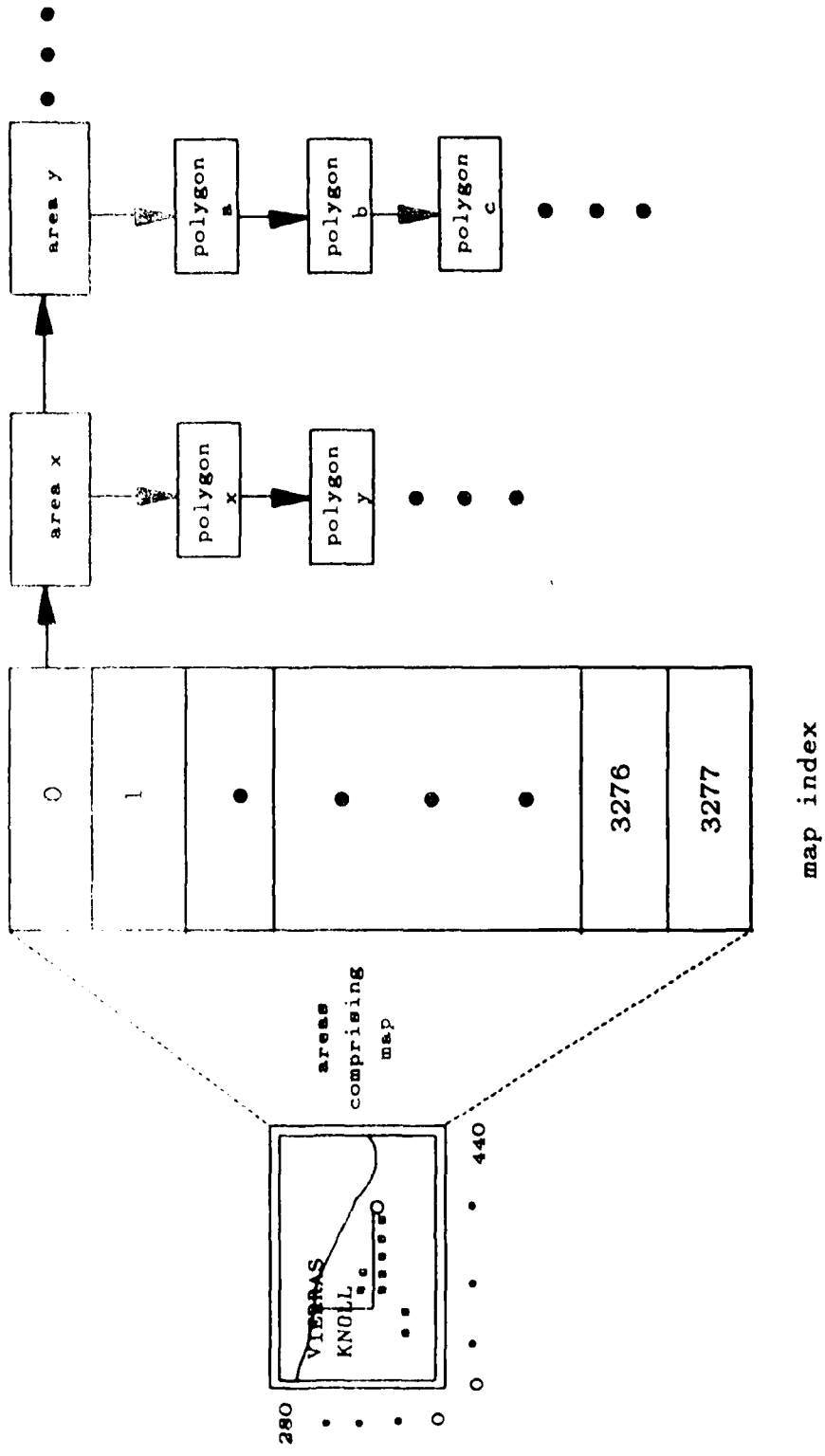


Figure 5.1 Internal Indexing Scheme

rate to the number of objects created. To help alleviate this problem, an artificial indexing scheme (see Figure 5.1) was added to the editing system.

The scheme consists of an indexing array subscripted from 0 through 3277. As an area of the map is entered into memory, it is assigned a short integer identification number. The area is then placed into a linked list, arranged in descending order, at an array location equal to the absolute value of the index number divided by ten.

To locate an object for updating or modifying, the pick mechanism from the IRIS graphics library is employed. The picking facility allows the user to identify an object by maneuvering the cursor as a pointing device. Objects intersecting a small rectangular area around the cursor are considered "hits" and their identification numbers, assigned during input, are placed into a names stack. The indexing scheme, which reduces search time to a maximum of 20 objects, can then be used to quickly locate each object by number for manipulation.

## B. UPDATING THE DATABASE

Upon completion of an editing session, changes to a map are saved by using the "Write to a File" option from the Main Menu. The four algorithms, listed in Appendix C, comprising this command selection include : Update\_Database, Data\_Compression, Expand\_Area and Subset\_Area. The latter three were written and used by Diehl in the Data Compression phase of the cartographic

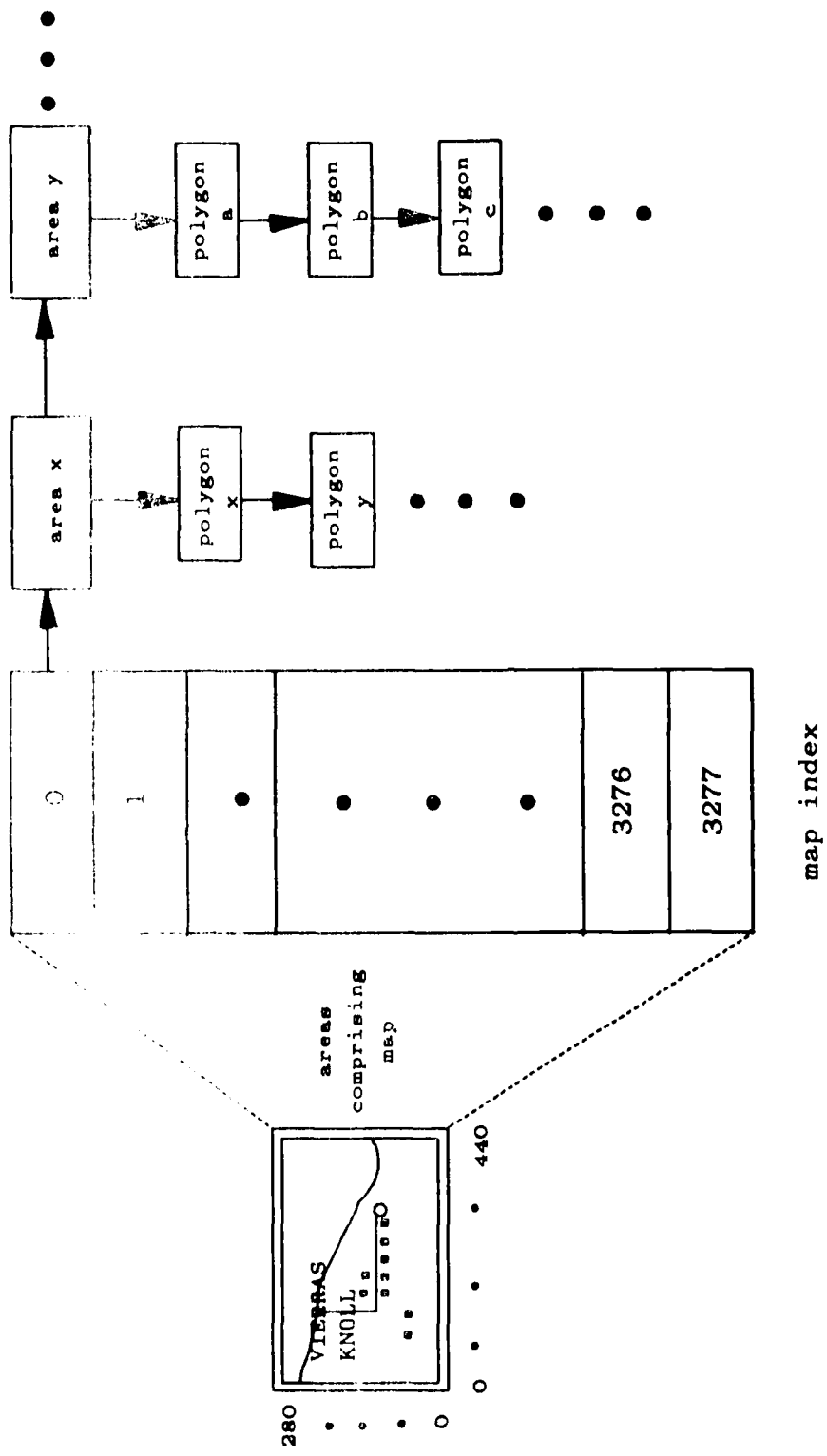


Figure 5.1 Internal Indexing Scheme

processing package. Slight modifications were added to enable handling of a larger set of colors. Each of the algorithms are described below.

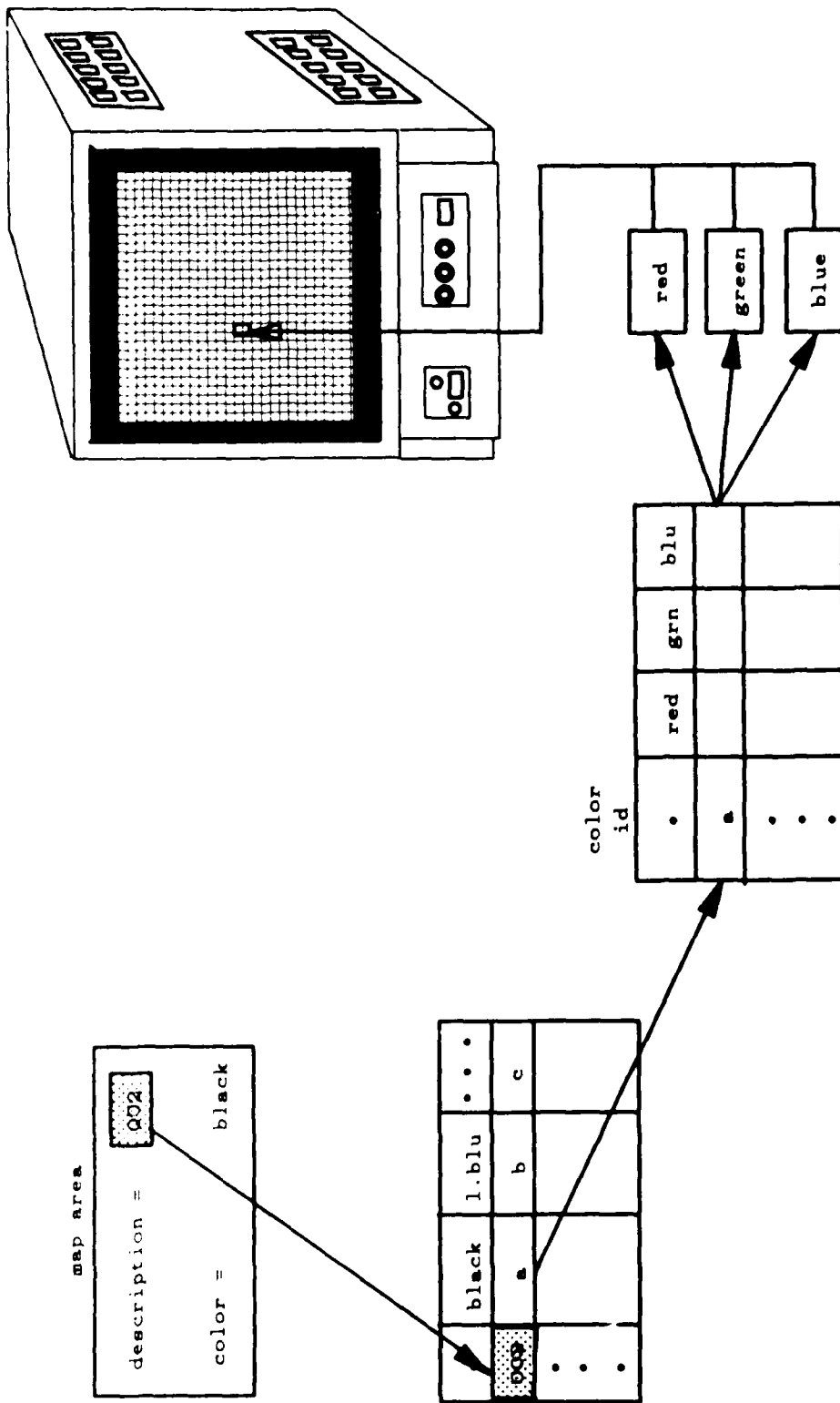
1. Update Database

Since the primary use for these data bases is in autonomous vehicle route planning, it is desirable for components to be uniquely represented in polygonal form. Duplicated areas, as seen in the case of overlying features, need to be eliminated. The Update\_Database Algorithm provides this capability.

The program itself assigns a set of unique color shades, representing the twelve possible map colors, to each feature description. During an editing session, default colors are actually used for display and to reflect entered changes. This algorithm replaces the default color for each area of the map with an appropriate color shade as dictated by the area's description code (see Figure 5.2a). After reassignment, the map is redrawn. The new map image is subsequently read off the screen and funneled through the Data\_Compression Algorithm (see Figure 5.2b).

2. Data Compression

The entire map surface is constructed as a grid, each cell being one pixel wide. A duplicate setup, hereafter known as the *scratch pad*, has all its cells initially set to "0". Working in a sweeping motion from left to right and top to bottom on the proposed map layout, the algorithm checks for unexamined pixels. When one is found, the process converts the pixel's color shade back to the default color (see Figure 5.2b) and writes the color, description code and speed



002 = Man Made Object

Figure 5.2a Updating the Data Base

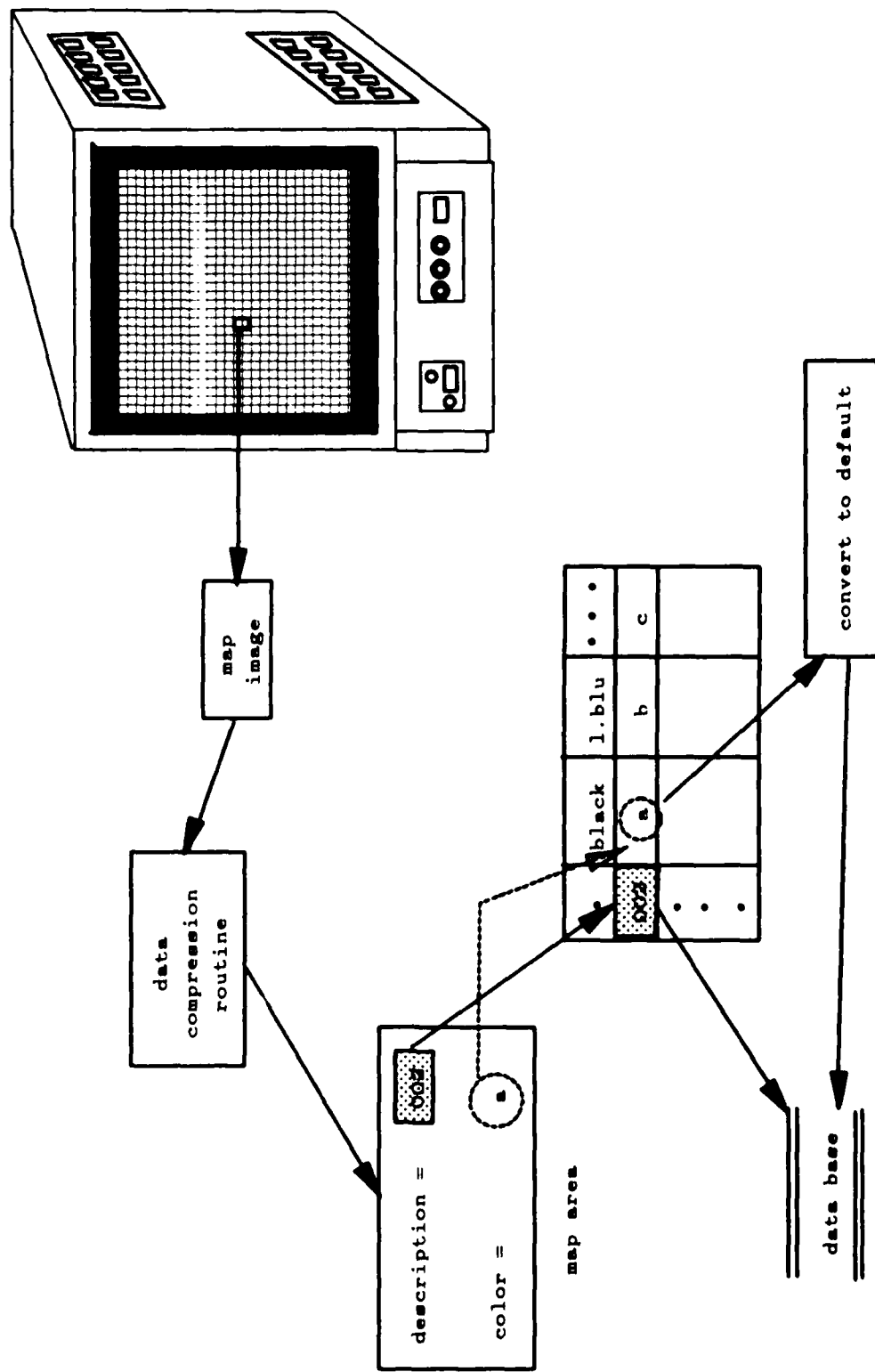


Figure 5.2b Updating the Data Base

code to a file. The Expand Area and Subset Area algorithms are invoked prior to outputting an end of area marker. This cycle persists until each pixel of the map has been surveyed.

### 3 Expand Area

This algorithm focuses on a three by three subgrid successively encompassing each cell of the map. Three actions occur when any of the surrounding eight neighbors have the same color as the center. For each similarly colored pixel:

- the pixel is tagged as used to avoid duplicate inspections,
- a flag is set to "1" in the appropriately corresponding position of the *scratch pad*, and
- the pixel's location is placed in a queue for later examination of its neighbors.

The interrogation process continues until the queue is empty, resulting in a uniformly colored region marked by set flags in the *scratch pad*.

### 4 Subset Area

The identified homogeneous area marked in the *scratch pad* is then subdivided using the ten possible geometric shapes shown in Figure 5.3. Items a through e represent right equilateral triangles, items f through i describe lines, and item j defines a point. [Ref. 2] addresses the actual selection criteria for these forms and the process involved in dividing intact area regions. Upon completion



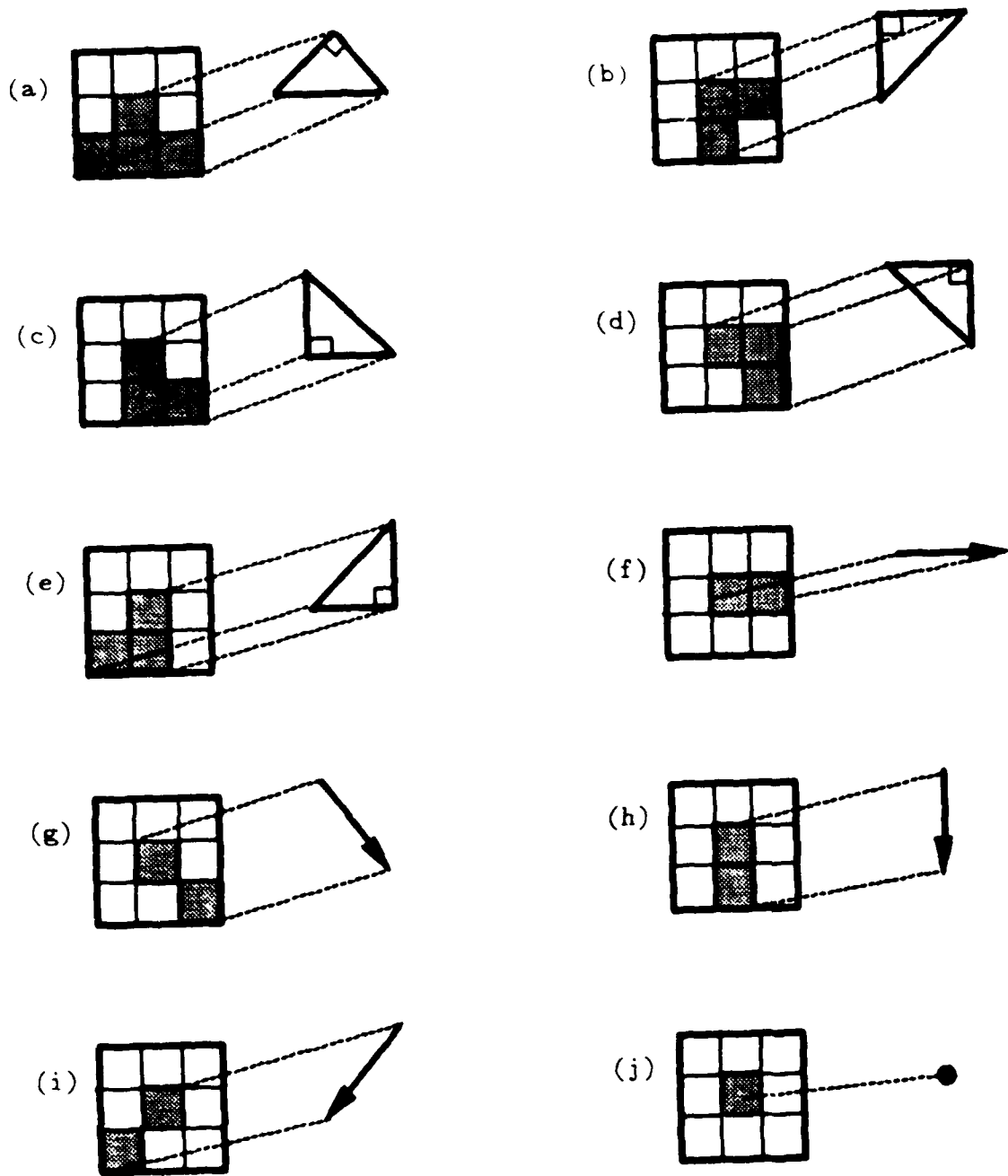


Figure 5.3 Polygonal Encoding Scheme

of the subdivision, *scratch pad* flags are reset to "0" and the area identification phase is reexecuted.

The above method is a simple scheme allowing compact storage for processed and edited information. The resulting data base itself contains a cartographic image divided into homogeneous areas which are further subdivided into polygons. Each polygonal side and/or vertex is uniquely defined avoiding redundant representations. One and only one feature classification is attached to each polygon.

## VI. CONCLUSIONS AND RECOMMENDATIONS

### A. CONCLUSIONS

This study examines an interactive editing system for maintaining data bases generated from a cartographic processing system currently under development at the Naval Postgraduate School. The amount of updating necessary on any stored map depends on the care taken in digitization, the accuracy of the image processing phase and the level of quality required in the finished product. In general, the time devoted towards this effort is proportional to the map's complexity. If each step involved in the generation of the data base has been carefully executed and there are no mistakes in content, the editing is mostly cosmetic.

The editor was written in the C programming language and contains approximately 10,000 lines of code. Driven by a set of *pop up* menus arranged in a hierarchical structure, the system has been designed to handle modifications to either large map areas or individual points. The latter capability is useful in correcting processing and digitizer induced errors around linear features. A mouse served as the primary input device and proved to be fairly accurate in manipulating and designing cartographic data. Display colors approximate those used on the paper map.

Preliminary tests indicate the editor's performance depends on the map's complexity and the hardware's internal memory size. The system performs best using an IRIS with six megabytes of memory and a floating-point accelerator.

## B LIMITATIONS AND RECOMMENDED EXTENSIONS

The editor developed in this study merely scratches the surface of the problem in applying interactive techniques to update and modify cartographic data bases. Several limitations prevent the present system from operating at an as yet practical level in a real time environment. A few of these limiting factors with possible research extensions are discussed below :

1. Each preprocessed map contained within the database defines a 280 x 440 pixel area. This self-imposed size constraint is addressed in [Ref. 2]. Many cartographic features commonly span across several adjoining map sections. Ideally, any portion of these features should be readily available for editing, reflecting all updates. The current editing system is unable to handle changes occurring over several neighboring regions. Modifications are instead restricted to within one defined map section at any given time.

EXTENSION : Addition of a facility to merge adjacent sectional map pieces and modification of the editor to handle the larger areas.

2. USGS cartographic features served as a standard for this research. They are designed by combining different colors, linewidths, linestyles and fill patterns. The editing system, however, lacks appropriate checks for preventing entry of illegal combinations which may render a data base as unusable.

EXTENSION : Incorporation of proper error checking to the editor to minimize such mishaps.

3. Some USGS cartographic features are not supported by the editor due to their inherent complexity and the difficulty of approximating their display by setting bit maps such as with Tailings, Intricate Surface Areas and

Gravel Beaches. Others require several overlays in order to capture the proper design. The latter case, in particular, is very tiring to the user and inefficient for the computer. Additionally, the algorithm employed in updating the database (discussed in Chapter Five) fails to preserve the integrity of areas composed of complex fill patterns. Instead, these areas are needlessly splintered into smaller regions. Many of the standard cartographic symbols look nice on paper but significantly tax memory requirements in duplication efforts on the computer.

EXTENSION : Development of a more efficient graphical representation for mapping symbols. The proposed scheme should allow one to depict the various natural, man-made and aerial features. The result achieved should be equal to traditional cartographic techniques as far as accuracy, integrity of design and quality of reproduction.

4. The Data Compression Algorithm used for subdividing a map produces areas composed of many irregularly sized polygons having no common sides. Due to these variations, editing of sub-areas tends to be awkward.

EXTENSION : Modification of the Data Compression Algorithm to produce more equally sized subdivisions.

5. Curved features (contours, winding roads/rivers, etc) can only be approximated by using several straight line segments. The accuracy of these features is dependent on the user's drawing skills.

EXTENSION : Development of a procedure capable of generating accurate curved lines without excessive memory or execution requirements.

6. No text labeling procedures are provided by the editor.

EXTENSION : This facility is not necessary unless the editor's role is extended to uses requiring that capability.

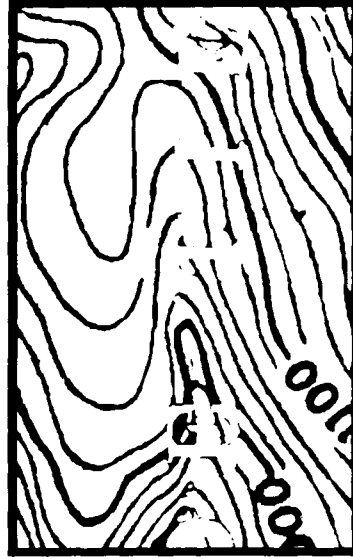
Despite a number of limitations, the editing system illustrates the useful role interactive computer graphics plays in maintaining geographic data bases. Additional studies are needed to expand and improve upon the current system's

capabilities of allowing new editions and revisions without expensive recompilation.

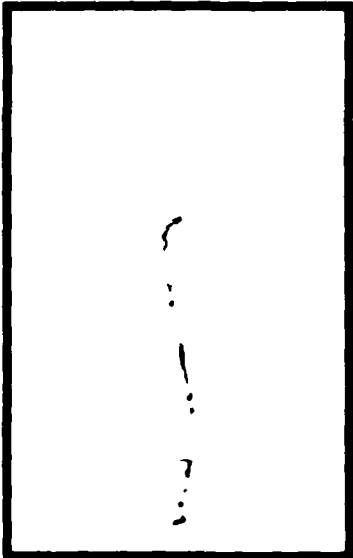
APPENDIX A SAMPLE EDITING SESSION



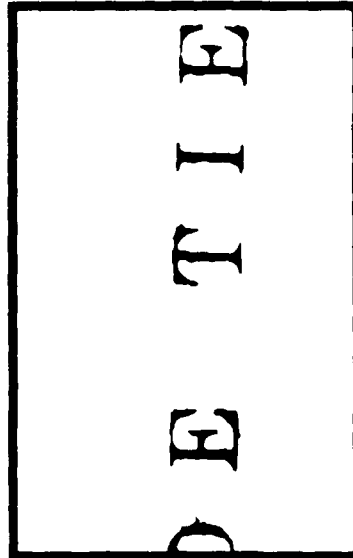
GREEN



BROWN

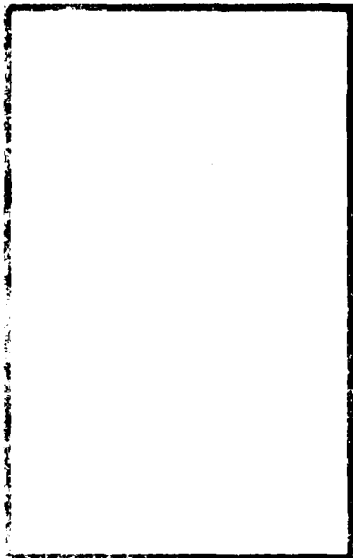


BLUE

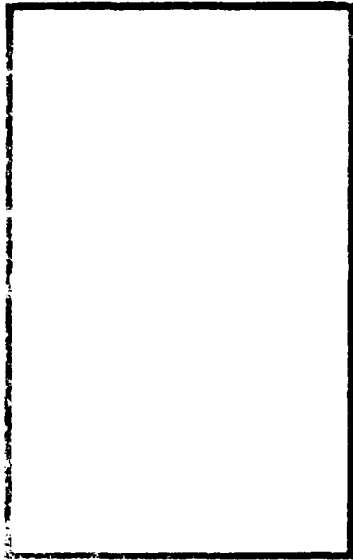


BLACK

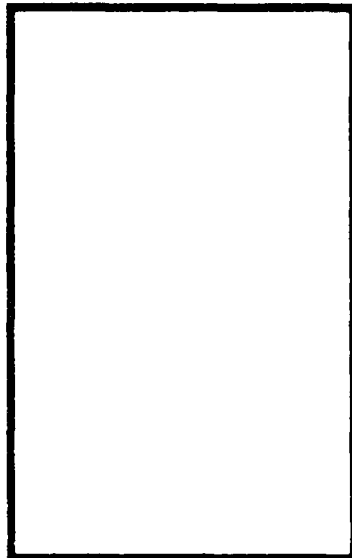
BEFORE 'SNAP SHOT' : prior to Edit Session



RED



UNKNOWN



PURPLE



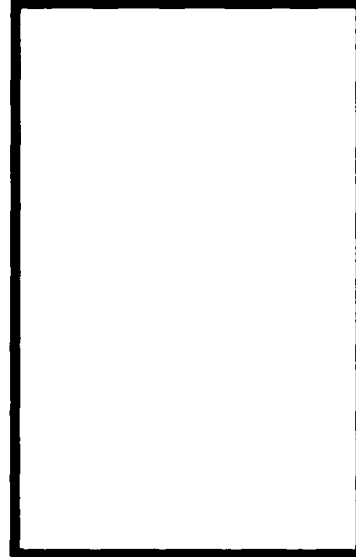
WHITE

BEFORE 'SNAP SHOT' : prior to Edit Session (cont'd)

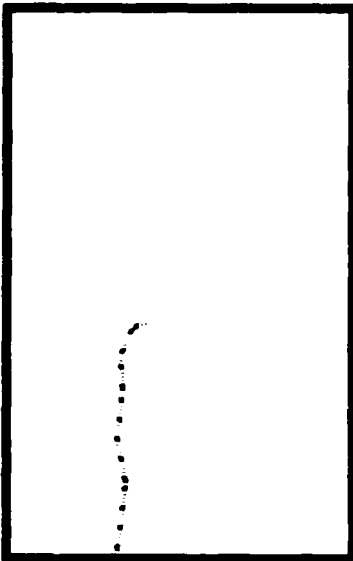




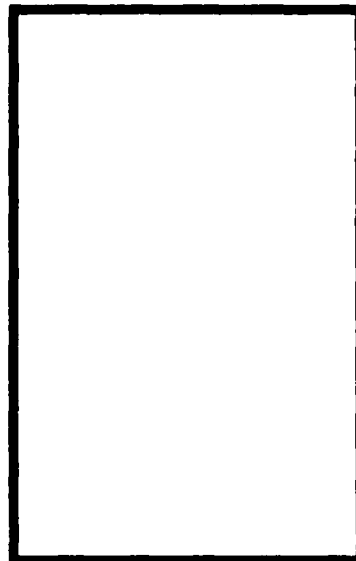
**GREEN**



**BROWN**

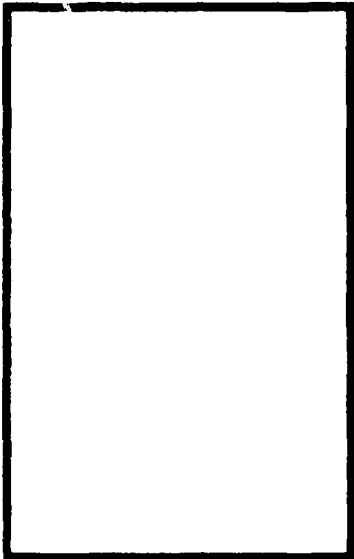


**BLUE**

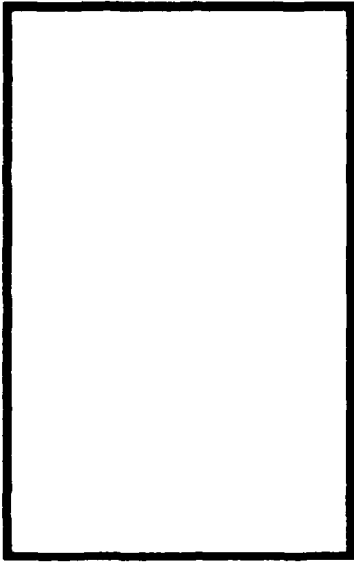


**BLACK**

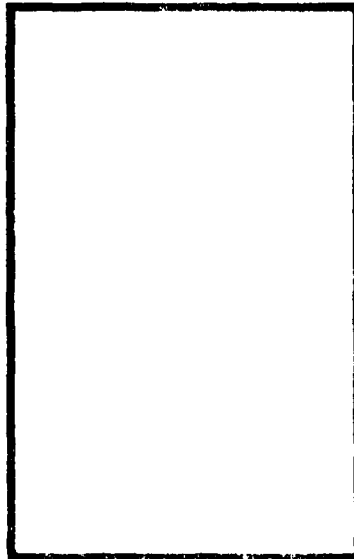
**AFTER 'SNAP SHOT' : after an Edit Session**



RED



UNKNOWN



PURPLE



WHITE

AFTER 'SNAP SHOT' : after an Edit Session (cont'd)

## APPENDIX B - USGS TOPOGRAPHIC MAP SYMBOLS

[Ref. 6]

CODE	DESCRIPTION	
010	<b>CONTROL DATA AND MONUMENTS</b>	
011	Aerial photograph roll and frame number	†
012	Horizontal control:	
013	Third order or better, permanent mark	†
014	With third order or better elevation	†
015	Checked spot elevation	†
016	Coincident with section corner	†
017	Unmonumented	†
018	Vertical control:	
019	Third order or better, with tablet	†
020	Third order or better, recoverable mark	†
021	Bench mark at found section corner	†
022	Spot elevation	†
023	Boundary monument:	
024	With tablet	†
025	Without tablet	†
026	With number and elevation	†
027	U. S. mineral or location monument	
030	<b>BOUNDARIES</b>	
031	National	
032	State or territorial	
033	County or equivalent	
034	Civil township or equivalent	
035	Incorporated city or equivalent	
036	Park, reservation, or monument	
037	Small park	
040	<b>LAND SURVEY SYSTEMS</b>	
041	U. S. Public Land Survey System :	
042	Township or range line	
043	Location doubtful	

CODE	DESCRIPTION	
044	Section line	
045	Location doubtful	
046	Found section corner	+
047	Found closing corner	+
048	Witness corner	+
049	Meander corner	+
050	Other land surveys:	
051	Township or range line	
052	Section line	
053	Land grant or mining claim	
054	Monument	
055	Fence line	

#### ROADS AND RELATED FEATURES

060	
061	Primary highway
062	Secondary highway
063	Light duty road
064	Unimproved road
065	Trail
066	Dual highway
067	Dual highway with median strip
068	Road under construction
069	Underpass
070	Overpass
071	Bridge
072	Drawbridge
073	Tunnel

#### BUILDINGS AND RELATED FEATURES

080	
081	Dwelling or place of employment : small
082	Dwelling or place of employment : large
083	School
084	Church
085	Barn, warehouse, etc. : small
086	Barn, warehouse, etc. : large

CODE	DESCRIPTION
087	House omission tint
088	Racetrack
089	Airport
090	Landing strip
091	Well (other than water)
092	Windmill
093	Water tank: small
094	Water tank: large
095	Other tank: small
096	Other tank: large
097	Covered reservoir
098	Gaging station
099	Landmark object
100	Campground
101	Picnic area
102	Cemetery: small
103	Cemetery: large
<b>RAILROADS AND RELATED FEATURES</b>	
110	Standard gauge single track
111	Standard gauge single track
112	Station
113	Standard gauge multiple track
114	Abandoned
115	Under construction
116	Narrow gauge single track
117	Narrow gauge multiple track
118	Railroad in street
119	Juxtaposition
120	Roundhouse and turntable
<b>TRANSMISSION LINES AND PIPELINES</b>	
130	Power transmission line: pole
131	Power transmission line: tower
132	Power transmission line: tower
133	Telephone or telegraph line
134	Aboveground oil or gas pipeline
135	Underground oil or gas pipeline

CODE	DESCRIPTION
------	-------------

140	<b>CONTOURS</b>
-----	-----------------

141	Topographic:
142	Intermediate
143	Index
144	Supplementary
145	Depression
146	Cut
147	Fill
148	Bathymetric:
149	Intermediate
150	Index
151	Primary
152	Index Primary
153	Supplementary

160	<b>MINES AND CAVES</b>
-----	------------------------

161	Quarry or open pit mine
162	Gravel, sand, clay, or borrow pit
163	Mine tunnel or cave entrance
164	Prospect
165	Mine shaft
166	Mine dump
167	Tailings +

170	<b>SURFACE FEATURES</b>
-----	-------------------------

171	Levee
172	Sand or mud area, dunes, or shifting sand
173	Intricate surface area +
174	Gravel beach or glacial moraine +
175	Tailings pond

CODE	DESCRIPTION
------	-------------

180	<b>VEGETATION</b>
-----	-------------------

181	Woods
182	Scrub
183	Orchard
184	Vineyard
185	Mangrove

190	<b>MARINE SHORELINE</b>
-----	-------------------------

191	Topographic maps:
192	Approximate mean high water
193	Indefinite or unsurveyed
194	Topographic-bathymetric maps:
195	Mean high water
196	Apparent (edge of vegetation)

200	<b>COASTAL FEATURES</b>
-----	-------------------------

201	Foreshore flat
202	Rock or coral reef
203	Rock bare or awash
204	Group of rocks bare or awash
205	Exposed wreck
206	Depth curve
207	Sounding
208	Breakwater, pier, jetty, or wharf
209	Seawall

210	<b>BATHYMETRIC FEATURES</b>
-----	-----------------------------

211	Area exposed at mean low tide
212	Sounding datum
213	Channel
214	Offshore oil or gas: well
215	Offshore oil or gas: platform
216	Sunken rock

## CODE

## DESCRIPTION

CODE	DESCRIPTION
220	<b>RIVERS, LAKES, AND CANALS</b>
221	Intermittent stream
222	Intermittent river
223	Disappearing stream
224	Perennial stream
225	Perennial river
226	Small falls
227	Small rapids
228	Large falls
229	Large rapids
230	Masonry dam
231	Dam with lock
232	Dam carrying road
233	Intermittent lake or pond
234	Dry lake
235	Narrow wash
236	Wide wash
237	Canal, flume, or aqueduct with lock
238	Elevated aqueduct, flue, or conduit
239	Aqueduct tunnel
240	Water well
241	Spring or seep
250	<b>GLACIERS AND PERMANENT SNOWFIELDS</b>
251	Contours and limits †
252	Form lines †



CODE	DESCRIPTION
260	<b>SUBMERGED AREAS AND BOGS</b>
261	Marsh or swamp
262	Submerged marsh or swamp
263	Wooded marsh or swamp
264	Submerged wooded marsh or swamp
265	Rice field
266	Land subject to inundation

\* Features not supported by the editor.

Note : The editor does not provide automatic symbol plotting. Features containing curved lines can only be approximated using small line segments.

## APPENDIX C UPDATE DATABASE ALGORITHMS

---

```
begin UPDATE DATABASE algorithm
{
  initialize variables;
  initialize map region (TOTAL_ROWS x TOTAL_COLUMNS)
    pixel colors to unknown;
  initialize scratch pad values to 0;
  create a border around map region;

  /* index through each area of the map */
  for (i = 0; i < MAX_DIMENSION; ++i)
  {
    new_area = mapindex[i].next_area;

    /* examine all areas at the same index reference point */
    while (new_area = exist)
    {
      change new_area colortype to new_area description colorshade;

      /* increment to next area */
      new_area = new_area -> next_area;
    }
  }

  redraw map on screen;
  read color values within map off the screen;
  convert colorshades to appropriate processing color;

  call data compression algorithm;
}
end UPDATE DATABASE algorithm
```

---

---

```

begin DATA_COMPRESSION algorithm
{
  initialize variables;

  /* create file to hold compressed data representation */
  create output file;

  /* successively examine each pixel of the map region */
  for (i = 1; i <= TOTAL_ROWS; i++)
  {
    for (j = 1; j <= TOTAL_COLUMNS; j++)
    {
      /* if pixel has not been examined as yet */
      if (pixel(i, j) = not used)
      {
        convert pixel color back to description colorshade;
        determine true map color;
        determine description of map area;

        write true map color to output file;
        write description code to output file;
        write speed code to output file;

        /* expand then subdivide homogeneous areas */
        call expand_area algorithm;
        call subset_area algorithm;

        write end of area marker to output file;
      }
    }
  }
}
end DATA_COMPRESSION algorithm

```

---

### Data Compression Algorithm

---

```

begin EXPAND AREA algorithm
{
    set pixel(x, y) = used;
    set corresponding (x, y) location in scratch pad to 1;
    insert pixel (x, y) location into the queue;

    /* examine only pixel values within the queue */
    while (queue = not empty)
    {
        remove (x, y) value off the queue;

        /* check the surrounding eight neighbors of the grid's center to
           see if any are similar in color */
        for (i = (x-1); i <= (x+1); i++)
        {
            for (j = (y-1); j <= (y+1); j++)
            {
                /* tag map region and scratch pad location of similiarly
                   colored neighbor. Make queue entry of pixel's location */
                if ((pixel_color(i, j) = pixel_color(x, y)) and (pixel(i, j) = not used))
                {
                    set pixel(i, j) = used;
                    set corresponding (x, y) location in scratch pad to 1;
                    set maximum x value = maximum(previous x value, i);
                    insert pixel (x, y) location into the queue;
                }
            }
        }
    }
}
end EXPAND AREA algorithm

```

---

### Expand Homogeneous Area Algorithm

---

```

begin SUBSET_AREA algorithm
{
  do
  {
    /* initialize temp variables to selected positions within the
       scratch pad */
    set temp0 = scratch pad area (x, y+1);
    set temp1 = scratch pad area (x+1, y+1);
    set temp2 = scratch pad area (x+1, y);
    set temp3 = scratch pad area (x+1, y-1);

    .. (temp0 = 1)
    {
      if (temp1 = 1)
      {
        if (temp 2 = 1)
        {
          expand type_two polygon; /* Figure 5.3, item b */
          write triangle vertices to output file;
        }
        else
        {
          expand type_four polygon: /* Figure 5.3, item d */
          write triangle vertices to output file;
        }
      }
      else if (temp2 = 1)
      {
        if (temp3 = 1)
        {
          expand type_five polygon; /* Figure 5.3, item e */
          write triangle vertices to output file;
        }
      }
    }
  }
}

```

---

Algorithm for Subdividing an Area into Polygons

---

```

else
{
    expand type six polygon:      /* Figure 5.3, item f */
    write triangle vertices to output file;
}
}
else if (temp2 = 1)
{
    if (temp1 = 1)
    {
        if (temp 3 = 1)
        {
            expand type one polygon:      /* Figure 5.3, item a */
            write triangle vertices to output file;
        }
        else
        {
            expand type three polygon:     /* Figure 5.3, item c */
            write triangle vertices to output file;
        }
    }
}
else if (temp3 = 1)
{
    expand type five polygon:      /* Figure 5.3, item e */
    write triangle vertices to output file;
}
else
{
    expand type eight polygon:      /* Figure 5.3, item h */
    write line endpoints to output file;
}
}

```

---

Algorithm for Subdividing an Area into Polygons (cont'd)

---

```

else if (temp3 = 1)
{
    expand type_nine polygon;          /* Figure 5.3, item i */
    write line endpoints to output file;
}
else if (temp1 = 1)
{
    expand type_seven polygon;        /* Figure 5.3, item g */
    write line endpoints to output file;
}
else
{
    expand type_ten polygon;          /* Figure 5.3, item j */
    write point location to output file;
}

    reset scratch pad region flags;
}
while (x <= maximum x value);

}
end SUBSET_AREA algorithm

```

---

Algorithm for Subdividing an Area into Polygons (cont'd)

## LIST OF REFERENCES

1. Richbourg, R. F., Neil C. Rowe and Michael J. Zyda. "Exploiting Capability Constraints to Solve Global, Two Dimensional Path Planning Problems." Technical Report NPS52-86-006, Monterey, California : Department of Computer Science, Naval Postgraduate School, January 1986.
2. Diehl, Roger K. "Two-Dimensional Polygonal Representation of Maps For Use with Autonomous Vehicle Route Planning." Master's Thesis, Department of Computer Science, Naval Postgraduate School, Monterey, California, June 1986.
3. Fegeas, Robin G. and others. USGS Digital Cartographic Data Standards. Land Use and Land Cover Digital Data. Geological Survey Circular 895-E. 1984.
4. Alder, William R. and Atef A. Ellassal. USGS Digital Cartographic Data Standards. Digital Line Graphs from 1:24,000 - Scale Maps. Geological Survey Circular 895-C. 1984.
5. Silicon Graphics Incorporation. IRIS User's Guide Version 2.1. Document Number 5001-051-001-1. 1985.
6. U. S. Department of the Interior Geological Survey. National Mapping Program. Topographic Map Symbols. National Large Scale Series.



*Distribution List*

Defense Technical Information Center,  
Cameron Station,  
Alexandria, VA 22314 2 copies

Library, Code 0142  
Naval Postgraduate School,  
Monterey, CA 93943 2 copies

Center for Naval Analyses,  
2000 N. Beauregard Street,  
Alexandria, VA 22311

Director of Research Administration,  
Code 012,  
Naval Postgraduate School,  
Monterey, CA 93943

Dr. Henry Fuchs,  
208 New West Hall (035A),  
University of North Carolina,  
Chapel Hill, NC 27514

Dr. Kent R. Wilson,  
University of California, San Diego  
B-014,  
Dept. of Chemistry,  
La Jolla, CA 92093

Dr. Guy L. Tribble, III  
Next, Inc.  
3475 Deer Creek Road,  
Palo Alto, California 94304

Bill Atkinson,  
Apple Computer,  
20525 Mariani Ave.  
Cupertino, CA 95014

Dr. Victor Lesser,  
University of Massachusetts, Amherst  
Dept. of Computer and Information Science,  
Amherst, MA 01003

Dr. Gunther Schrack,  
Dept. of Electrical Engineering,  
University of British Columbia,  
Vancouver, B.C., Canada V6T 1W5

Dr. R. Daniel Bergeron.  
Dept. of Computer Science.  
University of New Hampshire,  
Durham, NH 03824

Dr. Ed Wegman.  
Division Head,  
Mathematical Sciences Division,  
Office of Naval Research,  
800 N. Quincy Street,  
Arlington, VA 22217-5000

Dr. Gregory B. Smith,  
ATT Information Systems,  
190 River Road,  
Summit, NJ 07901

Dr. Lynn Conway,  
University of Michigan,  
263 Chrysler Center,  
Ann Arbor, MI 48109

Dr. John Lowrance,  
SRI International,  
333 Ravenswood Ave.,  
Menlo Park, CA 94025

Dr. David Mizell,  
Office of Naval Research,  
1030 E. Green St.,  
Pasadena, CA 91106

Dr. Richard Lau,  
Office of Naval Research,  
Code 411,  
800 N. Quincy St.,  
Arlington, VA 22217-5000

Dr. Y.S. Wu,  
Naval Research Laboratory,  
Code 7007,  
Washington, D.C. 20375

Dr. Joel Trimble,  
Office of Naval Research,  
Code 251,  
Arlington, VA 22217-5000

Robert A. Ellis,  
Calma Company,  
R & D Engineering,  
525 Sycamore Dr., M/S C510  
Milpitas, CA 95035-7489

Dr. James H. Clark.  
Silicon Graphics, Inc.  
2011 Stierlin Road,  
Mountain View, CA 94043

Edward R. McCracken,  
Silicon Graphics, Inc.  
2011 Stierlin Road,  
Mountain View, CA 94043

Shinji Tomita,  
Dept. of Information Science,  
Kyoto University,  
Sakyo-ku, Kyoto, 606, Japan

Hiroshi Hagiwara,  
Dept. of Information Science,  
Kyoto University,  
Sakyo-ku, Kyoto, 606, Japan

Dr. Alain Fournier.  
Dept. of Computer Science,  
University of Toronto,  
Toronto, Ontario, Canada  
M5S 1A4

Dr. Andries Van Dam,  
Dept. of Computer Science,  
Brown University,  
Providence, RI 02912

Dr. Brian A. Barsky.  
Berkeley Computer Graphics Laboratory,  
Computer Sciences Division,  
Dept. of Electrical Engineering and Computer Sciences,  
University of California,  
Berkeley, CA 94720

Dr. Ivan E. Sutherland,  
Carnegie Mellon University,  
Pittsburg, PA 15213

Dr. Turner Whitted,  
New West Hall (035A).  
University of North Carolina.  
Chapel Hill, NC 27514

Dr. Robert B. Grafton,  
Office of Naval Research,  
Code 433,  
Arlington, Virginia 22217-5000

Professor Eihachiro Nakamae,  
Electric Machinery Laboratory,  
Hiroshima University,  
Higashihiroshima 724, Japan

Carl Machover,  
Machover Associates,  
199 Main Street,  
White Plains, New York 10601

Dr. Buddy Dean,  
Naval Postgraduate School,  
Code 52, Dept. of Computer Science,  
Monterey, California 93943

Earl Billingsley,  
43 Fort Hill Terrace,  
Northhampton, MA 01060

Dr. Jan Cuny,  
University of Massachusetts, Amherst  
Dept. of Computer and Information Science,  
Amherst, MA 01003

Robert Lum,  
Silicon Graphics, Inc.  
2011 Stierlin Road,  
Mountain View, CA 94043

Jeff Hausch,  
Silicon Graphics, Inc.  
2011 Stierlin Road,  
Mountain View, CA 94043

Robert A. Walker,  
7657 Northern Oaks Court,  
Springfield, VA 22153

Dr. Barry L. Kalman,  
Washington University,  
Department of Computer Science,  
Box 1045,  
St. Louis, Missouri 63130

Dr. Will. Randolph Franklin,  
Electrical, Computer, and Systems Engineering Department,  
Rensselaer Polytechnic Institute,  
Troy, New York 12180-3590

Dr. Gershon Kedem,  
Microelectronics Center of North Carolina,  
PO Box 12889,  
3021 Cornwallis Road,  
Research Triangle Park,  
North Carolina 27709

Dr. Branko J. Gerovac,  
Digital Equipment Corporation,  
150 Locke Drive LMO4/H4, Box 1015  
Marlboro, Massachusetts 01752-9115

Robert A. Schumacker,  
Evans and Sutherland,  
PO Box 8700,  
580 Arapeen Drive,  
Salt Lake City, Utah 84108

R. A. Dammkoehler,  
Washington University,  
Department of Computer Science,  
Box 1045,  
St. Louis, Missouri 63130

Dr. Lynn Ten Eyck,  
Interface Software,  
79521 Highway 99N,  
Cottage Grove, Oregon 97424

Kazy K. Yokota,  
Japan Tech Services Corporation,  
3F Ohkura Building,  
1-4-10 Shiba-Daimon,  
Minato-Ku, Tokyo 105, Japan

Toshiaki Yoshinaga,  
Hitachi Works, Hitachi Ltd.  
1-1. Saiwaicho 3 Chome,  
Hitachi-shi, Ibaraki-ken,  
317 Japan

Takatoshi Kodaira,  
Omika Works, Hitachi Ltd.  
2-1. Omika-cho 5-chome,  
Hitachi-shi, Ibaraki-ken,  
319-12 Japan

Atsushi Suzuki,  
Hitachi Engineering, Co. Ltd.  
Model Group,  
2-1. Saiwai-cho 3-Chome,  
Hitachi-shi, Ibaraki-ken,  
317 Japan

Toshio Nishimura,  
Hitachi Engineering, Co. Ltd.  
Mochi Group,  
2-1, Saiwai-cho 3-Chome,  
Hitachi-shi, Ibaraki-ken,  
317 Japan

Dr. John Staudhammer,  
Dept. of Electrical Engineering,  
University of Florida,  
Gainesville, Florida 32611

Dr. Lewis E. Hitchner,  
Computer and Information Science Dept.,  
237 Applied Science Building,  
University of California at Santa Cruz,  
Santa Cruz, California 95064

Dr. Jane Wilhelms,  
Computer and Information Science Dept.,  
237 Applied Science Building,  
University of California at Santa Cruz,  
Santa Cruz, California 95064

Dr. Pat Mantey,  
Computer Engineering Department,  
University of California at Santa Cruz,  
Santa Cruz, California 95064

Dr. Walter A. Burkhardt,  
University of California, San Diego  
Department of Computer Science,  
La Jolla, California 92093

P. K. Rastogi,  
Silicon Graphics, Inc.,  
2011 Shafter Road,  
Mountain View, CA 94043

Bob Broadwell,  
Silicon Graphics, Inc.,  
2011 Shafter Road,  
Mountain View, CA 94043

Nancy Soder,  
Silicon Graphics, Inc.,  
2011 Shafter Road,  
Mountain View, CA 94043

Dr. Toshiyasu L. Kunii,  
Department of Information Science,  
Faculty of Science,  
The University of Tokyo,  
7-3-1 Hongo, Bunkyo-ku, Tokyo 113,  
Japan

Dr. Kazuhiro Fuchi,  
Institute for New Generation Computer Technology,  
Mita-Kokusai Building 21FL,  
1-4-28 Mita, Minato-ku, Tokyo 108. Japan

Tony Loeb,  
Silicon Graphics, Inc.  
1901 Avenue of the Stars,  
Suite 1774.  
Los Angeles. CA 90067

Kevin Hammons,  
NASA AMES-Dryden Flight Research Facility,  
PO Box 273,  
Mail Stop OF1.  
Edwards. California 93523

Sherman Gee,  
Code 221.  
Office of Naval Technology,  
800 N. Quincy St.  
Arlington. VA 22217

Dr. J.A. Adams,  
Department of Mechanical Engineering,  
US Naval Academy,  
Annapolis. MD 21402

Dr. David F. Rogers,  
Dept. of Aerospace Engineering.  
US Naval Academy,  
Annapolis. MD 21402

Dr. Robert F. Franklin.  
Environmental Research Institute of Michigan,  
PO Box 8618.  
Ann Arbor, MI 48107

LT Mark W. Hartong.  
900 Cambridge Dr 17,  
Benicia. CA 94510

Capt. Mike Gaddis,  
DCA/JDSSC/C720,  
1860 Wiehle Ave  
Reston. VA 22090

Lt. Cdr. Patrick G. Hogan, USN  
102 Borden Avenue,  
Wilmington, North Carolina 28403

Dr. Edwin Catmull,  
LucasFilm,  
PO Box 2009,  
San Rafael, CA 94912

Dr. John Beatty,  
Computer Science Department,  
University of Waterloo,  
Waterloo, Ontario,  
Canada N2L 3G1

Dr. James Foley,  
George Washington University,  
Dept. of Electrical Engineering and Computer Science,  
Washington, D.C. 20052

Dr. Donald Greenberg,  
Cornell University,  
Program of Computer Graphics,  
Ithaca, NY 14853

Dr. David Gries,  
Cornell University,  
Computer Science Department,  
405 Upson Hall,  
Ithaca, NY 14853

Dr. Leo J. Guibas,  
Systems Research Center,  
Digital Equipment Corporation,  
130 Lytton Avenue,  
Palo Alto, CA 94301

Dr. S. Ganapathy,  
Ultrasonic Imaging Laboratory,  
Dept. of Electrical and Computer Engineering,  
University of Michigan,  
Ann Arbor, MI 48109

Dr. Hank Christiansen,  
Brigham Young University,  
Dept. of Civil Engineering,  
368 Clyde Bldg.  
Provo, Utah 84602

Dr. Thomas A. DeFanti,  
Dept. of Electrical Engineering & Computer Science,  
University of Illinois at Chicago,  
Box 4348,  
Chicago, IL 60680



Dr. Lansing Hatfield,  
Lawrence Livermore National Laboratory,  
7000 East Avenue,  
PO Box 5504, L-156,  
Livermore, CA 94550

El Wells,  
Naval Ocean Systems Center,  
Code 443,  
San Diego, California 92152

Dr. Al Zied,  
Naval Ocean Systems Center,  
Code 443,  
San Diego, California 92152

Dr. Glen R. Allgaier,  
Naval Ocean Systems Center,  
Code 9302,  
San Diego, California 92152

Richard L. desJardins,  
Defense Advanced Research Projects Agency/IPTO,  
1400 Wilson Boulevard,  
Arlington, VA 22209

Zsuzsa Molnar,  
Silicon Graphics, Inc.  
2011 Stierlin Road,  
Mountain View, CA 94043

Robert Comperini,  
NASA ADFRF,  
PO Box 273,  
Datamax,  
Edwards, California 93523

Shohei Tomita,  
Hitachi Software Engineering Co., Ltd.  
6-81, Onoe-Machi,  
Naka-Ku, Yokohama 231, Japan

Tomo Yamada,  
Digital Computer Limited,  
No. 25 Kowa Building 8-7,  
Sanbancho, Chiyoda-Ku,  
Tokyo 102, Japan

Tohru Gotoh,  
Digital Computer Limited,  
No. 25 Kowa Building 8-7,  
Sanbancho, Chiyoda-Ku,  
Tokyo 102, Japan

Eiji Kumaara,  
Digital Computer Limited,  
No. 25 Kowa Building 8-7,  
Saubancho, Chiyoda-Ku,  
Tokyo 102, Japan

Koichi Morimura,  
Mitsubishi Heavy Industries, Ltd.  
CAD/CAM Engineering Section,  
Systems Engineering Department,  
1-1-1 Ichome, Wadasaki-cho,  
Hyogo-Ku, Kobe 652, Japan

Takayasu Obata,  
Mitsubishi Heavy Industries, Ltd.  
CAD/CAM Engineering Section,  
Systems Engineering Department,  
1-1-1 Ichome, Wadasaki-cho,  
Hyogo-Ku, Kobe 652, Japan

Dr. Arthur I. Karshmer,  
Box 2081,  
Computing Research Laboratory,  
New Mexico State University,  
Las Cruces, New Mexico 88003

Kiyomichi Ohmachi,  
Systems Development Laboratory,  
Hitachi Ltd.,  
1002-1 Izenji Asao-ku,  
Kiyama 661-shi, 215 Japan

John W. Denson,  
Computer Aided Engineering Program,  
Naval Weapons Center,  
Code 3042,  
China Lake, California 93555

Phyllis J. Lee,  
Photolithography, Inc.  
2341 Sterling Road,  
Menlo Park, CA 94043

Wong E. Mungsing,  
15/2 Thantorn Village,  
Wongwongrangsit Rd.  
Bangkok 10210 Thailand

Don Risher,  
Digital Equipment Corporation,  
200 Augustine Drive  
Santa Clara, California 95054

M. Creon Levit.  
NASA, Ames Research Center  
Mail Stop: 233-1  
Moffett Field, California 94035

Dr. Velvin R. Watson,  
NASA, Ames Research Center  
Mail Stop: 202A-14  
Moffett Field, California 94035

Phyllis F. Flynn.  
Trancept Systems, Inc.  
521F Uwharrie Ct.  
Raleigh, North Carolina 27606-1456

Mr. Zesheng Tang,  
Palo Alto Research Center,  
XEROX Corporation,  
3333 Coyote Hill Road.  
Palo Alto, California 94304

Larry Ledden,  
Hughes Aircraft,  
MS-604D216.  
PO Box 3310.  
Fullerton, California 92634

Dr. Robert Leighty,  
Research Institute (CUDE Bldg),  
U.S. Army Engineer Topographic Laboratory,  
Fort Belvoir, VA 22060-5546

Dr. Olin Mintzer.  
Research Institute (CUDE Bldg),  
U.S. Army Engineer Topographic Laboratory,  
Fort Belvoir, VA 22060-5546

Mr. Russell Davis,  
HQ. USACDEC,  
Attention: ATEC-IM,  
Fort Ord, California 93941

Capt. Roger K. Diehl,  
1105 Richmond Drive,  
Stafford, VA 22554

LT Joann M. Ammann,  
Naval Security Group Activity,  
Skaggs Island,  
Sonoma, California 95476-5000

Dr. Edward Riseman.  
University of Massachusetts, Amherst  
Dept. of Computer and Information Science.  
Amherst, MA 01003

END

DITIC

9 - 86